z/OS

Infoprint Server Customization

S544-5744-03

z/OS

Infoprint Server Customization

S544-5744-03

Note

Before using this information and the product it supports, be sure to read the general information in “Notices” on page 297.

Fourth Edition (April 2002)

This edition is a major revision of S544-5744-02. It applies to z/OS Version 1 Release 2, Program Number 5694-A01; to z/OS.e Version 1 Release 3, Program Number 5655-G52; to Infoprint Server Transforms Version 1 Release 1 Modification Level 1, Program Number 5697-F51; and to all subsequent releases and modifications until otherwise indicated in new editions or technical newsletters. Be sure to use the correct edition for the level of the product.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

The IBM Printing Systems Division welcomes your comments. A form for reader’s comments is provided at the back of this publication. If the form has been removed, you may send your comments to the following address:

INFORMATION DEVELOPMENT
THE IBM PRINTING SYSTEMS DIVISION
DEPARTMENT H7FE BUILDING 003G
BOULDER CO 80301-9191
USA

If you prefer to send comments electronically, use one of the following methods:

- Internet: printpub@us.ibm.com
- Fax: 1-800-524-1519 or 1-303-924-6873

Internet

Visit our home pages at www.ibm.com/printers/ and www.ibm.com/servers/eserver/zseries/zos/.

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

© Copyright International Business Machines Corporation 1995, 2002. All rights reserved.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Figures ix

Tables xi

Summary of Changes xiii

About This Publication xix
Who Should Use This Publication xix
How This Publication is Organized xix
Where to Find More Information xix
Web Sites xix
Accessing Licensed Books on the Web xx
Using LookAt to look up message explanations xxi
Preventive Service Planning Information xxi
Publications xxii
Understanding Syntax Notation xxiii

Chapter 1. Introducing Infoprint Server 1
Printer Inventory Manager 3
Windows Client 4
Print Interface 5
Print Interface with an SAP R/3 Application Server Running on the Same z/OS System 9
Print Interface with the SAP R/3 Application Server Running on a Remote System 10
Infoprint Server Transforms 12
Transform Manager 13
NetSpool 14
IP PrintWay 16
SNMP Subagent 19

Chapter 2. Customization Roadmap 23

Chapter 3. Customizing the Printer Inventory Manager 25
Identifying Software Requirements 25
Enabling the Infoprint Server Feature 26
Understanding the Infoprint Server HFS Directories 26
/etc/Printsrv Directory 26
/var/Printsrv Directory 26
Making LE Run-time Libraries Available 27
Creating the Infoprint Server Configuration File (**aopd.conf**) 28
Setting Environment Variables for the Printer Inventory Manager 31
Establishing Security for the Printer Inventory and Operator Commands 33
Customizing Startup and Shutdown Commands and Procedures 35
Editing the **aopstart** EXEC 36
Creating a User ID to Assign to the AOPSTART and AOPSTOP Procedures 38
Defining the AOPSTART and AOPSTOP Procedures to RACF 39
Editing the AOPSTART Startup Procedure 39
Editing the AOPSTOP Shutdown Procedure 40
Starting Infoprint Server Daemons Automatically 42
Enabling Infoprint Server ISPF Panels 42
Defining Libraries in the TSO Logon Procedure 42
Displaying the Infoprint Server Panel Option 43

© Copyright IBM Corp. 1995, 2002

Modifying the AOPINIT EXEC 43
Backing Up and Restoring the Printer Inventory 44
Steps for Backing up the Printer Inventory to an HFS File 44
Steps for Restoring the Printer Inventory from an HFS File 45
Steps for Backing Up the Printer Inventory to a Generation Data Group (GDG) 45
Steps for Restoring the Printer Inventory from a GDG 46

Chapter 4. Customizing Print Interface 49
Identifying Software Requirements 50
Common Customization for Print Interface 51
Customizing z/OS Communications Server 51
Customizing JES to Specify No Blank Truncation 53
Updating the BPXPRMxx Member of SYS1.PARMLIB 54
Editing the aopd.conf Configuration File 54
Writing a Filter Program 55
Customizing the z/OS UNIX Printing Commands 60
Customizing the AOPPRINT JCL Procedure 60
Customizing the Print Interface LPD 61
Customizing z/OS Communications Server (TCP/IP) 62
Editing the aopd.conf Configuration File 64
Using the LPD Compatibility Filter 64
Customizing LPR Clients 65
Customizing Print Interface Support for the z/OS SMB Server 66
Customizing the z/OS SMB Server 67
Customizing the Print Interface IPP Server 67
Ensuring Sufficient Memory is Available to Start the IPP Server 68
Editing the aopd.conf Configuration File 68
Setting Environment Variables for the IPP Server 69
APF-Authorizing Java Run-Time Libraries 69
Customizing the IPP Workstation Client 70
Customizing SAP R/3 71
Defining an SAP User for the Callback Daemon 73
Creating the SAP Callback Daemon Configuration File (**aopsapd.conf**) . . . 73
Setting Environment Variables for the SAP OMS 75
Defining a ROMS and LOMS to the SAP R/3 Application Server 75
Modifying the LOMS to Specify Job Attributes 76
Creating an SAP R/3 Output Device 77
Customizing the Print Interface Subsystem 79
Editing the aopd.conf Configuration File 80
Increasing the Maximum Number of Active Transform Daemons 80

Chapter 5. Customizing Infoprint Server Transforms and the Transform Manager 83
Identifying Software Requirements 85
Installing and Enabling Infoprint Server Transforms 86
Customizing the Transform Manager 86
Updating the BPXPRMxx Member of SYS1.PARMLIB 87
Editing the Infoprint Server Configuration File (**aopd.conf**) 87
Creating the Transform Configuration File (**aopxfd.conf**) 87
Setting Environment Variables for the Transform Manager 91
Customizing the PCL to AFP Transform 92
Editing the Transform Configuration File (**aopxfd.conf**) 92
Customizing the PostScript to AFP and PDF to AFP Transform 95
Ensuring Sufficient Memory is Available to Start the Transform 96
Establishing Security for the PostScript and PDF to AFP Transform 97

Editing the Transform Configuration File (**aopxfd.conf**) 98
Customizing PostScript and PDF to AFP Transform Resource Files 101
Customizing the SAP to AFP Transform 102
Setting Environment Variables for the SAP to AFP Transform 103
Installing Multibyte Conversion Tables for the SAP to AFP Transform . . . 103
Customizing SAP to AFP Configuration Files 104
Customizing the AFP to PCL, AFP to PDF, and AFP to PostScript Transforms 115
Editing the Transform Configuration File (**aopxfd.conf**). 116
Customizing the Font-Mapping Table 133
Scaling 240-Pel to 300-Pel Fonts using the AOXCF30 Program 135

Chapter 6. Customizing the SNMP Subagent 139
Identifying Software Requirements 139
Customizing and Starting the z/OS Communications Server SNMP Agent 140
Editing the **aopd.conf** Configuration File 141
Enabling SNMP Reporting in the Printer Inventory 141
Customizing the SNMP Manager 142

Chapter 7. Customizing NetSpool 143
Customizing JES to Specify No Blank Truncation 143
Allocating and Initializing the Message Log 144
Customizing the NetSpool Startup Procedure 144
Determining the Number of NetSpool Programs to Start 145
Using the Japanese National Language Version 145
Using an Extended MCS Console 146
Creating a User ID to Assign to the NetSpool Startup Procedure 146
Defining the NetSpool Startup Procedure to RACF 146
Editing the NetSpool Startup Procedure 147
Changing LE Run-time Options 150
Writing NetSpool Exit Programs 150
Programming Considerations 151
APIPP1X2 Control Block 151
Beginning of File Exit for SCS Data Streams (APIPPTD1) 153
Beginning of File Exit for 3270 Data Streams (APIUBF3) 155
Transparent Data Control Exit (APIPPTD2) 157
Graphic Escape Exit (APIUGEX) 159
Sample Exits and Macro 161
Installing the Exits 161
Modifying the Exits 162

Chapter 8. Customizing IP PrintWay 163
Identifying Software Requirements 165
Customizing z/OS Communications Server 165
Customizing the TCPIP.DATA Data Set 166
Customizing the PROFILE.TCPIP Data Set 167
Customizing z/OS UNIX Sendmail 167
Defining the IP PrintWay Functional Subsystem 169
Determining How Many Functional Subsystems to Define. 169
Determining How Many Functional Subsystem Applications to Define . . . 169
Determining Work-Selection Criteria. 169
Specifying a Default Font and Page Definition for Data Transforms 170
Defining the IP PrintWay Functional Subsystem in JES2 171
Defining the IP PrintWay Functional Subsystem in JES3 174
Updating the BPXPRMxx Member of SYS1.PARMLIB 178
APF-Authorizing SYS1.IMAGELIB 178
Allocating and Initializing the Transmission-Queue Data Set 178

Restricting Access to the Transmission-Queue Data Set 180
Allocating and Initializing the IP PrintWay Message-Log Data Set 180
Customizing the IP PrintWay Startup Procedure 181
Creating a User ID to Assign to the IP PrintWay Startup Procedure 181
Defining the IP PrintWay Startup Procedure to RACF 182
Editing the ANFWPROC Startup Procedure 182
Changing LE Run-time Options 185
Creating a VTAM APPL Resource Definition. 185
Creating FSS and FSA Definitions in the Printer Inventory 186
Creating FSS Definitions 186
Creating FSA Definitions 188
Writing IP PrintWay Exit programs 189
Resubmit for Filtering Considerations for IP PrintWay Exits 190
Programming Considerations 191
Common Control Blocks 192
Routing Exit (ANFUXRTG) 196
Begin Data Set Exit. 204
Record Exit. 210
End Data Set Exit 216
SMF Exit (ANFUXSMF) 222
Message Exit (ANFUXMSG) 227
Response Notification Exit (ANFUXRSP) 228
Using the Block Letter Program (ANFUBLK) 234
Sample Exits 235
Installing Exits. 236
Modifying Exits 236
Customizing PSF for OS/390 for Sharing Network Printers 236

Chapter 9. Installing and Customizing the Infoprint Server Windows Client 237

Appendix A. Checklist of Customization Tasks 241

Appendix B. Infoprint Server Files 245
aopd.conf -- Infoprint Server Configuration File. 245
Purpose 245
Format 245
Usage Notes 248
aopxfd.conf -- Infoprint Server Transform Configuration File 249
Purpose 249
Format 249
PCL to AFP Transform Entry 250
PostScript and PDF to AFP Transform Entry 252
AFP to PCL, AFP to PostScript, or AFP to PDF Transform Entry 254
Usage Notes 262
Examples 262
aopsapd.conf -- Infoprint Server SAP Callback Daemon Configuration File 264
Purpose 264
Format 264
Usage Notes 264
Examples 265

Appendix C. Environment Variables 267

Appendix D. AOPBATCH Program 271
Format 271
Description 271

Parameters 272
Usage Notes 272
Return Codes 273
Examples 273
Sample Jobs 274

Appendix E. Application Programming Interface (API) 275
API Files. 275
API Data Structures 275
DocumentInfo - Information about a document in a job 276
JobInfo - Information about an Infoprint Server print job and each document in the job 276
JobToken - Identifies an Infoprint Server print job 277
PrinterInfo - Information about an Infoprint Server printer 277
QueryToken - Identifies an Infoprint Server query request 278
Transform - Information About a Transform Connection 278
API Functions 278
Initializing and Terminating the API 278
Creating a Print Job 278
Canceling a Print Job 278
Querying Print Jobs. 279
Querying Printers 279
Connecting to the Transform Manager 279
Handling Errors 279
AbortPrintFile -- Terminate an Infoprint Server print job 279
BeginEnumJobs -- Begin an enumeration of Infoprint Server print jobs for a printer 280
BeginEnumPrinters -- Begin an enumeration of Infoprint Server printers 280
BufferSizeRequired -- Return the required size of the buffer after a buffer overflow error 281
CancelJob -- Cancel a print job 281
ClosePrintFile -- Close an Infoprint Server print job 282
CreatePrintFile -- Create an Infoprint Server print job 282
EndEnumJobs -- End the enumeration of Infoprint Server print jobs for a printer 283
EndEnumPrinters -- End the enumeration of Infoprint Server printers . . . 284
EnumJobs -- Enumerate Infoprint Server print jobs for a printer 284
EnumPrinters -- Enumerate Infoprint Server printers 285
ErrorNumber -- Return the API error code 285
ErrorString -- Return a pointer to the API error message 286
GetJobInfo -- Return the status of an Infoprint Server print job 287
GetPrinterInfo -- Return information about an Infoprint Server printer. . . . 288
InitAPI -- Initialize the API 288
ReleaseTransform -- Release a connection with a transform daemon . . . 289
RequestTransform -- Establish a connection with a transform daemon 289
SetTerminationHandler -- Register a termination function 290
TermAPI -- Terminate the API 290
WritePrintFile -- Write data to an Infoprint Server print job. 291
Programming Considerations 291
Redefining API Function Names 292
Linking to API Functions 292
APF-Authorizing Your Application 292
Writing a Termination Handler Function 292
Customization Requirements 293

Appendix F. Accessibility 295

Using Assistive Technologies 295
Keyboard Navigation of the User Interface 295

Notices 297
Programming Interfaces 298
Trademarks. 298

Glossary 301
Sources 301
References 301

Bibliography 311
Infoprint Server 311
Print Services Facility for OS/390. 311
Advanced Function Presentation (AFP) 311
Infoprint Manager for AIX and Windows 312
z/OS Version 1 Release 2 312
CICS 313
IMS/ESA® Version 6 313
3270 and SNA Data Streams 313

Index 315

Figures

1. Infoprint Server and Infoprint Server Transforms System Diagram . . . 1
2. Printer Inventory Manager . . . 3
3. Print Interface System Diagram . . . 6
4. Printing SAP R/3 Documents with the SAP R/3 Application Server Running on a z/OS System 9
5. Printing SAP R/3 Documents with the SAP R/3 Application Server Running on a Non-z/OS System . . . 11
6. Transform Manager and Infoprint Server Transforms System Diagram . . . 13
7. NetSpool System Diagram . . . 14
8. IP PrintWay System Diagram . . . 17
9. SNMP Subagent System Diagram . . . 20
10. AOPSTART Startup Procedure — SYS1.PROCLIB(AOPSTART) . . . 40
11. AOPSTOP Shutdown Procedure — SYS1.PROCLIB(AOPSTOP) . . . 41
12. Sample UNIX Filter . . . 58
13. AOPPRINT JCL Procedure in SYS1.PROCLIB . . . 61
14. Default **UserInit** Resource File . . . 102
15. Default **barcode.tab** Configuration File . . . 106
16. Default **defcp.tab** Configuration File . . . 108
17. Sample **defcp.tab.japan** Configuration File. . . . 109
18. Header and Courier Portions of Default **fonts.tab** Configuration File . . . 111
19. Heisei Gothic Portion of Default **fonts.tab** Configuration File . . . 111
20. Default **image.tab** Configuration File . . . 112
21. Default **pagedef.tab** Configuration File . . . 114
22. Default **80000000.tab** Configuration File . . . 115
23. Sample Font-Mapping Table — SYS1.SAMPLIB(AOXFONTS) . . . 135
24. Sample JCL to Run the AOXCF30 Font-Scaling Program — SYS1.SAMPLIB(AOXCF30J) 137
25. NetSpool Startup Procedure — SYS1.PROCLIB(APIJPJCL) . . . 148
26. LE Run-time Options for NetSpool . . . 150
27. Format of APIPP1X2 Control Block. . . . 152
28. Sample JES2 Initialization Statements . . . 172
29. Sample JES3 Initialization Statements . . . 175
30. IDCAMS Parameters to Allocate a Transmission-Queue Data Set — SYS1.SAMPLIB(ANFDEAL) 179
31. IP PrintWay Startup Procedure — SYS1.PROCLIB(ANFWPROC) . . . 183
32. LE Run-time Options for IP PrintWay . . . 185
33. Sample APPL Statements for IP PrintWay FSSs . . . 186
34. Format of the IP PrintWay Exit Parameter Area (ANFUEXTP) . . . 194

© Copyright IBM Corp. 1995, 2002

Tables

1. Summary of Infoprint Server Publications xxii
2. Summary of Infoprint Server Functions and Corresponding Infoprint Server Components 23
3. Where to Define Environment Variables Used by the Printer Inventory Manager 32
4. When to Edit the aopstart EXEC 36
5. IPP Attributes and Corresponding Infoprint Server Attributes 71
6. Infoprint Server Transforms Managed by Transform Manager 86
7. How SAP ABAP Box Characters Map to Code Points 107
8. SAP Code Page Configuration Files 115
9. Paper Names and Dimensions 117
10. IP PrintWay FSS Definition. 187
11. Contents of ANFUEXTP Fields When Resubmit for Filtering Option is Selected 191
12. Comparison of Windows Printing Methods 237
13. Where to Define Environment Variables 267

© Copyright IBM Corp. 1995, 2002

Summary of Changes

Summary of Changes
for S544–5744-03
z/OS™ Version 1 Release 2

The book contains information previously presented in S544-5744-02, which supports z/OS Version 1 Release 2.

This book describes functions that have been added in PTFs after the initial release of z/OS V1R2. Ensure that your installation has applied the following Infoprint Server PTFs; otherwise, some information in this book might not be accurate for your system.

- PTF UW85178
- PTF UW87698
- PTF UW88108
- PTF for APAR OW53202

New Information

- Information is added to indicate this book supports z/OS.e™.
- The new Print Interface subsystem can transform output data from a z/OS batch job and write the transformed data to the JES spool. The following sections describe the customization tasks for this subsystem:
  - "Identifying Software Requirements" on page 50 lists the required PTF.
  - "Customizing the Print Interface Subsystem" on page 79 describes the customization tasks.
- The IP PrintWay™ resubmit for filtering function now uses the original job name and JES-assigned job ID to identify the second sysout data set that Print Interface allocates on the JES spool with the transformed output. The following sections describe this support:
  - "Identifying Software Requirements" on page 165 lists the required PTF.
  - "Resubmit for Filtering Considerations for IP PrintWay Exits" on page 190 describes the contents of field XTPUSRID when the resubmit for filtering function is used.
- The AFP to PDF transform supports a new AOP_ANNOTATIONS environment variable, which you can use to suppress the creation of annotations in the PDF file. The following sections describe this support:
  - "Identifying Software Requirements" on page 85 lists the required PTF.
  - "AFP to PCL, AFP to PostScript, or AFP to PDF Transform Entry" on page 123 describes how to specify AOP_ANNOTATIONS.
- The Infoprint Port Monitor now runs on Windows Me and Windows XP systems. Chapter 9, "Installing and Customizing the Infoprint Server Windows Client" on page 237 lists the required PTFs and describes the new **Unattended port** option, which you might want to select if you define z/OS printers as Windows shared printers.

This book contains terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

© Copyright IBM Corp. 1995, 2002

Summary of Changes
for S544–5744-02
z/OS Version 1 Release 2

The book contains information previously presented in S544-5744-01, which supports z/OS Version 1 Release 2.

New Information

- IP PrintWay™ can now send output to e-mail destinations. The following sections describe new customization tasks related to this enhancement:
  - "Identifying Software Requirements" on page 165 lists the required PTFs.
  - "Customizing z/OS UNIX Sendmail" on page 167 describes considerations for customizing z/OS UNIX® sendmail.
  - "Customizing the IP PrintWay Startup Procedure" on page 181 describes environment variable AOPMAILER, which you must set if sendmail is not installed in the default directory.
  - "Writing IP PrintWay Exit programs" on page 189 describes how to change the e-mail address in the IP PrintWay Routing exit. It also describes changes to the XTPIPADR and XTP_ADDR_TYPE fields in the ANFUEXTP control block.
- Infoprint Server can now use Infoprint Manager for AIX to transform PostScript and PDF files to AFP format for printing on an IBM Infoprint Color 130 Plus printer. "Identifying Software Requirements" on page 50 lists the requirements.
- "Understanding the Infoprint Server HFS Directories" on page 26 describes the files that comprise the Printer Inventory.
- "Steps for Backing Up the Printer Inventory to a Generation Data Group (GDG)" on page 45 contains sample JCL for backing up and restoring the Printer Inventory.
- "Updating the BPXPRMxx Member of SYS1.PARMLIB" on page 54 describes how to increase the number of active z/OS UNIX processes allowed when you use Print Interface.
- "Updating the BPXPRMxx Member of SYS1.PARMLIB" on page 87 describes how to add the FILESYSTYPE statement when you customize the Infoprint® Server Transform Manager.
- "Customizing the NetSpool Startup Procedure" on page 144 describes how to create a z/OS UNIX user ID for the NetSpool™ startup procedure.
- "Updating the BPXPRMxx Member of SYS1.PARMLIB" on page 178 describes how to increase the number of active z/OS UNIX processes allowed when you start multiple IP PrintWay FSAs.
- "Customizing the IP PrintWay Startup Procedure" on page 181 describes how to create a z/OS UNIX user ID for the IP PrintWay startup procedure.
- A new appendix describes accessibility features of Infoprint Server.

This book contains terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

Summary of Changes
for S544–5744-01
z/OS Version 1 Release 2

The book contains information previously presented in S544-5744-00, which supports z/OS Version 1 Release 1.

New Information

- NetSpool now can convert data streams from VTAM® applications to PCL data streams. The following sections describe new customization tasks related to this enhancement:
  - "Creating the Infoprint Server Configuration File (**aopd.conf**)" on page 28 describes how to change the default EBCDIC and ASCII code pages that NetSpool uses when it converts EBCDIC data from VTAM applications to ASCII PCL data.
  - "Customizing the NetSpool Startup Procedure" on page 144 describes how to add Language Environment® (LE) and C++ run-time libraries to the NetSpool startup procedure if they are not in the system LNKLST.
  - "Changing LE Run-time Options" on page 150 describes how to modify the NetSpool LE run-time options if the default options are not suitable for your installation.
  - "Customizing JES to Specify No Blank Truncation" on page 143 describes how to specify no blank truncation.
- NetSpool has enhanced its support for installation exits. The following sections describe this enhancement:
  - "APIPP1X2 Control Block" on page 151 describes the new ST2FLAGS field in the APIPP1X2 control block, which is input to all NetSpool exits. Exits can use this field to determine the type of VTAM session and whether NetSpool is creating a PCL output data stream.
  - "Beginning of File Exit for 3270 Data Streams (APIUBF3)" on page 155 describes a new exit, which lets you add text and PCL commands to the beginning of each output data set.
  - "Graphic Escape Exit (APIUGEX)" on page 159 describes a new exit, which lets you replace Graphic Escape characters with other printable characters.
  - "Sample Exits and Macro" on page 161 lists two new sample exits.
- "Steps for Backing Up the Printer Inventory to a Generation Data Group (GDG)" on page 45 describes how to back up the Printer Inventory to a data set in a generation data group (GDG).
- Infoprint Server Transforms now supports new environment variables and values in the transform configuration file. "AFP to PCL, AFP to PostScript, or AFP to PDF Transform Entry" on page 123 describes the following variables:
  - The new AOP_CUTSHEET variable for the AFP™ to PCL, AFP to PDF, and AFP to PostScript transforms
  - The new AOP_FLATE variable for the AFP to PDF transform
  - The new AOP_PJL variable for the AFP to PCL transform
  - The AOP_PAPER variable, which now supports the A5 paper size

  You must apply the PTFs listed in "Identifying Software Requirements" on page 85 if you want to specify these new variables and values.
- "Customizing z/OS Communications Server (TCP/IP)" on page 62 now recommends that you replace the z/OS Communications Server (TCP/IP) LPD with the Print Interface LPD so that you can use all functions that Infoprint Server provides.
- Infoprint Server now provides the following new sample jobs:
  - SYS1.SAMPLIB(AOPCPETC), which copies all of the Infoprint Server configuration files to a directory where you can edit them. "Creating the Infoprint Server Configuration File (**aopd.conf**)" on page 28, "Creating the

SAP Callback Daemon Configuration File (**aopsapd.conf**)" on page 73, and "Creating the Transform Configuration File (**aopxfd.conf**)" on page 87 describe how to use this sample file.

  – SYS1.SAMPBLIB(AOPRACF), which contains all the RACF® commands you need in order to customize the security function in Infoprint Server. "Establishing Security for the Printer Inventory and Operator Commands" on page 33 describes how to use this CLIST.

- Infoprint Server now provides a new and more secure **aopstart** command. You must now specify some of the environment variables used by Infoprint Server in the new **aopstart** EXEC if the default values are not suitable for your installation. The following sections contain new information:
  – "Editing the **aopstart** EXEC" on page 36 describes how to edit the new EXEC.
  – "Starting Infoprint Server Daemons Automatically" on page 42 describes how to specify the **aopstart** command in the **/etc/rc** shell script.
  – Table 13 on page 267 summarizes where you should set the environment variables used by Infoprint Server.
- Infoprint Server now provides new AOPSTART and AOPSTOP procedures with enhanced function. The following sections contain new information related to these procedures:
  – "Creating a User ID to Assign to the AOPSTART and AOPSTOP Procedures" on page 38 provides more detailed instructions.
  – "Defining the AOPSTART and AOPSTOP Procedures to RACF" on page 39 provides more detailed instructions.
  – "Editing the AOPSTART Startup Procedure" on page 39 shows the new procedure and describes how to edit it.
  – "Editing the AOPSTOP Shutdown Procedure" on page 40 shows the new procedure and describes how to edit it.
- The SAP to AFP transform now supports SAP Release 4.6C. See "Identifying Software Requirements" on page 85 for the number of the APAR you need to apply. "barcode.tab Configuration File" on page 104 describes the new ModWidth and Ratio keywords.
- "Installing Multibyte Conversion Tables for the SAP to AFP Transform" on page 103 provides a step-by-step procedure for creating multibyte conversion tables required by installations that use Asian versions of SAP R/3.
- "Setting Environment Variables for the SAP OMS" on page 75 describes the requirement to set the LIBPATH environment variable.
- IP PrintWay now passes more information to its exits. New fields in the ANFUEXTP control block are: XTP_JCL_DEST, XTP_JCL_CLASS, XTP_JCL_FORMS, and XTP_MISC_FLAGS. "Writing IP PrintWay Exit programs" on page 189 describes this new support.

Changed Information

- You no longer need to create the the **/etc/Printsrv** directory because the install process now automatically creates it. "Understanding the Infoprint Server HFS Directories" on page 26 describes this change.
- The Infoprint Server IPP Server no longer supports Java™ 1.1.8. The IPP Server now requires Java 1.3.
- IP PrintWay now adds the printer instructions specified in the **Document header** field to the data to be transmitted to the printer *after* it calls the Begin Data Set exit. Also, IP PrintWay now adds the printer instructions specified in the

Document trailer field *before* it calls the End Data Set exit. “Begin Data Set Exit” on page 204 and “End Data Set Exit” on page 216 describe this change and what it means to the exits.

- “Allocating and Initializing the Transmission-Queue Data Set” on page 178 shows the changed ANFDEAL sample JCL.
- A redbook is now available for Infoprint Server at http://ibm.com/redbooks.

Deleted Information

- The AFP Printer Driver and the AFP Viewer plug-in for Windows® are no longer shipped with the Infoprint Server Windows client. Information about how to customize the AFP Printer Driver and AFP Viewer have been removed from section Chapter 9, “Installing and Customizing the Infoprint Server Windows Client” on page 237. For information about these products, visit http://www.ibm.com/printers/download.html.
- Infoprint Server no longer can issue messages or display ISPF panels in Spanish. Information about how to view messages and ISPF panels in Spanish has been removed.
- The appendices that show which SCS and 3270 code points NetSpool supports have been removed and are now located only in *z/OS Infoprint Server User’s Guide*.

This book contains terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

About This Publication

This publication describes how to customize the following products:

- Infoprint Server, an element of z/OS Version 1 Release 2 (5694–A01), and higher; and an element of z/OS.e Version 1 Release 3 (5655-G52), and higher
- Infoprint Server Transforms Version 1 Release 1 Modification Level 1, a separate IBM® program product (5697-F51)

Who Should Use This Publication

This publication 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).

How This Publication is Organized

This publication is organized so that you can read it sequentially or refer directly to individual tasks.

Chapter 1 introduces Infoprint Server and describes how the different components of Infoprint Server fit into your system and the functions each component provides. Use this introduction to determine which components of Infoprint Server you want to use in your installation. This chapter is the same as the introduction chapter in other Infoprint Server publications.

Chapter 2 lists the major functions provided by Infoprint Server, identifies the components of Infoprint Server that you need to customize to use each function, and directs you to the sections in this publication that describe the customization tasks you need to perform.

Chapters 3 – 9 describe the customization tasks required to customize each component of Infoprint Server. Read only the chapters that pertain to the component you want to customize for your installation.

Appendices provide the following reference information:

- Appendix A contains a checklist of customization tasks.
- Appendix B summarizes the syntax of Infoprint Server configuration files.
- Appendix C summarizes the environment variables used by Infoprint Server.
- Appendix D describes the AOPBATCH utility program, which simplifies running Infoprint Server programs in HFS files as batch JCL jobs.
- Appendix E describes the callable services provided by Infoprint Server.
- Appendix F describes accessibility features.

This publication also includes a glossary and bibliography.

Where to Find More Information

This section describes where to find information related to z/OS, Infoprint Server, and Infoprint Server Transforms.

Web Sites

These Web sites contain related information:

- http://www.ibm.com/printers/

© Copyright IBM Corp. 1995, 2002

This site contains information about printing products, including:
  - An overview of Infoprint Server, including the same printing scenarios that you can find in *z/OS Infoprint Server Introduction*.
  - Infoprint Server publications and other publications related to printing. These publications are in PDF format.
- http://www.ibm.com/printers/download.html

  This site contains downloads for Windows systems, including the Infoprint Port Monitor, the AFP Viewer plug-in, the AFP Printer Driver, and Network Printer Manager (NPM) for the Web.
- http://www.ibm.com/servers/eserver/zseries/zos/

  This site contains information about z/OS.
- http://www.ibm.com/servers/eserver/zseries/zos/bkserv/

  This site contains z/OS documentation, including:
  - All z/OS publications in both PDF and BookManager format.
  - Documentation updates that result from APARs and PTFs.
- http://ibm.com/redbooks

  This site contains IBM redbooks, including a redbook for Infoprint Server.
- http://www.ibm.com/servers/eserver/zseries/zos/unix/

  This site contains information about z/OS UNIX System Services.

Accessing Licensed Books on the Web

z/OS licensed documentation in PDF format is available on the Internet at the IBM Resource Link Web site at:

`http://www.ibm.com/servers/resourcelink/`

Licensed books are available only to customers with a z/OS license. Access to these books requires an IBM Resource Link Web userid and password, and a key code. With your z/OS order you received a memo that includes this key code.

To obtain your IBM Resource Link Web userid and password log on to:

`http://www.ibm.com/servers/resourcelink/`

To register for access to the z/OS licensed books:

1. Log on to Resource Link using your Resource Link userid and password.
2. Click on **User Profiles** located on the left-hand navigation bar.
3. Click on **Access Profile.**
4. Click on **Request Access to Licensed books.**
5. Supply your key code where requested and click on the **Submit** button.

If you supplied the correct key code you will receive confirmation that your request is being processed. After your request is processed you will receive an e-mail confirmation.

Note: You cannot access the z/OS licensed books unless you have registered for access to them and received an e-mail confirmation informing you that your request has been processed.

To access the licensed books:

1. Log on to Resource Link using your Resource Link userid and password.
2. Click on **Library**.

3. Click on **zSeries**.
4. Click on **Software**.
5. Click on **z/OS**.
6. Access the licensed book by selecting the appropriate element.

Using LookAt to look up message explanations

LookAt is an online facility that allows you to look up explanations for most of the z/OS, z/VM, and VSE messages you encounter, as well as system abends and some codes. Using LookAt to find information is faster than a conventional search because in most cases LookAt goes directly to the message explanation.

You can access LookAt from the Internet at:

`http://www.ibm.com/servers/eserver/zseries/zos/bkserv/lookat/`

or from anywhere in z/OS where you can access a TSO command line (for example, TSO prompt, ISPF, z/OS UNIX System Services running OMVS). You can also download code from the *z/OS Collection* (SK3T-4269) and the LookAt Web site so you can access LookAt from a PalmPilot (Palm VIIx suggested).

To use LookAt on the Internet to find a message explanation, go to the LookAt Web site and simply enter the message identifier (for example, `$HASP701` or `$HASP*`). You can select a specific release to narrow your search.

To use LookAt as a TSO command, you must have LookAt installed on your host system. You can obtain the LookAt code for TSO from a disk on your *z/OS Collection* (SK3T-4269) or from the LookAt Web site. To obtain the code from the LookAt Web site, do the following:

1. Go to http://www.ibm.com/servers/eserver/zseries/zos/bkserv/lookat/.
2. Click **News**.
3. Scroll to **Download LookAt Code for TSO and z/VM.**
4. Click the ftp link, which will take you to a list of operating systems. Click the appropriate operating system. Then click the appropriate release.
5. Open the **lookat.me** file and follow its detailed instructions.

After you have LookAt installed, you can access a message explanation from a TSO command line by entering: **lookat** *message-id*. LookAt will display the message explanation for the message requested.

Note: Some messages have information in more than one book. For example, IEC192I can be found in *z/OS MVS System Messages, Vol 7 (IEB-IEE)* and also in *z/OS MVS Routing and Descriptor Codes*. For such messages, LookAt displays a list of books in which the message appears. You can then select one of the books to view the message explanation.

Preventive Service Planning Information

Before installing Infoprint Server, you should review the current Preventive Service Planning (PSP) information, also called the PSP bucket. You should also periodically review the current PSP information. The PSP upgrade ID is: ZOSV1R2; the subset for Infoprint Server is: INFOPRINT.

To obtain the current PSP bucket, contact the IBM Support Center or use z/OS SoftwareXcel (IBMLink). If you obtained z/OS as part of a CBPDO, HOLDDATA and

PSP information is included on the CBPDO tape; however, this information might not be current if the CBPDO tape was shipped several weeks prior to installation.

Publications

See “Bibliography” on page 311 for a list of the publications referred to in this book and publications that contain additional information about related products. For titles and order numbers of the books for *all* products that are part of z/OS, refer to *z/OS Information Roadmap*.

Infoprint Server for z/OS Implementation Redbook, SG24-6234, is available on the Web at: `http://ibm.com/redbooks`

Table 1 summarizes the publications in the Infoprint Server product library.

Table 1. Summary of Infoprint Server Publications

Publication	Form number
z/OS Infoprint Server Introduction Introduces all components of Infoprint Server, including IP PrintWay, NetSpool, and Print Interface. It also introduces Infoprint Server Transforms. This publication contains printing scenarios that show how you can use Infoprint Server in your installation.	S544-5742
z/OS Infoprint Server Migration Summarizes the new function in Infoprint Server and Infoprint Server Transforms and describes the migration tasks required to implement each new function in your installation. It also describes the Infoprint Server migration program, which converts IP PrintWay, NetSpool, and Print Interface printer information to the format required by Infoprint Server for OS/390® V2R8 and higher.	G544-5743
z/OS Infoprint Server Customization Describes customization tasks for all components of Infoprint Server, including IP PrintWay, NetSpool, and Print Interface. It also describes customization tasks for Infoprint Server Transforms. This publication describes required environment variables, configuration files, startup procedures, how to write exit routines and filter programs, and how to use the Infoprint Server API.	S544-5744
z/OS Infoprint Server Operation and Administration Describes operator procedures and administrative tasks for all components of Infoprint Server, including IP PrintWay, NetSpool, and Print Interface. This publication describes how to start and stop Infoprint Server and how the operator can manage the IP PrintWay transmission queue. It describes how the administrator can create entries in the Printer Inventory using either ISPF panels or the Printer Inventory Definition Utility (PIDU) program, define NetSpool printer LUs to VTAM, and use accounting records written by IP PrintWay.	S544-5745

Table 1. Summary of Infoprint Server Publications (continued)

Publication	Form number
z/OS Infoprint Server User's Guide Describes how to submit print jobs from remote systems (including Windows systems), the local z/OS system, and Virtual Telecommunications Access Method (VTAM) applications. It describes these z/OS UNIX commands: **afp2pcl**, **afp2pdf**, **afp2ps**, **cancel**, **lp**, **lpstat**, **pcl2afp**, **pdf2afp**, **ps2afp**, and **sap2afp**; the AOPPRINT JCL procedure; the AOPBATCH program; DD and OUTPUT JCL parameters supported by Infoprint Server; and how to download and install the Infoprint Port Monitor for Windows.	S544-5746
z/OS Infoprint Server Messages and Diagnosis Describes messages issued by all components of Infoprint Server, including IP PrintWay, NetSpool, and Print Interface. It also describes Infoprint Server Transforms messages and how to use Infoprint Server tracing facilities to diagnose and report errors.	G544-5747

Understanding Syntax Notation

The following rules apply to coding illustrations throughout this publication:

- Uppercase or bold letters are to be coded as shown.
- Variable data is printed in italics. Enter specific data to replace the characters in italics.
- Do not enter the following symbols as part of a parameter or option unless directed to do so:

  Vertical Bar |

  Underscore ___

  Brackets [ ]

  Braces { }
- A vertical bar between two values means that you select one of the values.
- An underscored value means that if an option is not specified, the underscored value, called the default, is used.
- Brackets around a value mean that you do not have to select the value.
- Braces around a value generally mean that you must select one of the values; however, in some cases, you must code the braces. The text identifies when the braces are required.

Chapter 1. Introducing Infoprint Server

Infoprint Server and Infoprint Server Transforms provide support for LAN and host printing on your z/OS system. Figure 1 shows how the components of Infoprint Server and Infoprint Server Transforms fit into your system. The components of Infoprint Server and Infoprint Server Transforms are shaded. (When you view the PDF file, the components of Infoprint Server are yellow, and the components of Infoprint Server Transforms are blue.) Following the figure is a description of each component.

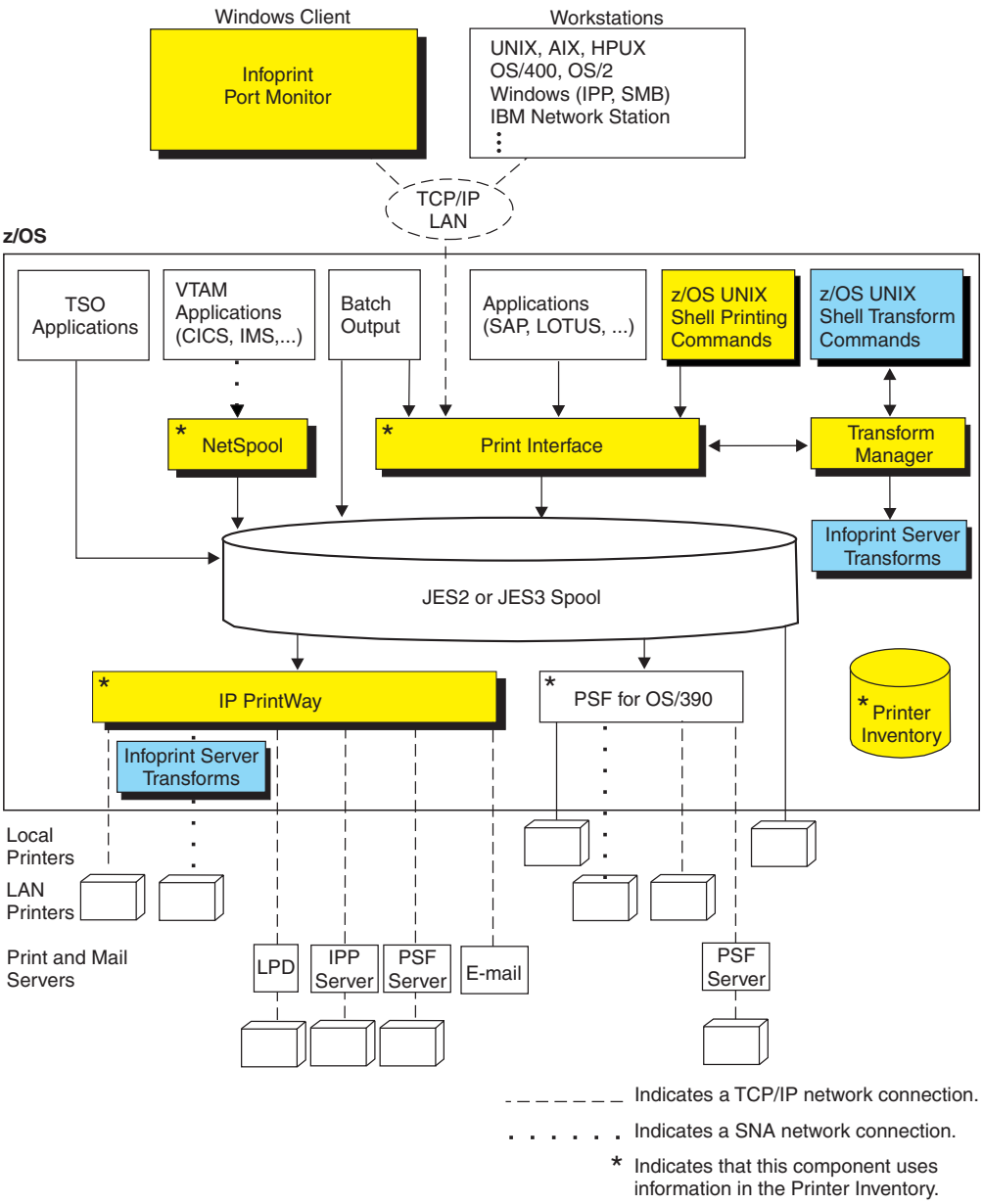


Figure 1. Infoprint Server and Infoprint Server Transforms System Diagram

© Copyright IBM Corp. 1995, 2002

Printer Inventory and Printer Inventory Manager
The Printer Inventory Manager component of Infoprint Server controls the Printer Inventory. The Printer Inventory consists of files in the hierarchical file system (HFS) that contain information about each printer and each e-mail destination. The Printer Inventory also contains system configuration information for IP PrintWay. Optionally, the Printer Inventory can contain system configuration information for PSF for OS/390.

Infoprint Server Windows Client
The Infoprint Server Windows client consists of the Infoprint Port Monitor, which runs on a Windows system and automatically sends print requests and job attributes to the Print Interface component of Infoprint Server running on the z/OS system.

Print Interface
The Print Interface component of Infoprint Server processes print requests from remote clients and from the local z/OS system. Print Interface accepts several different data formats, converts data between EBCDIC and ASCII, transforms data to a format accepted by the printer, and allocates output data sets on the JES spool.

Infoprint Server Transforms
Infoprint Server Transforms, a separate licensed program product (5697-F51), provides transforms that convert data from one format to another on the local z/OS system.

Transform Manager
The Infoprint Server Transform Manager component of Infoprint Server manages many of the transforms provided by Infoprint Server Transforms.

NetSpool
The NetSpool component of Infoprint Server processes print requests from VTAM applications, such as CICS® and IMS™. NetSpool accepts SCS, 3270, and binary data streams, converts SCS and 3270 data streams to either line data streams or PCL data streams, and allocates output data sets on the JES spool.

IP PrintWay
The IP PrintWay component of Infoprint Server transmits data sets from the JES spool to printers or print servers in a TCP/IP or SNA network. IP PrintWay also can transmit data sets to e-mail destinations.

Simple Network Management Protocol (SNMP) subagent (not shown in figure)
The SNMP subagent of Infoprint Server lets you use an SNMP manager to view printer characteristics and printer status for printers controlled by PSF for OS/390 that do not have internal SNMP agents or are not TCP/IP-attached to PSF.

PSF for OS/390 (a separate product)
PSF for OS/390 (5655–B17) is a separate product that can print output on IBM AFP printers. The PSF system programmer can specify PSF printer configuration information in the Printer Inventory that PSF can use when it starts a printer. For information about how to customize PSF to use the Printer Inventory, refer to *PSF for OS/390 & z/OS: Customization*.

The following sections describe each of the Infoprint Server components in more detail.

Printer Inventory Manager

The Printer Inventory Manager component of Infoprint Server controls the Printer Inventory. The Printer Inventory consists of HFS files that contain information about the printing environment. The administrator must create and manage information in the Printer Inventory.

Note: The Printer Inventory *cannot* be shared by Infoprint Server running at the same or different levels on other systems.

The administrator can create the following objects in the Printer Inventory:

- Printer definitions, which contain information about printers and e-mail destinations.
- Printer pool definitions, which contain information about groups of printers and e-mail destinations to which NetSpool can broadcast data.
- FSS definitions, which contain configuration information for IP PrintWay functional subsystems (FSSs). Optionally, the administrator can create FSS definitions for PSF for OS/390 FSSs.
- FSA definitions, which contain configuration information for IP PrintWay functional subsystem applications (FSAs). Optionally, the administrator can create FSA definitions for PSF for OS/390 FSAs.

Figure 2 shows how the administrator can create definitions in the Printer Inventory and which components of Infoprint Server use the Printer Inventory.

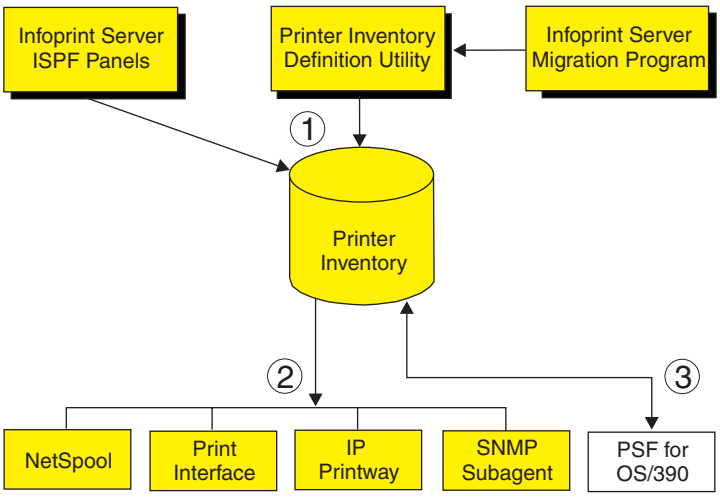


Figure 2. Printer Inventory Manager

1. The administrator can use Infoprint Server ISPF panels and the Printer Inventory Definition Utility (PIDU) to create and maintain the Printer Inventory. The PIDU is useful for creating many printer definitions at the same time and for backing up the Printer Inventory.
2. The following Infoprint Server components use information in the Printer Inventory:
   - NetSpool uses information in printer definitions and in printer pool definitions.
   - Print Interface uses information in printer definitions.
   - IP PrintWay uses information in printer definitions. IP PrintWay also uses configuration information in FSS and FSA definitions.

- The SNMP subagent uses printer information that PSF for OS/390 stores in the Printer Inventory about PSF printers.

3. PSF for OS/390, a separate product, can optionally use printer configuration information that the PSF system programmer specifies in FSS and FSA definitions in the Printer Inventory.

   The printer configuration information in the FSS and FSA definitions is the same as the configuration information that the system programmer can alternatively specify in PSF startup procedures and PSF exits. When the printer configuration information is specified in the Printer Inventory, however, the PSF system programmer can change it without restarting all PSF printers in the PSF functional subsystem (FSS); only the PSF printers with changed configuration information need to be restarted.

   For information about how the PSF system programmer can customize PSF for OS/390 to use the Printer Inventory, refer to *PSF for OS/390 & z/OS: Customization*.

Additional functions provided by the Printer Inventory Manager are:

- **Migration program**

  The Infoprint Server migration program helps the administrator migrate from earlier releases of IP PrintWay, NetSpool, and the OS/390 Print Server. The migration program merges printer information formerly specified in NetSpool print characteristics data sets, NetSpool tables, NetSpool startup procedures, IP PrintWay routing and options data sets, and the Print Interface printer inventory to create entries (such as printer definitions and printer pool definitions) in the new Infoprint Server Printer Inventory.

  The migration program can also move printer information located in PSF startup procedures to FSS and FSA definitions in the Printer Inventory.

- **Security**

  The administrator must restrict access to the Printer Inventory and to the operator commands that start and stop the Printer Inventory Manager, the Print Interface LPD, the Print Interface IPP server, the Transform Manager, and the SNMP subagent.

Windows Client

The Infoprint Server Windows client consists of the following program that runs on Windows 95/98, NT, and 2000 systems:

Infoprint Port Monitor for Windows

The Infoprint Port Monitor for Windows lets users print documents using standard print-submission methods from any Windows application that supports printing. After the Infoprint Port Monitor for Windows is installed and configured on the Windows system, the Port Monitor automatically sends documents to the Print Interface component of Infoprint Server.

Note: Infoprint Server also supports printing from a Windows system with the SMB protocol and the IPP protocol. To use these protocols, Windows users do not need to install the Infoprint Port Monitor for Windows.

The following related products also run on Windows 95/98, NT, and 2000 systems. Although they are *not* part of the Infoprint Server Windows client, you might want to use them if your installation has IBM AFP printers or AFP documents.

- **AFP Printer Driver for Windows**

  The AFP Printer Driver creates output files in AFP format. Files in AFP format can be printed on IBM 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.
- **AFP Viewer Plug-in for Windows**

  The AFP Viewer plug-in lets you view documents in AFP format, for example documents downloaded from the z/OS system or documents on the Web. The AFP Viewer plug-in also lets you print AFP documents to IBM AFP printers as well as non-AFP printers.

The Infoprint Port Monitor for Windows is shipped with Infoprint Server. You can also download the Infoprint Port Monitor for Windows, the AFP Printer Driver for Windows, and the AFP Viewer plug-in for Windows directly to your Windows systems from the IBM Printing Systems Division (PSD) Web site: http://www.ibm.com/printers/download.html.

Print Interface

The Print Interface component of Infoprint Server processes print requests received from both remote clients and local users. Figure 3 on page 6 shows the steps that occur from the time Print Interface receives a print request until it allocates an output data set on the JES spool. An explanation of each step follows.

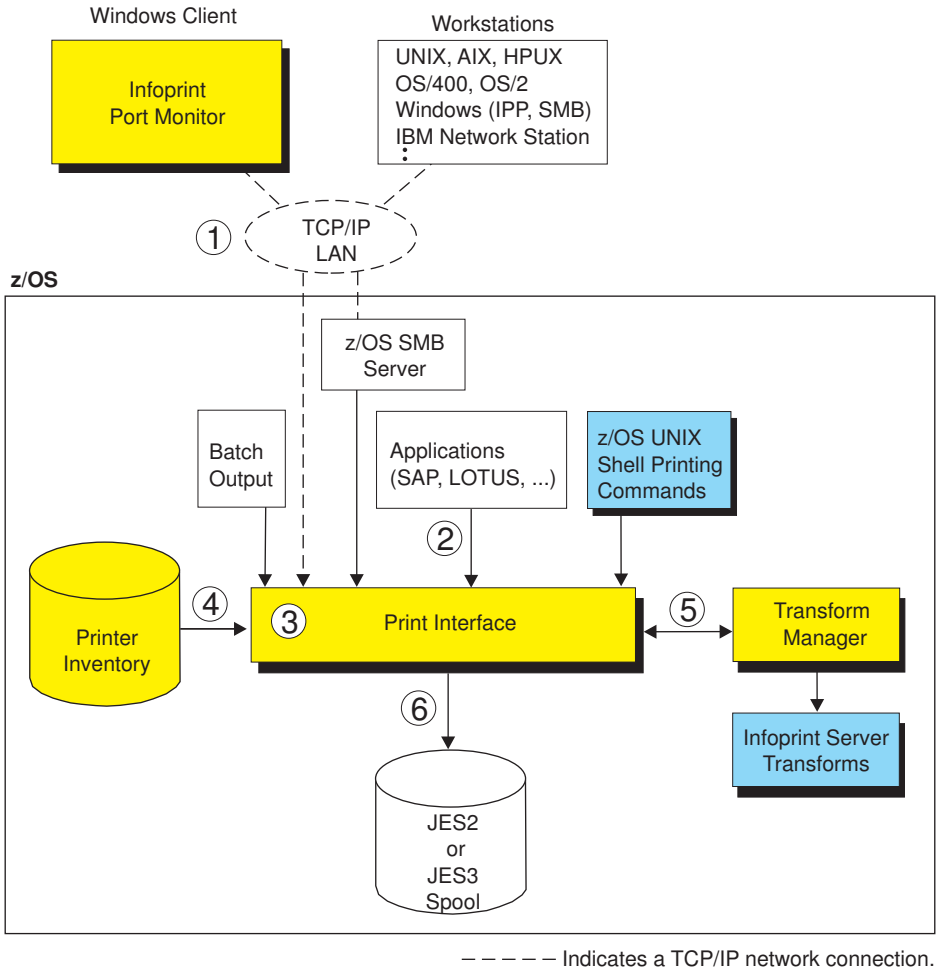


Figure 3. Print Interface System Diagram

1. 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: The LPR protocol is defined by RFC 1179. Clients that use this protocol include:
     - The Infoprint Port Monitor for Windows; Infoprint Server provides this client.
     - TCP/IP commands such as **lpr**, **enq**, and **lpq**.
     - An SAP R/3 application server that runs on a remote system.
   - Internet Printing Protocol (IPP): IPP is a standard protocol for printing over the Internet. An IPP client must run in the remote system.
   - Server Message Block (SMB) printing protocol: SMB is the standard printing protocol used by Windows systems. The z/OS SMB server must be installed on the z/OS system to receive print requests. The z/OS SMB server uses Print Interface callable services to allocate output data sets on the JES spool and return print job status to the client.
2. Users can submit print requests from the local z/OS system using one of the following methods:
   - Print Interface subsystem: Using the Print Interface subsystem, you can transform and print output data created by a batch application with minimal changes to your JCL.

- AOPPRINT: Using the AOPPRINT JCL procedure, you can print existing MVS data sets and UNIX files.
- z/OS UNIX printing commands (**lp**, **lpstat**, and **cancel**) provided by Infoprint Server; Using these commands, which adhere to the XPG4.2 standard, you can print MVS data sets and UNIX files, query the status of a print job, and cancel a print job. You can run these commands from the z/OS UNIX command line or from a UNIX application.
- Infoprint Server SAP Output Management System (OMS). Using the Infoprint Server SAP OMS and the SAP R/3 Application Server for z/OS, SAP R/3 users can submit a print job and receive immediate notification about job events.

3. Print Interface runs as a UNIX application that uses the services of z/OS UNIX System Services.

   Print Interface accepts data in any format, including but not limited to the following formats: line data, MO:DCA-P (also known as AFP), PostScript, PDF, PCL, SAP (OTF and ABAP), and text.
4. Each print request specifies the name of a printer definition in the Printer Inventory. The printer definition can describe a printer or an e-mail destination. 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.
5. Print Interface can, in most cases, automatically detect the data format of the input data and validate that the printer accepts that data format. Print Interface can convert data to EBCDIC or ASCII. It can also call transforms provided by Infoprint Server Transforms to convert data from one data format to another.
6. For each print request, Print Interface dynamically allocates an output data set on the JES2 or JES3 spool using JES allocation parameters specified in the printer definition, including:
   - 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.

Some additional functions provided by Print Interface include:

- **Validation of print requests**

  Before accepting print requests, Print Interface can validate, with some exceptions, that the document can print as requested on the selected printer. For example, Print Interface can reject documents with data formats that the printer does not support or that are too large to print on the selected printer.
- **Automatic data transforms**

  Print Interface can, in most cases, automatically detect the input data format and transform data into the format required by the printer or e-mail destination. Print Interface can perform the following data transforms:
  - Print Interface can transform line data (for example, in a sequential data set or a partitioned data set) into text data for printing on a printer such as an IBM network printer.
  - Print Interface can transform text data into line data for printing on an IBM AFP printer.

- Print Interface can transform PCL, PostScript, PDF, and SAP (OTF and ABAP) data into AFP or line data for printing on an IBM AFP printer. The Infoprint Server Transforms product is required.
- Print Interface can transform line data or AFP data into PCL, PostScript, or PDF format. Infoprint Server Transforms is required.
- Print Interface can transform PCL, PDF, and PostScript data to AFP format on an AIX or Windows system. Infoprint Manager V3R2 for AIX or Infoprint Manager V1R1 for Windows NT and Windows 2000 is required. Print Interface can use the color transform provided with Infoprint Manager V3R2 for AIX to transform color PDF and PostScript data to AFP format for printing on an IBM Infoprint Color 130 Plus printer.

• **Notification of completion**

  Print Interface can notify users on the local z/OS system when processing of a document is complete and the data set has been removed from the JES spool. It can also notify users who request mail notification with a command, such as **lpr**, that uses the LPR to LPD protocol.

• **Status reporting**

  Print Interface can report the status of its data sets that are still on the JES spool. It can report if the data set has been selected for processing, held by the system, retained due to a failed transmission to a LAN printer or an e-mail destination, or deleted before printing.

• **Identification of printed output**

  Print Interface retains the user ID of the job submitter for printing on separator pages and for display on the JES spool, so that the user ID can be printed on separator pages and the operator can view the name of the job submitter when the data set is on the JES spool.

• **Double-byte character set (DBCS) support**

  Print Interface can convert DBCS data from one code page to another before writing the data to the JES spool.

• **Filter support**

  An installation can write a filter program to modify data before Print Interface writes the data to an output data set. A filter can be used to add a separator page or modify data. For example, an installation can write a filter to transform data from one format to another.

• **SAP Output Management System (OMS)**

  Print Interface provides an SAP OMS with a Callback daemon to support printing with the SAP R/3 Application Server for z/OS. The OMS and Callback daemon let SAP users print, cancel jobs, obtain job status, and receive immediate notification about job events.

  The SAP-certified functions provided in Print Interface are:

  - OMS Polling Interface
  - OMS/XOM Callback Interface
  - OMS Operations Supplement

  For more information about SAP certification, visit the SAP Web site at `www.sap.com/solutions/compsoft/cspdirectory`. “Print Interface with an SAP R/3 Application Server Running on the Same z/OS System” on page 9 describes this support.

  The Print Interface LPD can print documents received from an SAP R/3 application server running on another system, such as an AIX or Windows NT

system. “Print Interface with the SAP R/3 Application Server Running on a Remote System” on page 10 describes this support.

Print Interface with an SAP R/3 Application Server Running on the Same z/OS System

Figure 4 shows how the Print Interface SAP Output Management System (OMS), Callback daemon, and the SAP R/3 Application Server for z/OS fit into your system. When an SAP R/3 application server runs on the z/OS system, the Print Interface OMS receives print and status requests, and the Callback daemon provides immediate notification of job events. The SAP R/3 Application Server for z/OS and its spool work process must run on the same system as Infoprint Server. 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 that the SAP spool work process runs on the z/OS system.

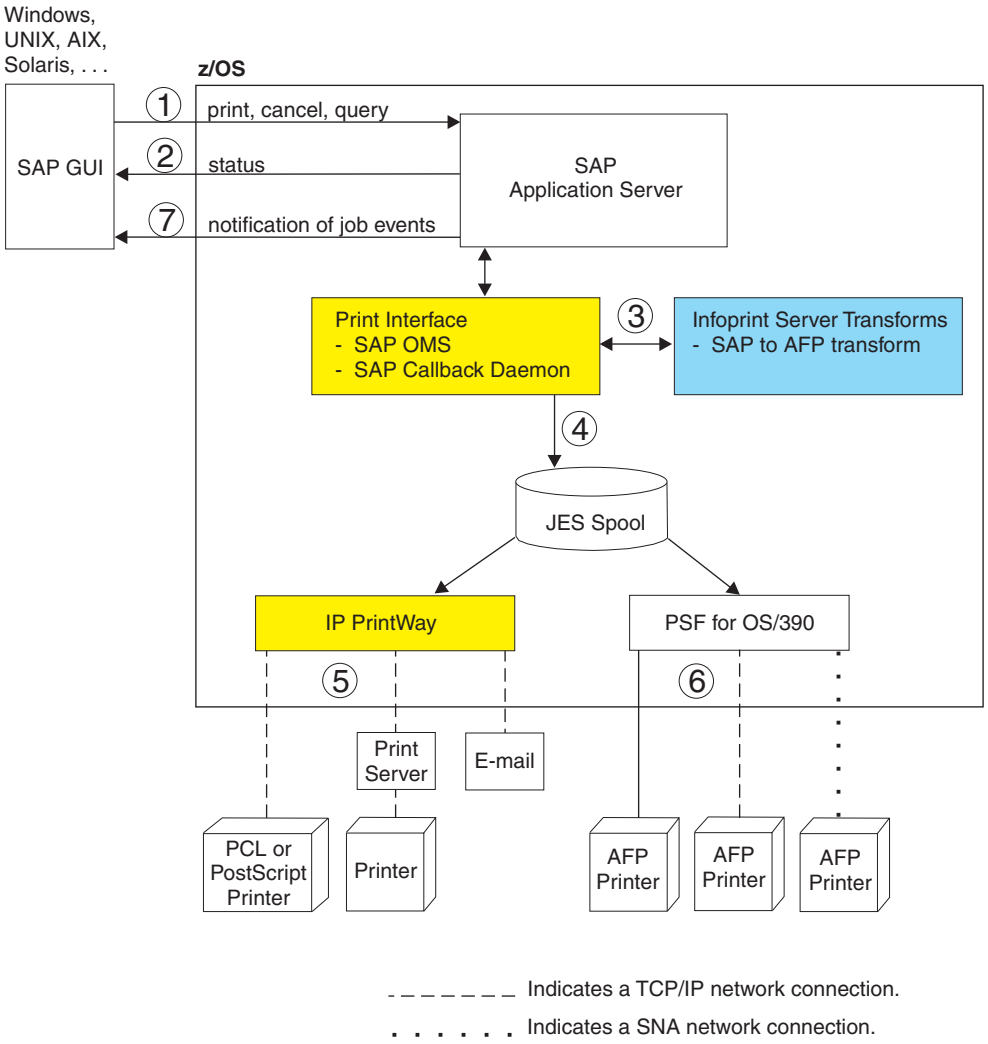


Figure 4. Printing SAP R/3 Documents with the SAP R/3 Application Server Running on a z/OS System

1. From an 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.

2. For a status request, the Print Interface SAP OMS returns the status of a print job or a list of print jobs that the printer is processing.
3. For a print request, Print Interface detects the data format of the input document and performs different processing depending on the type of data:
   - If SAP OTF or ABAP data is to be printed on an AFP printer, Print Interface calls Infoprint Server Transforms to transform the data to AFP format. The SAP to AFP transform uses transform options specified in the printer definition. The administrator must specify the correct transform filter in the printer definition to use transforms.
   - If PCL or PostScript data is to be printed on an IP PrintWay-controlled printer, Print Interface typically does not modify the data; however, the administrator can specify a filter that modifies data in the printer definition.
4. Print Interface creates an output data set on the JES spool. From the JES spool, IP PrintWay, PSF for OS/390, or JES can print the document.
5. IP PrintWay selects data sets from the JES spool and transmits them to remote printers, print servers, or e-mail destinations.
6. PSF for OS/390 selects data sets from the JES spool and prints them on IBM AFP printers. The printers can be local, TCP/IP-attached, or SNA-attached.
7. As print jobs complete (successfully or unsuccessfully), the Print Interface SAP Callback daemon sends notification back to the SAP R/3 system.

Print Interface with the SAP R/3 Application Server Running on a Remote System

Figure 5 on page 11 shows how the Print Interface LPD and an SAP R/3 application server running on a remote system (such as AIX or Windows NT) fit into your system. When no SAP R/3 application server is running on the z/OS system, the Infoprint Server LPD (as opposed to the Infoprint Server OMS) receives print and status requests, and notification of job events does not occur.

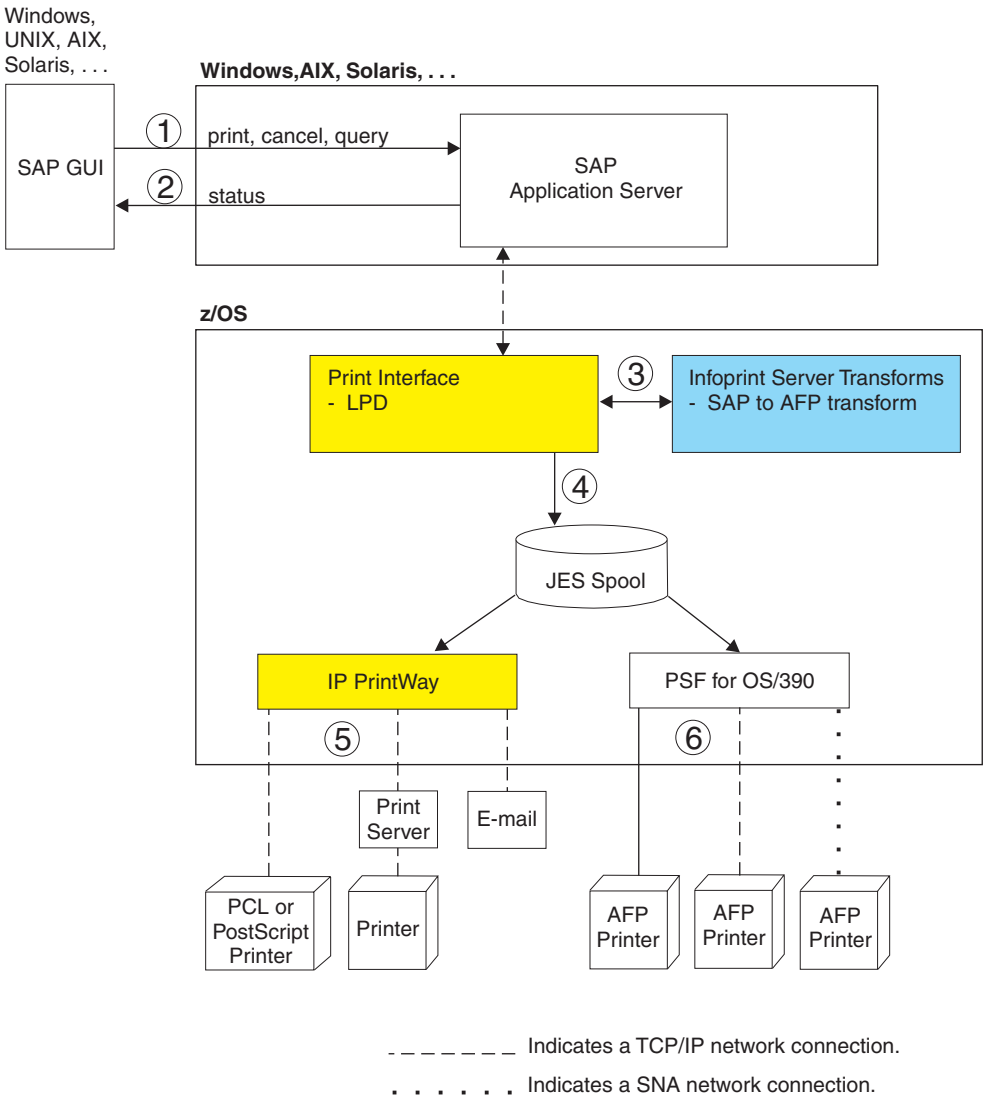


Figure 5. Printing SAP R/3 Documents with the SAP R/3 Application Server Running on a Non-z/OS System

1. From an SAP GUI, users make print, cancel, job query, and device query requests to an SAP R/3 application server running on a remote system (such as Windows NT, AIX, or Solaris. These print requests specify the name of an SAP R/3 output device that is defined to the SAP R/3 system. The SAP administrator associates each output device with a printer definition in the Infoprint Server Printer Inventory.
2. For a status request, the Print Interface LPD returns the status of a print job or a list of print jobs that the printer is processing.
3. For a print request, the Print Interface LPD detects the data format of the input document and performs different processing depending on the type of data:
   - If SAP OTF or ABAP data is to be printed on an AFP printer, Print Interface calls Infoprint Server Transforms to transform the data to AFP format. The SAP to AFP transform uses transform options specified in the printer definition. The administrator must configure the printer definition to use transforms and specify transform options.

- If PCL or PostScript data is to be printed on an IP PrintWay-controlled printer, Print Interface typically does not modify the data; however, the administrator can specify a filter that modifies data in the printer definition.

4. Print Interface creates an output data set on the JES spool. From the JES spool, IP PrintWay, PSF for OS/390, or JES can print the document, or IP PrintWay can send it to an e-mail destination.
5. IP PrintWay selects data sets from the JES spool and transmits them to remote printers or print servers or sends them to e-mail destinations.
6. PSF for OS/390 selects data sets from the JES spool and prints them on IBM AFP printers. The printers can be local, TCP/IP-attached, or SNA-attached.

Infoprint Server Transforms

Infoprint Server Transforms is a licensed program product (5697-F51). It provides transforms that convert data from one format to another on the z/OS system. Infoprint Server Transforms consists of the following features:

- **Transforms to AFP**

  This feature consists of the following transforms that convert data streams to monochrome Mixed Object Document Content Architecture for Presentation (MO:DCA-P) data streams, which can be printed on IBM AFP printers:

  – PCL to AFP Transform

    This transform converts Printer Control Language (PCL) 5e data streams to MO:DCA-P data streams.

  – PDF to AFP Transform

    This transform converts Adobe Portable Data Format (PDF) 1.2 data streams to MO:DCA-P data streams.

  – PostScript to AFP Transform

    This transform converts PostScript Language Level 3 data streams to MO:DCA-P data streams.

  – SAP to AFP Transform

    This transform converts (1) SAP R/3 Release 4.6C (and lower releases) Output Text Format (OTF) data streams to MO:DCA-P data streams and (2) SAP R/3 Release 4.6C (and lower releases) Advanced Business Application Programming (ABAP) data streams to line data streams. IBM AFP printers controlled by PSF can print both MO:DCA-P and line data streams.

- **Kanji AFP Print**

  This feature can be used with the PDF to AFP and PostScript to AFP transforms. It lets you print Japanese data streams that use Heisei Kaku Gothic W5 and Heisei Mincho W3 fonts, as well as embedded fonts. These two Heisei fonts, which are provided with this feature, must be installed on the z/OS system. The transform can map some other commonly used Japanese fonts, including Ryumin-Light and Gothic BBB-Medium, to these two Heisei fonts.

- **AFP to PCL Transform**

  This transform converts MO:DCA-P and line data streams to PCL 5, 5e, or 5c (color) data streams.

- **AFP to PDF Transform**

  This transform converts MO:DCA-P and line data streams to PDF 1.2 (monochrome or color) data streams.

- **AFP to PostScript Transform**

  This transform converts MO:DCA-P and line data streams to PostScript Language Level 2 (monochrome or color) data streams.

- **Coax Printer Support**
  This feature converts line data streams to Data Stream Compatibility/Data Stream Extended (DSC/DSE) and SNA Character String (SCS) data streams and in conjunction with IP PrintWay transmits the data to VTAM-controlled printers defined as VTAM LU0, LU1, or LU3 printers.

The Transforms to AFP feature and the Kanji AFP Print feature are available at no additional charge to customers who have purchased Infoprint Server. The other features are separately priced features.

A z/OS UNIX command for each transform lets users transform data in a z/OS UNIX file or MVS data set without printing it. The z/OS UNIX transform command creates an output file, which you can print or transmit to another system for viewing or printing. A filter provided for each transform lets Print Interface transform data before writing it to the JES spool. The z/OS UNIX commands and filters are: **afp2pcl**, **afp2pdf**, **afp2ps**, **pcl2afp**, **ps2afp**, **pdf2afp**, and **sap2afp**.

For more information about each transform, including its limitations, refer to *z/OS Infoprint Server User's Guide*.

Transform Manager

The Transform Manager component of Infoprint Server controls the transform daemons provided with Infoprint Server Transforms. The Transform Manager starts and stops the transform daemons using configuration information specified by the administrator. For example, the administrator can limit the number of transform daemons that are active at a time.

Figure 6 shows how the Transform Manager and Infoprint Server Transforms fit into your system. An explanation of each step follows:

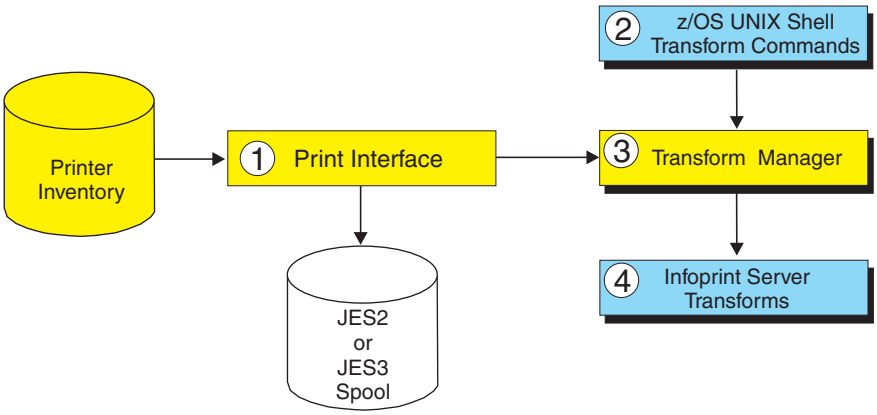


Figure 6. Transform Manager and Infoprint Server Transforms System Diagram

1. Print Interface uses transform filters provided by Infoprint Server Transforms to transform data on the z/OS system. The administrator must configure the printer definitions to use the transform filters because, by default, Print Interface does not transform data. The administrator and users can also specify transform options to control the transforms.
2. The z/OS UNIX transform commands provided with Infoprint Server Transforms let users transform data without printing it.

3. The Transform Manager manages the transform daemons and controls how many transform daemons are active at one time.

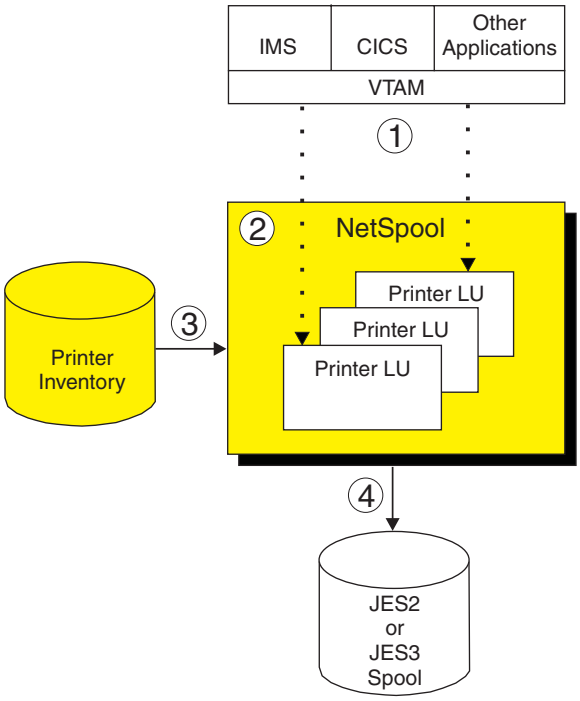
   Note: The Transform Manager does not manage the SAP to AFP transform and the Coax Printer Support feature, because they are not implemented as daemons.

4. Transforms provided with Infoprint Server Transforms perform the data transform.

NetSpool

The NetSpool component of Infoprint Server intercepts print data from VTAM applications, such as CICS and IMS, converts the data into line data, and creates output data sets on the JES2 or JES3 spool. 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.

Figure 7 shows the steps that occur from the time VTAM applications send print requests to NetSpool printer logical units (LUs) until NetSpool allocates output data sets on the JES spool. An explanation of each step follows.



· · · · · Indicates an SNA network connection.

Figure 7. NetSpool System Diagram

1. VTAM applications, such as CICS or IMS, establish communication sessions with NetSpool printer logical units (LUs) instead of with SNA-network printers. Each NetSpool printer LU must be defined to VTAM as an application logical-unit (LU).

   NetSpool can process 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

2. NetSpool runs as a VTAM application on the same or different z/OS system. 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.
3. Each NetSpool printer LU must be defined in a printer definition or in a printer pool definition in the Printer Inventory. NetSpool uses information in the printer definition to format data into lines and pages, create either a line data stream or a PCL data stream, and group the data into output data sets.
4. NetSpool dynamically allocates output data sets on the JES2 or JES3 spool using JES allocation parameters specified in the printer definition, including:
   - 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

Additional functions provided by NetSpool are:

- **Operator control**

  The system operator can control NetSpool processing from the system console and from extended MCS consoles by issuing NetSpool commands while NetSpool is running. The operator can start and stop individual printer LUs and display the status of printer LUs. To assist in managing data sets from the console, the names of the output data sets created by NetSpool identify the VTAM application that generated the print request.
- **SCS and 3270 data stream support**

  NetSpool can convert SCS and 3270 data streams into line data streams or PCL data streams. Refer to the appendices in *z/OS Infoprint Server User's Guide* for information about how NetSpool supports SCS and 3270 data streams.

  NetSpool also supports the Transparent (TRN) control in SCS data. The TRN control identifies the start of a transparent data stream.
- **Broadcasting output**

  You can print output to several printers at the same time. Also, you can print output and send it to e-mail destinations at the same time. To do this, the administrator creates a printer pool definition in the Printer Inventory. When VTAM application data is printed to the printer pool definition, NetSpool creates multiple output data sets on the JES spool.
- **Installation exits**

  NetSpool supports exits written by an installation to customize NetSpool processing. NetSpool exits let you add data to the beginning of an output data set, map graphic escape characters to other printable characters, and modify or delete transparent data in an SCS data stream.
- **Binary data support**

  The administrator can request in the printer definition that NetSpool treat the data stream as binary data. NetSpool writes binary data to the output data set as variable length records without formatting the data and without rejecting

unsupported commands, orders, or data. This function is useful if you want to pass through all data without change and without including transparent (TRN) controls.

IP PrintWay

The IP PrintWay component of Infoprint Server transmits output data sets from the JES spool to remote printers or print servers and to e-mail destinations, using one of the following transmission protocols:

- LPR: The LPR protocol is a TCP/IP protocol defined by RFC 1179. An LPD that adheres to RFC 1179 must be running in the remote printer or system.
- Direct-sockets printing: The direct sockets printing protocol is a TCP/IP protocol in which data is transmitted directly to a designated port. The remote printer or print server must support direct sockets printing.
- Internet Printing Protocol (IPP): IPP is a standard TCP/IP protocol for printing over the Internet. An IPP server must be running in the remote printer or system.
- Virtual Telecommunications Access Method (VTAM): IP PrintWay supports printing to any printer that is defined to VTAM as LU type 0, LU type 1, or LU type 3. Supported output data streams are SNA character string (SCS) and Data Stream Compatible/Data Stream Extended (DSC/DSE). The Coax Printer Support feature of Infoprint Server Transforms is required to print to VTAM-controlled printers.
- E-mail: IP PrintWay can use the z/OS UNIX sendmail function to send your print output to one or more e-mail addresses. IP PrintWay sends the output, which can be in any data format, as an e-mail attachment.

Note: IP PrintWay cannot transmit data sets larger than two gigabytes to a remote LPD. Also, depending on the IP PrintWay options the administrator selects in the printer definition, IP PrintWay might not be able to transmit data sets larger than two gigabytes to a remote printer that uses the IPP or direct-sockets printing protocol, or to e-mail destinations.

Figure 8 on page 17 shows the steps that occur from the time IP PrintWay selects output data sets from the JES spool until IP PrintWay transmits the data sets to the target destination and then deletes the data sets from the JES spool. An explanation of each step follows.

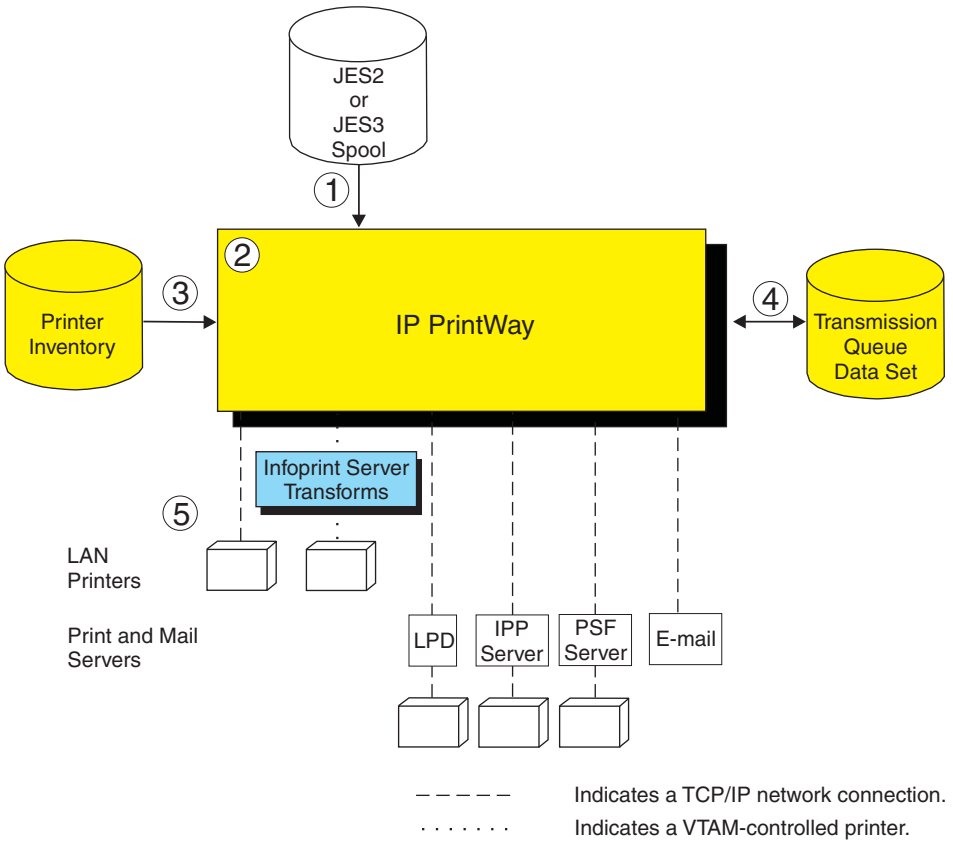


Figure 8. IP PrintWay System Diagram

1. IP PrintWay selects output data sets from the JES spool according to the JES work-selection parameters defined for the IP PrintWay FSA. For example, IP PrintWay might select all data sets in JES output class P.

   IP PrintWay can select data sets that were allocated on the JES spool by NetSpool or Print Interface, or submitted from TSO or batch applications. The data sets can contain line data, ASCII text data, or formatted data, such as PCL, PostScript, SAP, or MO:DCA-P (AFP) data.
2. IP PrintWay runs as a functional subsystem application (FSA) of JES2 or JES3. Several IP PrintWay FSAs can run in one functional subsystem address space (FSS) to handle a high volume of data; however, one PrintWay FSA can transmit data sets to multiple printers or print servers.
3. IP PrintWay uses information in the printer definition in the Printer Inventory to process data sets, select the transmission protocol (LPR, direct sockets, IPP, VTAM, or e-mail), and obtain the address of the target printer. IP PrintWay can also use the IP address of a target printer specified directly on an OUTPUT JCL statement or in an Infoprint Server job attribute.

   IP PrintWay recognizes data sets allocated on the JES spool by Print Interface and NetSpool and does not convert data from EBCDIC to ASCII or format the data if Print Interface or NetSpool has already converted data to ASCII. For other data sets, IP PrintWay can convert data between EBCDIC and ASCII, can add a header to each page, and can format data using the carriage-control characters in line data, an FCB, or pagination attributes specified in the printer definition.

   IP PrintWay can use transforms provided by Infoprint Server Transforms to convert data from one format to another. IP PrintWay calls Print Interface, if

necessary, to perform the data transform. Refer to *z/OS Infoprint Server Operation and Administration* for more information about how IP PrintWay calls Print Interface when the printer definition requests data transforms.

4. IP PrintWay maintains a transmission queue to keep track of data sets being processed. This transmission queue contains the status of each transmission, routing information, and so on. Using Infoprint Server ISPF panels, the system operator can monitor the status of transmissions, reroute data sets to another printer or e-mail destination, and change transmission options.
5. IP PrintWay transmits data sets to the target system using the protocol selected in the printer definition (LPR, direct sockets, IPP, VTAM, or e-mail). When IP PrintWay transmits data to a VTAM-controlled printer, IP PrintWay uses the Coax Printer Support feature of Infoprint Server Transforms.

   IP PrintWay can also transmit LPD options and IPP job attributes to the target LPDs and IPP servers. For example, IP PrintWay can transmit information that the LPD prints on a separator page.

   IP PrintWay can retry an unsuccessful transmission for a specified number of times at a specified interval. Retry limits and retry times can be specified in the printer definition and on an OUTPUT JCL statement. In addition to the requested retries, IP PrintWay retries an unsuccessful transmission automatically for a short period of time right after transmission.

Additional functions provided by IP PrintWay include:

- **Retaining jobs on the JES spool**

  After successfully transmitting each data set, or after completing the requested number of transmission attempts, IP PrintWay can retain the data set on the JES spool forever or for a period of time. Retention periods can be specified in the printer definition or on an OUTPUT JCL statement.
- **Printer selection using an OUTPUT JCL statement**

  On an OUTPUT JCL statement, a user can select the printer definition by specifying either (1) the name of the printer definition in the FSSDATA parameter or (2) the DEST, CLASS, or FORMS parameter (or a combination of these parameters) associated with the printer definition.

  Users can also specify the IP address for the target printer directly on the OUTPUT JCL statement, thereby eliminating the need for the administrator to create a printer definition for each printer in the Printer Inventory.
- **Accounting**

  For each data set processed, IP PrintWay writes a System Management Facility (SMF) type-6 record, which includes the number of bytes transmitted and the IP address of the target system.
- **Installation exits**

  IP PrintWay supports exits written by an installation to customize IP PrintWay processing. For example, an exit can change the IP address of the remote printer, add separator pages, modify SMF accounting records, and notify users of processing events.
- **Maintaining transmission order**

  IP PrintWay preserves the order of the data sets on the JES spool when transmitting data sets. IP PrintWay retains this order even if the transmission of the data sets must be retried. If a JES output group contains more than one output data set, IP PrintWay acquires all of the data sets in the output group before transmitting any of them and can transmit these data sets to the printer as a single file. Although the data sets are a single file, each data set starts printing on a new page.

- **Transmitting printer commands**

  The administrator can specify printer commands in the printer definition for IP PrintWay to send to the printer before or after the data to be printed. Printer commands can be used to change fonts or switch between simplex and duplex printing.

SNMP Subagent

The Infoprint Server SNMP subagent, in conjunction with support provided by PSF for OS/390 and the z/OS SNMP agent, lets administrators monitor printer characteristics (such as the printer resolution) and printer status (such as paper jams) for any printer controlled by PSF for OS/390. Also, administrators can be notified as soon as an intervention situation (such as a paper jam) occurs on the printer. This support does not let administrators change any printer characteristics.

To monitor PSF printers, the z/OS SNMP agent must be configured and an SNMP manager, such as IBM Network Printer Manager (NPM) for the Web, must be installed.

You can monitor PSF printers that do not contain internal SNMP agents, such as the IBM 3900 printer, and also PSF printers that have internal SNMP agents but are not TCP/IP-attached to PSF. You can also monitor PSF TCP/IP-attached printers that contain internal SNMP agents; however, consider defining PSF printers that have internal SNMP agents directly to the SNMP manager. When you define a printer directly to the SNMP manager, you can also monitor printer statistics and change some printer characteristics. Refer to the documentation for your printers to determine if they have internal SNMP agents.

Figure 9 on page 20 shows how the SNMP subagent fits into your system. An explanation of each step follows:

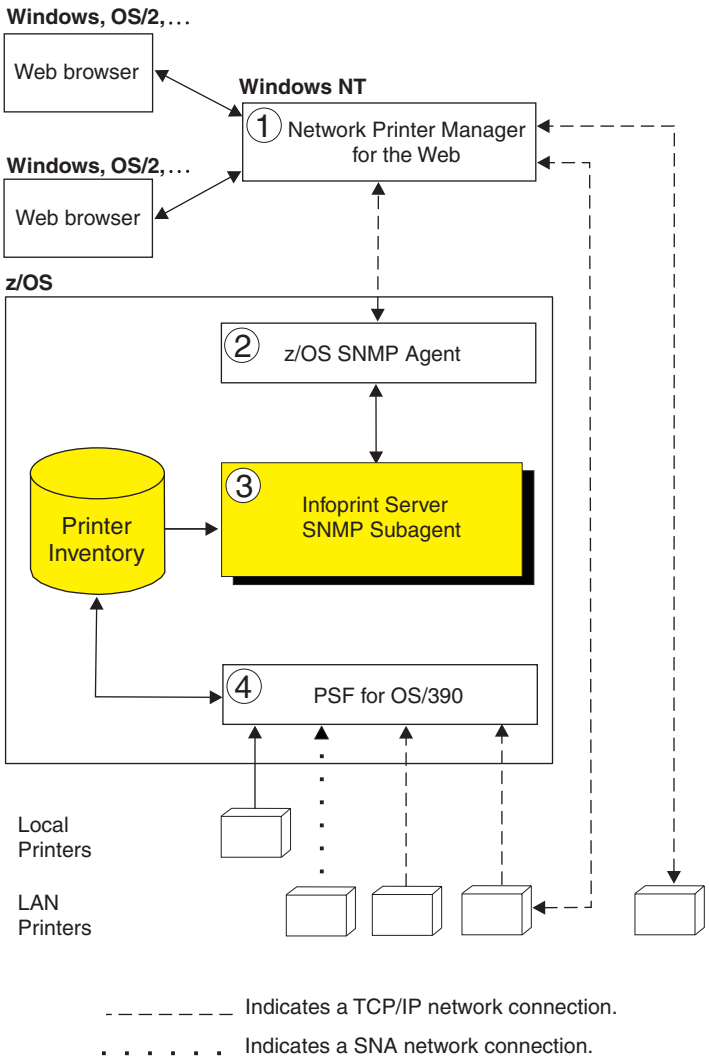


Figure 9. SNMP Subagent System Diagram

1. To monitor PSF printers, you must install an SNMP manager. You can install any SNMP manager that communicates with an SNMP agent that supports the general printer MIB (defined in RFC 1759) and has implemented support for multiple printers defined with one IP address.

   One SNMP manager that provides the required support is IBM Network Printer Manager (NPM) for the Web, which is shown in Figure 9. You can download NPM from the IBM Printing Systems Division Web site. Administrators can use a Java-enabled version of Netscape Navigator or Microsoft® Internet Explorer to monitor PSF printers, while the NPM server runs on a Windows NT® system. Note that NPM limits the number of PSF printers an administrator can monitor at the same time; see the NPM online help for the maximum number of printers NPM lets you monitor.

   As shown in the figure, the SNMP manager (NPM) can also communicate directly with any TCP/IP-attached printer that contains an internal SNMP agent; this printer can be controlled by PSF, but this is not necessary.
2. The SNMP agent, part of the z/OS Communications Server, communicates with the SNMP manager and with the Infoprint Server SNMP subagent.
3. The Infoprint Server SNMP subagent communicates with PSF for OS/390 through the Printer Inventory. PSF for OS/390 stores printer characteristics and

printer status in the Printer Inventory for any printer that has SNMP-reporting enabled. (The administrator enables SNMP reporting in the FSA definition for the PSF printer).

The SNMP subagent transmits the information stored by PSF for OS/390 to the SNMP manager through the z/OS SNMP agent. The SNMP subagent also notifies the SNMP manager immediately when PSF for OS/390 detects a change to printer characteristics or printer status.

4. PSF for OS/390 obtains printer characteristics and printer status from any PSF-controlled printer. The PSF printer can be channel-attached, TCP/IP-attached, or SNA-attached; however, the PSF printer cannot be attached using the Download for OS/390 feature of PSF.

   If a PSF-controlled printer is shared with another printing application, and is not connected to PSF when an intervention required situation occurs, PSF cannot report the change in printer status.

Chapter 2. Customization Roadmap

This chapter can help you determine which Infoprint Server components you must customize to use the different functions that Infoprint Server provides. Table 2 lists the functions provided by Infoprint Server, the Infoprint Server components you need to customize, and the sections that describe the customization tasks.

Notice that you must customize the Printer Inventory Manager component in order to use any function provided by Infoprint Server. You can customize other components later as your installation decides to use additional functions. If you plan to use more than one Infoprint Server function, customize all related components. For example, to print from VTAM applications to TCP/IP-attached printers, customize the Printer Inventory Manager, NetSpool, and IP PrintWay.

Table 2. Summary of Infoprint Server Functions and Corresponding Infoprint Server Components

Infoprint Server Function	Components	See Page:
Receive print requests from the following sources, transform data from one data format to another, and allocate output data sets on the JES spool: • Clients that use LPR to LPD protocol • Clients that use Internet Printing Protocol (IPP) • Windows clients that use Server Message Block (SMB) protocol • z/OS UNIX **lp**, **lpstat**, and **cancel** commands • The AOPPRINT JCL procedure • Any Windows application that supports printing • Infoprint Server Application Programming Interface • Batch jobs that specify the Print Interface subsystem on a DD statement.	Printer Inventory Manager Print Interface Infoprint Server Transforms and Transform Manager (to transform data) Infoprint Port Monitor for Windows (optional)	25 49 83 237
Receive print requests from VTAM applications (such as CICS and IMS) and allocate output data sets on the JES spool.	Printer Inventory Manager NetSpool	25 143
Transform data from one format to another using z/OS UNIX commands: **afp2pcl**, **afp2pdf**, **afp2ps**, **pcl2afp**, **ps2afp**, **pdf2afp** and **sap2afp**.	Printer Inventory Manager Infoprint Server Transforms and Transform Manager	25 83
Transmit output data sets on the JES spool to a remote system using one of the following transmission protocols: • LPR to LPD protocol • Internet Printing Protocol (IPP) • Direct sockets printing • VTAM • E-mail Transform data from one format to another if necessary.	Printer Inventory Manager IP PrintWay Print Interface (to transform data) Infoprint Server Transforms and Transform Manager (to transform data)	25 163 49 83
View printer characteristics and status of PSF printers.	Printer Inventory Manager SNMP subagent	25 139
Store PSF for OS/390 system information in the Printer Inventory.	Printer Inventory Manager	25

© Copyright IBM Corp. 1995, 2002

Chapter 3. Customizing the Printer Inventory Manager

This chapter describes how to customize the Printer Inventory Manager component of Infoprint Server. You must customize the Printer Inventory Manager to use any functions provided by Infoprint Server and Infoprint Server Transforms.

The Printer Inventory Manager maintains the Printer Inventory and provides Infoprint Server ISPF panels and the Printer Inventory Definition Utility program (PIDU) for the administrator to create and manage entries in the Printer Inventory. See "Printer Inventory Manager" on page 3 for an overview of the Printer Inventory Manager and how this component fits into your system.

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

The following table shows the tasks for customizing the Printer Inventory Manager. Required tasks are required by all installations; optional tasks are required only if the listed condition applies to your installation.

Task	Condition	See Page:
Identifying Software Requirements	Required	25
Enabling the Infoprint Server Feature	Optional: If the Infoprint Server feature was not in your original z/OS order	26
Making LE Run-time Libraries Available	Optional: If required libraries are not in LNKLST	27
Understanding the Infoprint Server HFS Directories	Optional	26
Creating the Infoprint Server Configuration File (**aopd.conf**)	Optional: To change default configuration	28
Setting Environment Variables for the Printer Inventory Manager	Required	31
Establishing Security for the Printer Inventory and Operator Commands	Required	33
Customizing Startup and Shutdown Commands and Procedures	Optional: To change default environment variables or to start and stop Infoprint Server with JCL procedures	35
Enabling Infoprint Server ISPF Panels	Required	42
Backing Up and Restoring the Printer Inventory	Optional: To schedule backup	44

Related task: After you customize the Printer Inventory Manager, the operator must start the Printer Inventory Manager daemon with the **aopstart** command or AOPSTART procedure. Refer to *z/OS Infoprint Server Operation and Administration* for information.

Identifying Software Requirements

To use the Printer Inventory Manager, the following additional software beyond the z/OS base elements is required:

© Copyright IBM Corp. 1995, 2002

- z/OS Security Server (RACF) feature or another access control manager with equivalent function
- If you want to print output from Infoprint Server on IBM AFP printers, PSF V3R1.0 for OS/390 (5655-B17) or a later release
- If you want PSF to store information in the Printer Inventory:
  - PSF V3R1.0 with PTF UW63353, or a later release
  - Infoprint Server PTF UW83615

Enabling the Infoprint Server Feature

If the Infoprint Server priced feature was not included in your original z/OS order, you must dynamically enable Infoprint Server. To do this, you must add an entry for Infoprint Server to the active IFAPRDxx member of SYS1.PARMLIB. The feature name is INFOPRINT SERVER, and the product identifier is 5694-A01. For information about dynamic enablement, refer to *z/OS and z/OS.e Planning for Installation*.

Understanding the Infoprint Server HFS Directories

The Printer Inventory Manager uses the following hierarchical file system (HFS) directories:

- The **/etc/Printsrv** directory, which is the default location for Infoprint Server configuration files.
- The **/var/Printsrv** directory, which is the default location for the Printer Inventory files.

/etc/Printsrv Directory

The **/etc/Printsrv** directory contains all Infoprint Server configuration files. This directory is created automatically with the appropriate permissions when you install Infoprint Server.

Recommendation: Do not change the owner or permissions of the **/etc/Printsrv** directory. For a secure environment, this directory should be:
- Owned by UID of 0.
- Writable only by users with an effective UID of 0.

Related task: You can create Infoprint Server configuration files in a directory other than the **/etc/Printsrv** directory. If you do so, specify the location of the configuration files in Infoprint Server environment variables. Appendix C, “Environment Variables” on page 267 summarizes all Infoprint Server environment variables.

/var/Printsrv Directory

The **/var/Printsrv** directory contains the Printer Inventory. The **aopsetup** shell script creates the **/var/Printsrv** directory with the appropriate permissions.

Infoprint Server creates the Printer Inventory files automatically the first time the administrator uses the Infoprint Server ISPF panels or the Printer Inventory Definition Utility (PIDU) to create objects in the Printer Inventory, such as printer definitions. The Printer Inventory files also contain objects that the administrator does not create. For example, Print Interface creates objects for each job processed; these job objects are deleted when the SYSOUT data sets to which they correspond are deleted from the JES spool.

The Printer Inventory is comprised of the following files:

- **master.db**
- **jestoken.db**

The **master.db** and **jestoken.db** files are data base files optimized for rapid direct access to objects. As you add objects to the Printer Inventory, these files increase in size. When you remove objects, the files do not decrease in size because the Printer Inventory Manager simply designates as available the space within the file that had been occupied by the removed objects. When you add objects in the future, the Printer Inventory Manager uses available space within the files; the files increase in size only when they do not contain sufficient available storage. So, the size of each file can be characterized as a high-water mark.

The data transforms provided by Infoprint Server Transforms create subdirectories in the **/var/Printsrv** directory and create temporary files in these subdirectories.

Recommendations:

- Do not change the owner or permissions of the **/var/Printsrv** directory after it is created. For a secure environment, this directory should be:
  - Owned by UID of 0
  - Readable and writable only by users with an effective UID of 0 or members of the AOPADMIN group
  - Executable by everyone
- IBM requires that you mount a separate file system at the **/var** mount point and create the **/var/Printsrv** directory in that file system. **/var** is a symbolic link to a system-specific HFS data set. For information on creating the HFS data set for **/var**, refer to *z/OS UNIX System Services Planning*.
- Because the transforms provided by Infoprint Server Transforms create temporary files in subdirectories of the **/var/Printsrv** directory, ensure that the **/var** file system has enough space to contain all data sets that are transformed at the same time.

Related tasks:

- You can use a different directory name for the **/var/Printsrv** directory. If you do so, specify the directory name in the **base-directory** attribute in the **aopd.conf** configuration file, as described in “Creating the Infoprint Server Configuration File (**aopd.conf**)” on page 28, *before* you run **aopsetup**.
- Run the **aopsetup** shell script to create the **/var/Printsrv** directory. See “Establishing Security for the Printer Inventory and Operator Commands” on page 33 for information.

Making LE Run-time Libraries Available

Infoprint Server requires that the Language Environment (LE) run-time library (CEE.SCEERUN) and the C^{++} run-time library (CBC.SCLBDLL) reside in an APF-authorized library and be available to Infoprint Server z/OS UNIX commands, Infoprint Server ISPF panels, Infoprint Server utilities such as PIDU and AOPBATCH, the AOPPRINT JCL procedure, NetSpool, and IP PrintWay.

IBM recommends that you add the SCEERUN and SCLBDLL data sets to the system LNKLST. However, if you do not add the SCEERUN and SCLBDLL data sets to LNKLST, then you can take the following steps:

- Add the SCEERUN and SCLBDLL data sets to the STEPLIBLIST facility of the BPXPRMxx PARMLIB member.

- Add the following statement to the **/etc/rc** file:

  ```
  export STEPLIB=hlq.SCEERUN:hlq.SCLBDLL
  ```

  For example, if you use the standard high-level qualifiers, add this statement:

  ```
  export STEPLIB=CEE.SCEERUN:CBC.SCLBDLL
  ```

- Remove the following statements from the **/etc/profile** file:

  ```
  if [-z "$STEPLIB"] && tty -s;
  then
    export STEPLIB=none
    exec sh -L
  fi
  ```

  Replace these statements in the **/etc/profile** file with the following statement:

  ```
  export STEPLIB=hlq.SCEERUN:hlq.SCLBDLL
  ```

  For example, if you use the standard high-level qualifiers, add this statement:

  ```
  export STEPLIB=CEE.SCEERUN:CBC.SCLBDLL
  ```

- Uncomment and edit the following statement in the **aopstart** EXEC:

  ```
  /*n=n+1;env.n='STEPLIB=hlq.SCEERUN:hlq.SCLBDLL'          */
  ```

  For example, if you use the standard high-level qualifiers, code this statement:

  ```
  n=n+1;env.n='STEPLIB=CEE.SCEERUN:CBC.SCLBDLL'
  ```

  See “Editing the **aopstart** EXEC” on page 36 for information.

- Add the SCEERUN and SCLBDLL data sets to the TSO logon procedures for administrators who use Infoprint Server ISPF panels; see “Defining Libraries in the TSO Logon Procedure” on page 42 for information.
- Add the SCEERUN and SCLBDLL data sets to JCL that invokes AOPBATCH. The AOPSTART and AOPPRINT JCL procedures in SYS1.PROCLIB use AOPBATCH.
- Add the SCEERUN and SCLBDLL data sets to the NetSpool and IP PrintWay startup procedures; see “Customizing the NetSpool Startup Procedure” on page 144 and “Customizing the IP PrintWay Startup Procedure” on page 181 for more information.

Creating the Infoprint Server Configuration File (aopd.conf)

The Infoprint Server configuration file, **aopd.conf**, lets you customize the Printer Inventory Manager and other components of Infoprint Server. This file is optional; if the configuration file does not exist or if an attribute in the configuration file is omitted, default values are used.

This section describes the attributes in the **aopd.conf** configuration file that are used by the Printer Inventory Manager. See Appendix B, “Infoprint Server Files” on page 245 for a summary of all the attributes you can specify in **aopd.conf**.

To create and edit the configuration file, use the following z/OS UNIX commands:

1. Switch to an effective UID of 0:

   ```
   su
   ```

   To use the **su** command, you must be permitted to the BPX.SUPERUSER profile in the FACILITY class within RACF.

2. Copy the sample configuration file, **/usr/lpp/Printsrv/samples/aopd.conf**, to **/etc/Printsrv/aopd.conf**:

   ```
   cp /usr/lpp/Printsrv/samples/aopd.conf  /etc/Printsrv/aopd.conf
   ```

   If you copy the sample configuration file to a different location, specify the full path name of the configuration file in the AOPCONF environment variable.

   Tip: You can use the sample job provided in SYS1.SAMPLIB(AOPCPETC) to copy all Infoprint Server sample configuration files, including the Infoprint Server configuration file, from **/usr/lpp/Printsrv/samples** to the **/etc/Printsrv** directory. Review the comments in the sample job to determine which configuration files your installation requires, and then run the job to create the configuration files. We recommend that a user with an effective UID of 0 run this job.
3. Edit the configuration file using your preferred editor, for example:

   ```
   oedit /etc/Printsrv/aopd.conf
   ```

   See “Guidelines” for the syntax rules for editing the file.
4. If you edit the configuration file while Infoprint Server is running, stop and restart all Infoprint Server daemons to pick up the changes:

   ```
   aopstop
   aopstart
   ```

   If you change the **inventory** attribute in the configuration file, also stop and restart other components and products that use this name, including NetSpool, IP PrintWay, and PSF for OS/390.

Recommendation: For a secure environment, do *not* change the owner or permissions of the configuration file. The file should be owned by UID 0 and be writable only by a user with a UID of 0. To switch to an effective UID of 0, you can use the z/OS UNIX **su** command if you are permitted to the BPX.SUPERUSER profile in the FACILITY class within RACF.

Syntax Rules: When you edit this file, follow these syntax rules:
- Use lowercase characters for the attributes and use uppercase or lowercase characters for the values, with blank characters before or after the equal sign, if desired.
- 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.

To customize the Printer Inventory Manager, specify the following attributes in the **aopd.conf** configuration file:

ascii-codepage = *codepage*

The name of an ASCII code page supported by z/OS. For code page names, refer to *z/OS C/C++ Programming Guide*. The Printer Inventory Manager, Print Interface, and NetSpool use this code page in the following ways:

- The Printer Inventory Manager displays this code page in the **Printer code page** field in ISPF panels for new IP PrintWay printer definitions that use the LPR, direct sockets, or IPP protocol to transmit data sets to the remote system.

- Print Interface uses this code page as the default source (document) code page when it converts data streams from ASCII to EBCDIC. A document code page specified in the printer definition or by the job submitter overrides this value.
- Print Interface uses this code page as the target (printer) code page when it converts PostScript documents that were created with a non-ASCII code page.
- NetSpool uses this code page as the default target (printer) code page when it converts data streams from EBCDIC to ASCII. A printer code page specified in the printer definition overrides this value.

Default: `ascii-codepage = ISO8859-1`

base-directory = *path*
The directory path in which the Printer Inventory Manager creates Printer Inventory files. Other components of Infoprint Server also create files in this directory.

Attention: Do *not* change this attribute while any Infoprint Server daemons are running.

Default: `base-directory = /var/Printsrv`

ebcdic-codepage = *codepage*
The name of an EBCDIC code page supported by z/OS. For code page names, refer to *z/OS C/C++ Programming Guide*. The Printer Inventory Manager, Print Interface, and NetSpool use this code page in the following ways:

- The Printer Inventory Manager displays this code page in the **Printer code page** field in ISPF panels for new PSF for OS/390 printer definitions, General printer definitions, and IP PrintWay printer definitions that use the VTAM or e-mail protocol to transmit data sets to the remote system.
- Print Interface uses this code page as the source (document) code page when Print Interface cannot determine the user locale for the **lp** command.
- NetSpool uses this code page as the default source (document) code page when it converts data streams from EBCDIC to ASCII. A document code page in the printer definition overrides this value.

Default: `ebcdic-codepage = IBM-1047`

inventory = *inventory*
The name of the Printer Inventory and the Print Interface subsystem. Specify exactly four uppercase or lowercase letters or numbers. This name is case sensitive.

If you use the Print Interface subsystem, the first character must be an uppercase letter; it *cannot* be a lowercase letter or numeric. The job submitter specifies this name in the SUBSYS parameter on the DD JCL statement.

If you start NetSpool or IP PrintWay, specify this name on the EXEC statement of the startup procedure. Also, specify this name in the PSF for OS/390 startup procedure if you want PSF for OS/390 to use information that is specified in PSF FSS and FSA definitions in the Infoprint Server Printer Inventory.

If you change the name of the Printer Inventory and Print Interface subsystem after the Printer Inventory Manager has been started, you must stop and restart all Infoprint Server daemons. You must also stop and restart all other components and products that specify this Inventory name in their startup procedures. You must stop and restart NetSpool and IP PrintWay. If the PSF startup procedure specifies the name of the Printer Inventory, you must also stop and restart PSF for OS/390.

Default: `inventory = AOP1`

start-daemons = { }

Identifies the daemons that the **aopstart** command starts. To start only the Printer Inventory Manager daemon, simply code braces without specifying any daemons. The Printer Inventory Manager daemon always starts when you use the **aopstart** command, regardless of which daemons you specify in this attribute. By default, the Printer Inventory Manager starts, as well as the Print Interface LPD.

Default: `start-daemons = {lpd}`

Setting Environment Variables for the Printer Inventory Manager

Infoprint Server uses environment variables to define its environment within the z/OS system. Refer to *z/OS UNIX System Services User's Guide* for a description of environment variables.

This section describes only those environment variables that affect the Printer Inventory Manager. Other environment variables affect other components of Infoprint Server, such as Print Interface. If you want to set all environment variables that affect Infoprint Server at the same time, see Appendix C, "Environment Variables" on page 267 for a summary of all the environment variables used by Infoprint Server.

Infoprint Server environment variables can be set in the following two locations:

- In the **aopstart** EXEC: The Printer Inventory Manager, and other Infoprint Server daemons, use environment variables specified in this file. See "Editing the **aopstart** EXEC" on page 36 for information about how to edit this file.
- In the **/etc/profile** file: The z/OS UNIX printing commands and other commands, such as **lp**, **pidu**, and **ps2afp**, use environment variables specified in this file.

To edit the **/etc/profile** file, you can use the TSO/E OEDIT command or the z/OS UNIX **oedit** command. To set and export an environment variable, use the z/OS UNIX **export** command. For example, if you installed Infoprint Server libraries in the default locations, add the following commands to the **/etc/profile** file:

```
export LIBPATH=/usr/lpp/Printsrv/lib:$LIBPATH
export NLSPATH=/usr/lpp/Printsrv/%L/%N:$NLSPATH
export PATH=/usr/lpp/Printsrv/bin:$PATH
```

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

```
echo $PATH
```

Table 3 on page 32 summarizes the variables that affect the Printer Inventory Manager, including the **pidu** command. (The **pidu** command lets the administrator use the Printer Inventory Definition Utility to create objects, such as printer definitions, in the Printer Inventory.) This table indicates whether you need to set the variable in the **/etc/profile** file, in the **aopstart** EXEC, or in *both* locations.

Required variables must be set; **optional** variables must be set only if the default value is not suitable for your installation.

Table 3. Where to Define Environment Variables Used by the Printer Inventory Manager

Set this environment variable	In /etc/profile file	In aopstart EXEC
AOPCONF	Optional	Optional
LIBPATH	Required	Optional
NLSPATH	Required	Optional
PATH	Required	Optional

To edit the **aopstart** EXEC, see "Editing the **aopstart** EXEC" on page 36 for details.

If you change the environment variables while any Infoprint Server daemons are running, the settings do not change for the current login shell, daemons, or other processes already started.

These environment variables affect the behavior of the Printer Inventory Manager:

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.

Default: `/etc/Printsrv/aopd.conf`

LIBPATH The path used to find dynamic link library (DLL) files. This environment variable is *required*. 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.

Default: `/usr/lpp/Printsrv/lib`
(This default applies only to the **aopstart** EXEC.)

NLSPATH The path of directories that contain message catalogs. This environment variable is *required*. 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:

English **/usr/lpp/Printsrv/En_US/%N**

Japanese **/usr/lpp/Printsrv/Ja_JP/%N**

%L represents the value of the LANG environment variable. **%N** is the catalog filename.

Default: `/usr/lpp/Printsrv/%L/%N:/usr/lpp/Printsrv/En_US/%N:/usr/lib/nls/msg/%L/%N`
(This default applies only to the **aopstart** EXEC.)

PATH The path used to locate executables. This environment variable is *required*. If you installed Infoprint Server executables in the default directory, add **/usr/lpp/Printsrv/bin** to the existing values. Be sure to add the directory *before* **/bin** in the PATH environment variable to ensure that the Infoprint Server versions of the **lp**, **lpstat**, and **cancel** commands are invoked.

Default: `/usr/lpp/Printsrv/bin:/bin`
(This default applies only to the **aopstart** EXEC.)

Establishing Security for the Printer Inventory and Operator Commands

This section describes how to customize the security function in Infoprint Server. The security function in Infoprint Server lets you determine which users can read and update the Printer Inventory and start and stop Infoprint Server daemons. You *must* follow the steps in this section; otherwise, only users with a UID of 0 can read and update the Printer Inventory and start and stop Infoprint Server daemons.

You can use the Resource Access Control Facility (RACF) or another program that follows system authorization facility (SAF) protocol to establish security. For more information about RACF, refer to *z/OS Security Server RACF Security Administrator's Guide*.

To establish security, follow these steps.

Tip: The sample CLIST in SYS1.SAMPLIB(AOPRACF) contains all the RACF commands that are described in this section. If you use AOPRACF, you must still specify the z/OS UNIX commands in steps 5b, 5c, and 7. You must modify AOPRACF before you run it.

1. Define a group to RACF for Infoprint Server administrators.

   An Infoprint Server administrator can read and update the Printer Inventory. The suggested group name for administrators is AOPADMIN; however, you can use any name. The group profile must have an OMVS segment and an OMVS group identifier (GID). For example, the following RACF command defines group AOPADMIN; for *group-identifier*, specify an integer that is different from other GIDs in your installation:

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

2. Define a group to RACF for Infoprint Server operators.

   Infoprint Server operators can start and stop Infoprint Server daemons 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 group identifier (GID). For example, the following RACF command creates group AOPOPER; for *group-identifier*, specify an integer that is different from other GIDs in your installation.

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

   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.

3. Define a resource profile to RACF named AOPADMIN in the FACILITY class.

   The name of the resource profile must be AOPADMIN. For example, the following RACF commands create resource AOPADMIN:

   ```
   RDEFINE FACILITY (AOPADMIN) UACC(NONE)
   SETROPTS CLASSACT(FACILITY) RACLIST(FACILITY)
   SETROPTS RACLIST(FACILITY) REFRESH
   ```

4. Give the Infoprint Server administrator group READ access to the AOPADMIN resource profile.

You can permit group AOPADMIN to the AOPADMIN resource profile or you can permit each user separately. Users must have READ (or higher) access to the AOPADMIN resource profile in order to view or update the Printer Inventory. For example, the following RACF command permits the AOPADMIN group to the AOPADMIN resource in the FACILITY class:

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

5. Define Infoprint Server administrators and operators to RACF as z/OS UNIX users.

   Refer to *z/OS UNIX System Services Planning* for complete information about how to define z/OS UNIX users to RACF. Following is a summary of the required steps:

   a. For each user, you must establish an OMVS segment, define the user's home directory, and set the user's default group to a group that has an OMVS segment with a group identifier (GID). Use either the ADDUSER command to define a new user ID or the ALTUSER command to modify an existing user ID.

      In the following ALTUSER example, for *userid*, specify an existing user ID; for *user-identifier*, specify an integer that is different from other UIDs in your installation; and for *group*, specify AOPOPER, AOPADMIN, or any another group that has an OMVS GID. The user ID must be connected to the default group.

      ```
      ALTUSER userid OMVS(UID(user-identifier) HOME('/u/userid')
         PROGRAM('/bin/sh')) DFLTGRP(group)
      ```

      Notes:

      1) The user's default group must have a GID defined so that the user can use z/OS UNIX.
      2) See "Creating a User ID to Assign to the AOPSTART and AOPSTOP Procedures" on page 38 for instructions about how to define a user ID for use with the AOPSTART and AOPSTOP started procedures.

   b. Switch to an effective UID of 0:

      ```
      su
      ```

      To use the **su** command, you must be permitted to the BPX.SUPERUSER profile in the FACILITY class within RACF.

   c. 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 automounter facility to create the home directory and set permissions. For example:

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

   d. Use the z/OS UNIX **chown** command to make the user the owner of the home directory:

      ```
      chown -R user /u/userid
      ```

6. Connect Infoprint Server administrators and operators to the appropriate group.

   Use the RACF CONNECT command to connect Infoprint Server administrators to the AOPADMIN group. Use the RACF CONNECT command to connect Infoprint Server operators to the AOPOPER group. In the following examples, specify an existing user ID for *userid*:

   ```
   CONNECT (userid) GROUP(AOPADMIN)
   CONNECT (userid) GROUP(AOPOPER)
   ```

If you did not define a separate group for Infoprint Server operators in step 2, connect the operators to the group you defined for Infoprint Server administrators (for example AOPADMIN).

7. Run the **aopsetup** shell script to create the **/var/Printsrv** directory and 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 did not define a separate group for Infoprint operators in step 2, specify the group name for Infoprint Server administrators (for example AOPADMIN) in both arguments.

   Guidelines:

   a. Run **aopsetup** from a user ID with a UID of 0. To switch to an effective UID of 0, you can use the z/OS UNIX **su** command if you are permitted to the BPX.SUPERUSER profile in the FACILITY class within RACF.
   b. If you want to change the **/var/Printsrv** directory name, specify the new name in the **base-directory** attribute in the **aopd.conf** configuration file, as described in "Creating the Infoprint Server Configuration File (**aopd.conf**)" on page 28, *before* you run **aopsetup**.

Customizing Startup and Shutdown Commands and Procedures

To start and stop Infoprint Server daemons, including the Printer Inventory daemon, the operator can use one or the following methods:

- The AOPSTART and AOPSTOP JCL procedures
- The z/OS UNIX **aopstart** and **aopstop** commands

In addition, you can use z/OS automation tools or the **/etc/rc** shell script to start Infoprint Server daemons automatically during sytem initialization.

Infoprint Server daemons include:

- **aopd**, the Printer Inventory Manager daemon: This is the main Infoprint Server daemon, which manages the Printer Inventory.
- **aopippd**, the IPP server daemon: This daemon supports printing from remote clients that use the Internet Printing Protocol (IPP).
- **aoplpd**, the line printer daemon (LPD): This daemon supports printing from remote clients that use the TCP/IP line printer requester (LPR) protocol.
- **aopsnmpd**, the SNMP subagent daemon: This daemon provides status information for PSF-controlled printers to the z/OS SNMP agent for viewing with an SNMP manager.
- **aopxfd**, the Transform Manager daemon: This daemon manages other transform daemons. The Transform Manager daemon starts and stops transform daemons that are configured in the **aopxfd.conf** configuration file.
- **aopsapd**, the SAP Callback daemon

  This daemon handles callback notification for the SAP Output Management System (OMS). Infoprint Server starts this daemon automatically when it is needed.
- **aopsubd**, the Print Interface subsystem daemon

This daemon processes output data sets that request the Print Interface subsystem on the SUBSYS parameter of the DD statement.

The following table summarizes the customization tasks related to setting up your startup and shutdown procedures. Required tasks are required by all installations; optional tasks are required only if the listed condition applies to your installation.

Task	Condition	See Page:
Editing the **aopstart** EXEC	Optional: To change default environment variables	36
Creating a User ID to Assign to the AOPSTART and AOPSTOP Procedures	Optional: To use the AOPSTART and AOPSTOP JCL procedures	38
Defining the AOPSTART and AOPSTOP Procedures to RACF	Optional: To use the AOPSTART and AOPSTOP JCL procedures	39
Editing the AOPSTART Startup Procedure	Optional: To use the AOPSTART JCL procedure and change default settings	39
Editing the AOPSTOP Shutdown Procedure	Optional: To use the AOPSTOP JCL procedure and change default settings	40
Starting Infoprint Server Daemons Automatically	Optional: To start Infoprint Server daemons automatically	42

Editing the aopstart EXEC

The **aopstart** command is a REXX EXEC. It sets environment variables used by Infoprint Server daemons and starts the Printer Inventory daemon and any other daemons specified in the **start-daemons** attribute in the Infoprint Server configuration file (**aopd.conf**).

The **aopstart** EXEC provided with Infoprint Server is suitable for most installations. However, you can edit the EXEC to change the environment variables for your installation. Table 4 summarizes the conditions for which you need to edit the **aopstart** EXEC and the environment variables you need to change or specify. This table also refers you to the section that contains more information about the environment variable.

Table 4. When to Edit the aopstart EXEC

If this condition applies:	Specify this variable:	See Page:
Infoprint Server files were *not* installed in the default directory: **/usr/lpp/Printsrv**	LIBPATH, NLSPATH, PATH CLASSPATH[1]	31 69
The Infoprint Server configuration file is *not* in the default location: **/etc/Printsrv/aopd.conf**	AOPCONF	31
The LE and C++ libraries are not in the system LNKLST	STEPLIB	27
You run the Print Interface LPD, and you have set up multiple TCP/IP stacks on your system	_BPXK_SETIBMOPT_TRANSPORT	63

Table 4. When to Edit the aopstart EXEC (continued)

If this condition applies:	Specify this variable:	See Page:
You run Infoprint Server Transforms, and the transform configuration file is *not* in the default location: **/etc/Printsrv/aopxfd.conf**	AOPXFD_CONF	91
You run the Infoprint Server SAP Output Management System (OMS)	LIBPATH	75
You run the SAP OMS, and the SAP Callback configuration file is *not* in the default location: **/etc/Printsrv/aopsapd.conf**	AOPSAPD_CONF	75
You run the SAP to AFP transform, and transform resources are *not* in the default location: **/usr/lpp/Printsrv/sap2afp**	AOP_SAP2AFP_RESOURCES	103
You run the Print Interface IPP server, and Java was *not* installed in the default directory: **/usr/lpp/java/IBM/J1.3**	JAVA_HOME	69
You do *not* want Infoprint Server daemon trace files to be written to the default directory: **/var/Printsrv/trace**	AOPTRACEDIR	[2]

1. If all Infoprint Server files were installed in the same path, you can specify the **install_path** variable in the **aopstart** REXX EXEC instead of changing each of the environment variables. See the comments in the **aopstart** EXEC for more information.
2. Refer to *z/OS Infoprint Server Messages and Diagnosis* for information about AOPTRACEDIR.

To edit the **aopstart** EXEC, use the following z/OS UNIX commands:

1. Switch to an effective UID of 0:

   ```
   su
   ```

   To use the **su** command, you must be permitted to the BPX.SUPERUSER profile in the FACILITY class within RACF.
2. Copy the default **aopstart** file to another directory, for example to directory **/usr/sbin/**. You must copy the file because you should not modify the directory that contains Infoprint Server executable files.

   ```
   cp /usr/lpp/Printsrv/bin/aopstart /usr/sbin/
   ```
3. Change to the directory where you copied **aopstart**:

   ```
   cd /usr/sbin
   ```
4. Set the group of the **aopstart** file to AOPOPER or to the RACF group you defined for Infoprint Server operators:

   ```
   chgrp AOPOPER aopstart
   ```

   If you have already run the **aopsetup** shell script, you can omit this step.
5. Edit the new **aopstart** file using your preferred editor, for example:

   ```
   oedit aopstart
   ```

   Change or specify variables that are not suitable for your installation. See Table 4 on page 36 to determine which environment variables you need to change or specify. See the comments in the **aopstart** EXEC for additional instructions.

6. Change the permissions for the file. Set the **set-uid-flag** on; make the file writable by only UID 0, readable by everyone, and executable only by UID 0 and the AOPOPER group. For example:

   ```
   chmod 4754 aopstart
   ```

   Note: Perform this step after you edit the file, because the **set-uid-flag** is reset when you edit a file.
7. Place the directory that contains the new **aopstart** EXEC in the PATH environment variable of each operator's **/u/***userid***/.profile** file *before* the directory that contains Infoprint Server files (default directory **/usr/lpp/Printsrv/bin**). For example, you can add this line to the end of the operators' profile files.

   ```
   export PATH=/usr/sbin:$PATH
   ```
8. Specify the full path name of the **aopstart** EXEC in the **/etc/rc** shell script. See "Starting Infoprint Server Daemons Automatically" on page 42 for more details.
9. Specify the full path name of the **aopstart** EXEC on the PARM parameter in the AOPSTART procedure. See "Editing the AOPSTART Startup Procedure" on page 39 for more details.

Rule: Do *not* change the owner or permissions of the **aopstart** file. The file owner must be UID 0. Also, we recommend that the file be writable only by a user with a UID 0. To switch to an effective UID of 0, you can use the z/OS UNIX **su** command if you are permitted to the BPX.SUPERUSER profile in the FACILITY class within RACF.

Creating a User ID to Assign to the AOPSTART and AOPSTOP Procedures

Follow these steps to create a user ID to assign to the AOPSTART and AOPSTOP procedures. This user ID must be defined to RACF as a z/OS UNIX user and be a member of the AOPOPER group.

Tip: The sample CLIST in SYS1.SAMPLIB(AOPRACF) contains the RACF commands that are described in this section. If you use AOPRACF, you still must create the home directory for this user ID as described in step 2. You must modify AOPRACF before you run it.

1. Define group AOPOPER to RACF as described in "Establishing Security for the Printer Inventory and Operator Commands" on page 33. You can use another name for the AOPOPER group or use the AOPADMIN group.
2. Create a z/OS user ID. The user ID must have an OMVS segment, a home directory, and a default group that has an OMVS segment with a group identifier (GID).

   The following RACF commands define user AOPSTC with default group AOPOPER. You can select any name for the user ID, or you can alter an existing user ID to be a z/OS UNIX ID. You can select any default group that has an OMVS segment and a GID. The user ID must be connected to the default group.

   For *user-identifier*, specify an integer that is different from other UIDs in your installation. The NOPASSWORD operand defines the user ID as a *protected* user ID; protected user IDs cannot log on to the z/OS system, and they cannot be revoked due to incorrect password attempts.

   ```
   ADDUSER AOPSTC OMVS(UID(user-identifier) HOME('/u/aopstc')
      PROGRAM('/bin/sh')) DFLTGRP(AOPOPER) NOPASSWORD
   ```

Create the home directory specified in the ADDUSER command. The following z/OS UNIX commands switch to an effective UID of 0 and create a home directory for user ID AOPSTC with the required permissions:

```
su
cd /u
mkdir -m 0700 aopstc
chown -R AOPSTC /u/aopstc
```

To use the **su** command, you must be permitted to the BPX.SUPERUSER profile in the FACILITY class within RACF.

3. Connect the user ID to group AOPOPER. The following RACF command connects user ID AOPSTC to group AOPOPER:

```
CONNECT (AOPSTC) GROUP(AOPOPER)
```

Guidelines:

1. Be sure to define the user ID assigned to the AOPSTART and AOPSTOP procedures to RACF as a z/OS UNIX user. If the user ID is not a z/OS UNIX user, the Infoprint Server daemons will not start or stop, and you might not receive any error messages.
2. The user ID assigned to the AOPSTART procedure must have READ (or higher) access to system and user AFP resource libraries if your installation uses the AFP to PCL, AFP to PostScript, or AFP to PDF transforms.

Defining the AOPSTART and AOPSTOP Procedures to RACF

You must define the AOPSTART and AOPSTOP procedures to the RACF STARTED profile in the FACILITY class. If you want to use the started procedure table (ICHRIN03) instead, refer to *z/OS Security Server RACF Security Administrator's Guide* for more information.

For example, the following commands define the AOPSTART and AOPSTOP procedures. This example assumes that the user ID you defined for use with these procedures is AOPSTC and that AOPSTC is connected to group AOPOPER.

```
RDEFINE STARTED AOPSTART.* STDATA(USER(AOPSTC) GROUP(AOPOPER))
RDEFINE STARTED AOPSTOP.* STDATA(USER(AOPSTC) GROUP(AOPOPER))
SETROPTS RACLIST(STARTED) REFRESH
```

Tip: The sample CLIST in SYS1.SAMPLIB(AOPRACF) contains the RACF commands described in this section. You must modify AOPRACF before you run it.

Editing the AOPSTART Startup Procedure

The AOPSTART procedure provided in SYS1.PROCLIB invokes the **aopstart** command to start the Printer Inventory Manager daemon (**aopd**) and any other daemons that are specified in the **start-daemons** attribute in the **aopd.conf** configuration file.

You must edit the AOPSTART procedure if any of the following conditions apply to your installation:

- If the default region size set by your installation is not sufficient, specify the REGION parameter on the EXEC statement. If you start the IPP server, a region size of at least 100M is required. If you start the Transform Manager, the minimum region size depends on the memory used by the transform. The PDF and PostScript to AFP transform requires at least 42M.
- If the **aopstart** command is not in the default directory **/usr/lpp/Printsrv/bin**, change the directory in the PARM parameter on the EXEC statement.

- If you want to turn tracing on or view messages in Japanese, specify the following environment variables in a data set referred to by the STDENV DD statement:
  - To trace Infoprint Server daemons, specify the AOPTRACEON=1 variable.
  - To view Infoprint Server daemon messages in Japanese, specify the LANG=Ja_JP variable.

  An MVS data set referred to by the STDENV DD statement must have the following DCB attributes: RECFM=VB and LRECL=255. See the comments in the AOPSTART procedure for more information about how to define an environment file.

  If you need to specify environment variables other than AOPTRACEON and LANG, specify them in the **aopstart** EXEC, as described in "Editing the **aopstart** EXEC" on page 36.

Figure 10 shows the AOPSTART procedure provided in SYS1.PROCLIB. See Appendix D, "AOPBATCH Program" on page 271 if you need more information about the AOPBATCH program.

```
//*-------------------------------------------------------------------
//*
//*  AOPSTART - This procedure starts the Infoprint Server daemons.
//*             The USERID assigned to this proc must be a member of
//*             the AOPOPER group.
//*
//*-------------------------------------------------------------------
//AOPSTART PROC
//*-------------------------------------------------------------------
//*  AOPBATCH PARM specifies the path of the program to be run.
//*  If Infoprint Server was not installed in /usr/lpp/Printsrv,
//*  then this path must be changed as required for the installation
//*  directory.
//*-------------------------------------------------------------------
//AOPSTART EXEC PGM=AOPBATCH,PARM='//usr/lpp/Printsrv/bin/aopstart',
//    TIME=NOLIMIT
//STDOUT   DD   SYSOUT=*
//STDERR   DD   SYSOUT=*
//*-------------------------------------------------------------------
//*  STDENV is optionally used to specify environment variables.
//*
//*  A Unix System Services file can be specified, for example:
//*    //STDENV  DD  PATH='/etc/Printsrv/env.conf',PATHOPTS=ORDONLY
//*
//*  or an MVS data set, for example:
//*    //STDENV  DD  DISP=SHR,DSN=hlq.stdenv
//*
//*  This file or data set should contain VARIABLE=VALUE pairs
//*  starting in column 1, for example:
//*
//*  AOPTRACEON=1
//*-------------------------------------------------------------------
```

Figure 10. AOPSTART Startup Procedure — SYS1.PROCLIB(AOPSTART)

Editing the AOPSTOP Shutdown Procedure

The AOPSTOP procedure provided in SYS1.PROCLIB invokes the **aopstop** command to stop some or all active Infoprint Server daemons immediately or after current activity stops.

You need to edit the AOPSTOP procedure if any of the following conditions applies to your installation:

- If the **aopstop** command is not in the default directory **/usr/lpp/Printsrv/bin**, change the directory in the PARM parameter on the EXEC statement.
- If the Infoprint Server configuration file is *not* in the default location **/etc/Printsrv/aopd.conf**, specify the AOPCONF environment variable in a data set referred to by the STDENV DD statement.

  The AOPSTOP procedure does *not* use environment variables that are set elsewhere, for example in the **/etc/profile** file.

When the operator invokes this procedure without any options, all active daemons stop when current activity ends. The operator can specify **aopstop** command options to stop daemons immediately and to stop selected daemons. For example, to stop the Transform Manager and line printer daemons immediately, specify the following MVS START command:

```
START AOPSTOP OPTIONS='-d xfd -d lpd now'
```

Refer to *z/OS Infoprint Server Operation and Administration* for the **aopstop** command options and more examples of how to specify options on the AOPSTOP procedure.

Figure 10 on page 40 shows the AOPSTOP procedure provided in SYS1.PROCLIB. See Appendix D, “AOPBATCH Program” on page 271 if you need more information about the AOPBATCH program.

```
//*-------------------------------------------------------------------
//*
//*  AOPSTOP - This procedure stops the Infoprint Server daemons.
//*            The USERID assigned to this proc should be a member
//*            of the AOPOPER group.
//*
//*-------------------------------------------------------------------
//AOPSTOP  PROC OPTIONS=
//*-------------------------------------------------------------------
//*  AOPBATCH PARM specifies the path of the program to be run.
//*  If Infoprint Server was not installed in /usr/lpp/Printsrv,
//*  then this path should be changed as required for the installation
//*  directory.
//*-------------------------------------------------------------------
//AOPSTOP  EXEC PGM=AOPBATCH,
//   PARM='//usr/lpp/Printsrv/bin/aopstop &OPTIONS'
//STDOUT   DD   SYSOUT=*
//STDERR   DD   SYSOUT=*
//*-------------------------------------------------------------------
//*  STDENV is optionally used to specify environment variables.
//*
//*  A Unix System Services file can be specified, for example:
//*    //STDENV  DD  PATH='/etc/Printsrv/env.conf',PATHOPTS=ORDONLY
//*
//*  or an MVS data set, for example:
//*    //STDENV  DD  DISP=SHR,DSN=hlq.stdenv
//*
//*  This file or data set should contain VARIABLE=VALUE pairs
//*  starting in column 1, for example:
//*
//*  AOPCONF=/etc/Printsrv/aopd.conf
//*-------------------------------------------------------------------
```

Figure 11. AOPSTOP Shutdown Procedure — SYS1.PROCLIB(AOPSTOP)

Starting Infoprint Server Daemons Automatically

To start Infoprint Server daemons automatically during system initialization, you can use one of the following methods:

- z/OS automation tools to execute the AOPSTART procedure
- The **/etc/rc** shell script to run the **aopstart** command. For example, if the **aopstart** EXEC is in the default directory, add the following command to **/etc/rc**:

  ```
  /usr/lpp/Printsrv/bin/aopstart
  ```

Guidelines:

- Start the Printer Inventory Manager daemon (**aopd**) *before* you start IP PrintWay or NetSpool.
- If the PSF for OS/390 functional subsystem (FSS) uses the Printer Inventory, start the Printer Inventory Manager daemon (**aopd**) before you start any printers in the PSF FSS. Refer to *PSF for OS/390 & z/OS: Customization* for information about how to start PSF.
- Start the z/OS SNMP agent (**osnmpd**) *before* you start the Infoprint Server SNMP subagent daemon (**aopsnmpd**). Refer to *z/OS Communications Server: IP Configuration Reference* for information about how to start **osnmpd**.
- Start TCP/IP and make sure it has finished initialization before you start the Infoprint Server LPD, IPP server, or SNMP subagent daemons. You do not have to start TCP/IP if you start only the Printer Inventory Manager daemon (**aopd**) and the Transform Manager daemon (**aopxfd**).

  If TCP/IP is not completely active, you might receive the following error message:

  ```
  BPXF024I (BPXOINIT aoplpd: socket(AF_INET,SOCK_STREAM,0)failed in
  TCPSocket::TCPSocket (int&,int,int) at ./src/tcpsock.cpp 83:
  EDC5112I Resource temporarily unavailable. errno2=0x74b30296
  ```

  If you add the **aopstart** command to **/etc/rc**, use the **setstack** command to ensure that TCP/IP has finished initialization. The **setstack** command is not formally supported by IBM; however, you can download it from the “Tools & Toys” link at Web site (`http://www.ibm.com/servers/eserver/zseries/zos/unix/`) to a directory identified in the **PATH** environment variable. For example, if the **aopstart** EXEC is in the default directory, you could add this line to **/etc/rc**:

  ```
  setstack -w /usr/lpp/Printsrv/bin/aopstart
  ```

Enabling Infoprint Server ISPF Panels

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

Infoprint Server ISPF panels are available in English and Japanese. The Japanese version of the panels have been translated to code page IBM-939.

Defining Libraries in the TSO Logon Procedure

If you have followed the instructions in the ISPF Setup sections in the *z/OS Program Directory*, you can skip the instructions in this section; otherwise, concatenate the following libraries in the TSO logon procedure:

- Concatenate one of the following panel libraries to DD statement ISPPLIB, depending on the language in which you want to view the panels.

  English AOP.SAOPPENU

  Japanese AOP.SAOPPJPN

- Concatenate one of the following message libraries to DD statement ISPMLIB, depending on the language in which you want to receive messages.

  English AOP.SAOPMENU

  Japanese AOP.SAOPMJPN
- Concatenate library AOP.SAOPEXEC to DD statement SYSPROC or SYSEXEC.

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 LNKLST, concatenate these data sets to the ISPLLIB DD statement, or concatenate them to STEPLIB; the TSOLIB function of TSO/E could be used.

Displaying the Infoprint Server Panel Option

ISPF provides sample ISPF panels to enable panels for most z/OS elements. The system programmer panel, ISR@390S, includes an entry for Infoprint Server. If you are not using the default ISPF panels, customize your ISPF to display the **Infoprint Server** option. Refer to the *z/OS Program Directory* for information about how to use these sample panels to customize your ISPF .

Modifying the AOPINIT EXEC

Before using the ISPF panels, review the AOPINIT EXEC provided in AOP.SAOPEXEC(AOPINIT) and modify it, if necessary, for your installation. The AOPINIT EXEC sets default values for fields that display on the Infoprint Server Configuration panel. You can change any of the following values in the AOPINIT EXEC; ISPF users can also change the values for their ISPF sessions on the Configuration panel.

- Configuration file

  The path name of the Infoprint Server configuration file. The default file name is **/etc/Printsrv/aopd.conf**.
- 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**.
- Language

  The value of the **LANG** environment variable. The default is **En_US**.
- 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**

  The default is **/usr/lpp/Printsrv/%L/%N**.
- IP PrintWay message 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.

- 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**.

To modify AOPINIT, copy AOPINIT to a local library and follow the instructions in the comments in AOPINIT to make your changes. Then add the local library that contains AOPINIT to the SYSPROC or SYSEXEC DD statement of each user's TSO logon procedure at the beginning of the search order.

Backing Up and Restoring the Printer Inventory

You should back up the Printer Inventory on a regular basis. You can 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; see "Establishing Security for the Printer Inventory and Operator Commands" on page 33 for information.

Rule: Use the PIDU **export** command to backup the Printer Inventory. Do *not* use other copy commands to create a backup copy of the Printer Inventory because the copy might contain inconsistent data that makes the Printer Inventory unusable.

When you restore the Printer Inventory, no print jobs are lost. However, information in the Printer 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.

You can back up the Printer Inventory to either an HFS file or a data set in a generation data group (GDG), as described in the following sections.

Steps for Backing up the Printer Inventory to an HFS File

Perform the following steps to back up the Printer Inventory to an HFS file at the same time each day. These steps show how to back up the Printer Inventory at 3 AM each day to file **/var/Printsrv/inventory.backup.***yymmdd*, where *yymmdd* is the date of the backup.

1. 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 **/etc/profile** or **/u/***userid***/.profile**, put the following lines in this file:

   ```
   . /etc/profile
   . /u/userid/.profile
   extension=$(date + "%y%m%d")
   pidu -c "export /var/Printsrv/inventory.backup.$extension;"
   ```

   Otherwise, put the following lines in this file to export the environment variables. You can omit the AOPCONF variable if the Infoprint Server configuration file is **/etc/Printsrv/aopd.conf**.

   ```
   export PATH=/usr/lpp/Printsrv/bin:/bin:$PATH
   export NLSPATH=/usr/lpp/Printsrv/%L/%N:$NLSPATH
   export AOPCONF=/etc/Printsrv/aopd.conf
   extension=$(date + "%y%m%d")
   pidu -c "export /var/Printsrv/inventory.backup.$extension;"
   ```

2. Set the execute permission bit for the **/u/***userid***/do.backup** file. For example:

   ```
   chmod +x /u/userid/do.backup
   ```

3. 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*.

Steps for Restoring the Printer Inventory from an HFS File

Perform the following steps to restore the Printer Inventory from an HFS file:

1. Log in to z/OS UNIX System Services as **root** (z/OS UNIX superuser).
2. Stop NetSpool and IP PrintWay.
3. From the z/OS UNIX command line, run the **aopstop** command to stop all Infoprint Server daemons.
4. 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.
5. From the z/OS UNIX command line, run the **aopstart** command to restart the Infoprint Server daemons. Perform step 6 immediately after you start the Infoprint Server daemons.
6. Run 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
   ```

7. Restart NetSpool and IP PrintWay.

Refer to *z/OS Infoprint Server Operation and Administration* for more information about how to stop and restart Infoprint Server.

Steps for Backing Up the Printer Inventory to a Generation Data Group (GDG)

Perform the following steps to back up the Printer Inventory to a data set in a GDG:

1. Create a GDG. The following sample JCL creates a GDG for which five generation data sets are maintained:

   ```
   //AOPGDG JOB  MSGLEVEL=(1,1)
   //STEP1 EXEC  PGM=IDCAMS
   //SYSPRINT  SYSOUT=*
   //SYSIN DD *
     DEFINE GDG (NAME(hlq.INV.BACKUPS) NOEMPTY SCRATCH LIMIT(5))
   /*
   ```

2. Use one of the following methods to run the **pidu export** command to back up the Printer Inventory to a new data set in GDG *hlq*.INV.BACKUPS:
   - Run the **pidu export** command as a batch job:

     ```
     //AOPBINV  JOB  MSGLEVEL=(1,1)
     //STEP1    EXEC PGM=AOPBATCH,PARM='/pidu -c "export //DD:INVENT;"'
     //INVENT   DD   DISP=(NEW,CATLG),DSN=hlq.INV.BACKUPS(+1),
     ```

```
//              DCB=(RECFM=VB,LRECL=32756,BLKSIZE=32760),
//              UNIT=SYSDA,SPACE=(TRK,(50,10),RLSE)
//STDOUT   DD   SYSOUT=*
//STDERR   DD   SYSOUT=*
//STDENV   DD   *
PATH=/usr/lpp/Printsrv/bin:/bin:$PATH
NLSPATH=/usr/lpp/Printsrv/%L/%N:$NLSPATH
AOPCONF=/etc/Printsrv/aopd.conf
/*
```

If Infoprint Server files are located in the default locations, you can omit the STDENV DD statement.

- Run the **pidu export** command as a **cron** job:
  a. 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 **/etc/profile** or **/u/***userid***/.profile**, put the following lines in this file:

     ```
     . /etc/profile
     . /u/userid/.profile
     pidu -c "export //\'hlq.INV.BACKUPS\(\+1\)\';"
     ```

     Otherwise, put the following lines in this file to export the variables. You can omit the AOPCONF variable if the Infoprint Server configuration file is **/etc/Printsrv/aopd.conf**.

     ```
     export PATH=/usr/lpp/Printsrv/bin:/bin:$PATH
     export NLSPATH=/usr/lpp/Printsrv/%L/%N:$NLSPATH
     export AOPCONF=/etc/Printsrv/aopd.conf
     pidu -c "export //\'hlq.INV.BACKUPS\(\+1\)\' ;"
     ```

  b. Set the execute permission bit for the **/u/***userid***/do.backup** file. For example:

     ```
     chmod +x /u/userid/do.backup
     ```

  c. Use the **crontab** command with the **-e** option to insert the following line into the **crontab** entry to back up the Printer Inventory at 3 AM each day.

     ```
     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*.

Steps for Restoring the Printer Inventory from a GDG

Perform the following steps to restore the Printer Inventory from the latest data set in the GDG:

1. Log in to z/OS UNIX System Services as **root** (z/OS UNIX superuser).
2. Stop NetSpool and IP PrintWay.
3. From the z/OS UNIX command line, run the **aopstop** command to stop all Infoprint Server daemons.
4. 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.
5. From the z/OS UNIX command line, run the **aopstart** command to start Infoprint Server daemons. Perform step 6 immediately after you start Infoprint Server daemons.

6. Use one of the following methods to restore the Printer Inventory from the most recent generation data set of a GDG named *hlq*.INV.BACKUPS:
   - Run the **pidu** command as a batch job:

     ```
     //AOPRINV  JOB  MSGLEVEL=(1,1)
     //STEP1    EXEC PGM=AOPBATCH,PARM='/pidu //DD:INVENT'
     //INVENT   DD   DISP=SHR,DSN=hlq.INV.BACKUPS(0)
     //STDOUT   DD   SYSOUT=*
     //STDERR   DD   SYSOUT=*
     //STDENV   DD   *
     PATH=/usr/lpp/Printsrv/bin:/bin:$PATH
     NLSPATH=/usr/lpp/Printsrv/%L/%N:$NLSPATH
     AOPCONF=/etc/Printsrv/aopd.conf
     /*
     ```

     If Infoprint Server files are located in the default locations, you can omit the STDENV DD statement.
   - Run the **pidu** command from the **sh** command:
     a. Create a file to contain a short shell script, for example **/u/***userid***/do.restore**, where *userid* is your login user ID. The following example shows how to restore the Printer Inventory from the most recent generation data set of a GDG named *hlq*.INV.BACKUPS. This example 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 //\'hlq.INV.BACKUPS\(0\)\' >pidu.output 2>pidu.errors
        ```

     b. Run the **sh** command to restore the Printer Inventory by running the **pidu** command in the **/u/***userid***/do.restore** file:

        ```
        sh /u/userid/do.restore
        ```

        Tip: Use the **sh** command to run the **pidu** command in the **/u/userid/do.restore** file instead of running the **pidu** command directly. GDG relative generations are maintained for the duration of the UNIX shell created when you log into z/OS UNIX System Services. Using the **sh** command to create a new shell ensures that the current generation is always 0. For example, the following commands create a new generation data set and then display the generation data set prior to the newly created data set:

        ```
        pidu -c "export //\'hlq.INV.BACKUPS\(\+1\)\';"
        cat //\'hlq.INV.BACKUPS\(0\)\'
        ```

        To create a new generation data set and then display that data set, you would need to run the following command:

        ```
        pidu -c "export //\'hlq.INV.BACKUPS\(\+1\)\';"
        cat //\'hlq.INV.BACKUPS\(\+1\)\'
        ```
7. Restart NetSpool and IP PrintWay.

Refer to *z/OS Infoprint Server Operation and Administration* for more information about how to stop and restart Infoprint Server.

Chapter 4. Customizing Print Interface

This chapter describes how to customize the Print Interface component of Infoprint Server. You must customize Print Interface in order to perform the following functions:

- Submit print requests from the following sources:
  - z/OS UNIX shell commands (**lp**, **lpstat**, or **cancel**) or the AOPPRINT JCL procedure
  - Remote clients that use the TCP/IP LPR to LPD protocol, Internet Printing Protocol (IPP), or Server Message Block (SMB) protocol
  - Infoprint Port Monitor for Windows
  - Infoprint Server Application Programming Interface (API)
  - SAP R/3 Application Server for z/OS
  - A batch job that requests the Print Interface subsystem in the SUBSYS parameter on a DD statement
- Transform output data sets on the JES spool from one data stream format to another using Infoprint Server Transforms

Print Interface processes print requests from local users and remote clients and allocates output data sets on the JES spool. See “Print Interface” on page 5 for an overview of Print Interface and how this component fits into your system.

This chapter describes the following customization tasks. Required tasks are required by all installations that want to use Print Interface. Optional tasks are required only if the listed condition applies to your installation.

Task	Condition	See Page:
Identifying Software Requirements	Required	50
Common Customization for Print Interface	Required	51
Customizing the z/OS UNIX Printing Commands	Optional: To use **lp** command	60
Customizing the AOPPRINT JCL Procedure	Optional: To use AOPPRINT procedure	60
Customizing the Print Interface LPD	Optional: To print from an LPR, the Infoprint Port Monitor for Windows, or a remote SAP R/3 application server; also required to use the IP PrintWay transform function.	61
Customizing Print Interface Support for the z/OS SMB Server	Optional: To print from a Windows client using SMB protocol	66
Customizing the Print Interface IPP Server	Optional: To print from an IPP client	67
Customizing SAP R/3	Optional: To print from SAP R/3	71
Customizing the Print Interface Subsystem	Optional: To use the Print Interface subsystem	79

© Copyright IBM Corp. 1995, 2002

The following table describes other components of Infoprint Server that you might need to customize:

Task	Condition	See Page:
Customizing the Printer Inventory Manager	Required	25
Customizing Infoprint Server Transforms	Optional: To transform data.	83
Customizing IP PrintWay	Optional: To print to printers in a TCP/IP network	163
Customizing the Windows client	Optional: To submit print requests from the Infoprint Port Monitor for Windows	25

Related tasks: After you customize Print Interface, the administrator and operator must perform the following tasks. Refer to *z/OS Infoprint Server Operation and Administration* for information about these tasks:

- The administrator must create a printer definition in the Printer Inventory for each printer and specify the fields in the printer definition that Print Interface requires.
- If you want to use the Print Interface LPD, the IPP server, or the Print Interface subsystem, the operator must run the **aopstart** command or the AOPSTART procedure to start the corresponding Infoprint Server daemon.

Identifying Software Requirements

This section documents the IBM software and PTFs that are required beyond the z/OS base elements.

To print using the SMB printing protocol, the z/OS SMB print server, which is part of the z/OS Distributed File Service element, must be configured and active.

To use the Print Interface IPP server, Java for z/OS (5655-A46) level 1.3 is required.

To print from SAP R/3, one of the following SAP R/3 application servers, which can be ordered from SAP, is required:

- The SAP R/3 Application Server for z/OS
- An SAP R/3 application server for another system (such as Windows NT, AIX, or Solaris)

To print SAP R/3 output to IBM AFP printers on the z/OS system:

- Infoprint Server Transforms (5697-F51). The Transforms to AFP feature of Infoprint Server Transforms contains the SAP to AFP transform and is provided at no extra cost to Infoprint Server customers.
- PSF V3R1.0 for OS/390 (5655-B17) or later.

To transform data remotely using Infoprint Manager, one of the following IBM products is required:

- Infoprint Manager V3R2 for AIX (5785–E42)
- Infoprint Manager V1R1 for Windows NT and Windows 2000 (5639–I27)

To transform PostScript or PDF data to AFP format for printing on the IBM Infoprint Color 130 Plus printer, the following are required:

- Infoprint Manager V3R2 for AIX (5785–E42), with the latest service applied. Refer to *IBM Infoprint Manager for AIX: Administrator's Guide* for information about AIX system requirements.
- Infoprint Server PTF UW85327

To use the Print Interface subsystem, the PTF for Infoprint Server PTF OW53202 is required.

Common Customization for Print Interface

This section describes the following customization tasks, which you must perform before you submit print requests to Print Interface. Required tasks are required by all installations that want to use Print Interface; optional tasks are required only if the listed condition applies to your installation.

Task	Condition	See Page:
Customizing z/OS Communications Server	Optional: To receive print requests from remote systems, print to TCP/IP-attached printers, or send output to e-mail destinations	51
Customizing JES to Specify No Blank Truncation	Required	53
Updating the BPXPRMxx Member of SYS1.PARMLIB	Required	54
Editing the aopd.conf Configuration File	Optional: To change default configuration	54
Writing a Filter Program	Optional: To inspect or modify input data stream	55

Customizing z/OS Communications Server

You must customize and start the TCP/IP component of z/OS Communications Server if (1) you plan to submit print jobs to Print Interface from remote systems or (2) you plan to use IP PrintWay to transmit print jobs to remote printers or e-mail destinations. This section describes considerations related to customizing TCP/IP. For complete information about how to customize TCP/IP, refer to the following publications:

- *z/OS Communications Server: IP Configuration Guide*
- *z/OS Communications Server: IP Configuration Reference*

Editing the *hlq*.PROFILE.TCPIP Data Set

You can customize the *hlq*.PROFILE.TCPIP data set when you customize TCP/IP. The *hlq*.PROFILE.TCPIP data set contains statements that initialize the TCP/IP address space. Special considerations exist for Print Interface and IP PrintWay for two of these statements: TCPCONFIG and PORT.

Tips:

- If you change any of the values in the *hlq*.PROFILE.TCPIP data set, restart TCP/IP to pick up the changes.
- See "Customizing z/OS Communications Server" on page 165 for considerations that apply if you also run IP PrintWay.

TCPCONFIG Statement: The TCPCONFIG statement in the *hlq*.PROFILE.TCPIP data set updates the TCP layer of TCP/IP. Following is a sample TCPCONFIG statement that you might want to include in the *hlq*.PROFILE.TCPIP data set, followed by an explanation of the parameters and considerations for both Print Interface and IP PrintWay.

```
TCPCONFIG INTERVAL 10 SENDGARBAGE TRUE UNRESTRICTLOWPORTS TCPRCVBUFRSIZE
32768 TCPSENDBFRSIZE 32768
```

INTERVAL *minutes* **SENDGARBAGE TRUE**

Number of minutes TCP waits after receiving a packet for a connection before it sends a keep-alive packet for that connection.

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 its use; therefore, specify this parameter to ensure 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.

Note: You can also specify some of these parameters on other TCP/IP statements; the settings from the last statement processed are used.

PORT Statement: The port numbers that are used by the Print Interface LPD, IPP server, and the z/OS SNMP agent must not be reserved in the *hlq*.PROFILE.TCPIP data set. The default port numbers are:

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

Recommendation: Use the default ports for Infoprint Server components (such as the LPD) unless they are not suitable for your installation. See “Editing the aopd.conf Configuration File” on page 64 for limitations that exist when you configure the Print Interface LPD to use a port other than 515.

To ensure that the port numbers are not reserved, remove or comment out the port numbers in the PORT statement. For example, if the Print Interface LPD uses port 515, comment out the line in the PORT statement in your *hlq*.PROFILE.TCPIP data set that lists port 515. Use a semicolon in the first column to indicate the line is a comment:

Example:

```
PORT
:
;    515 TCP LPSERVE             ; LP Server
```

Tip: Port 515 is reserved in the sample *hlq*.PROFILE.TCPIP data set; therefore, if your installation uses the sample data set, you must comment out the line that reserves port 515.

Customizing JES to Specify No Blank Truncation

JES2 and JES3, by default, remove blank characters from the end of each record on the JES spool in order to save spool space. IBM recommends that you request no blank truncation for those output classes in which Print Interface can allocate output data sets, because ASCII data streams (such as PCL and PostScript) can be corrupted when JES removes EBCDIC blank characters.

Print Interface allocates output data sets in the output classes specified in the CLASS fields of the printer definitions. If no output class is specified in a printer definition, Print Interface allocates output data sets in the JES default output class, typically class A; however, for print requests submitted with the Windows SMB protocol, the default class might be the MSGCLASS from the z/OS SMB server if it is a started job.

To request no blank truncation, specify the following JES parameters:

- **JES2:** BLNKTRNC=NO on the JES2 OUTCLASS statement. The default is BLNKTRNC=YES.
- **JES3:** TRUNC=NO on the JES3 SYSOUT statement. The default is TRUNC=YES; however, you can change the default in the TRUNC parameter of the JES3 BUFFER statement.

For example, if Print Interface can allocate data sets in output class A, P, and S, specify BLNKTRNC=NO (in JES2) and TRUNC=NO (in JES3) for all three classes: A, P, and S.

Tips:

- You *must* specify *no* blank truncation for the output classes whenever the administrator has specified the AFP to PCL, AFP to PostScript, or AFP to PDF transform in an IP PrintWay printer definition. This is because these transforms create PCL, PostScript, or PDF output which Print Interface writes to the JES spool.
- Whenever the administrator specifies a new JES output class in the Allocation section of a printer definition, be sure that no blank truncation is specified for this output class.

For more information, refer to the following publications:

- *z/OS JES2 Initialization and Tuning Reference* for information about the JES2 OUTCLASS statement.
- *z/OS JES3 Initialization and Tuning Reference* for more information about the JES3 SYSOUT statement.

Updating the BPXPRMxx Member of SYS1.PARMLIB

When you use Print Interface, you should increase the number of active z/OS UNIX processes that the z/OS system allows. You can specify the number of allowed processes in the following parameters in the BPXPRMxx member of SYS1.PARMLIB:

Parameter **Meaning**

MAXPROCUSER
The maximum number of processes that a single z/OS UNIX user ID can have concurrently active.

Recommendation: Specify at least 70.

MAXPROCSYS
The maximum number of processes that the z/OS system allows.

Recommendation: Add 200 to the current MAXPROCSYS value. If you also use IP PrintWay, also add 70 for each IP PrintWay FSS that you create.

Example: If you use Print Interface and also have defined 10 IP Printway FSSs, specify the following statements in the BPXPRMxx member:

```
MAXPROCUSER(70)   /*70(Print Interface)                          */
MAXPROCSYS(1100)  /*200(Print Interface)+700(PrintWay)+200(initial) */
```

Tip: You can use the SETOMVS or SET OMVS command to dynamically increase the MAXPROCUSER and MAXPROCSYS values; however, to make a permanent change, edit the BPXPRMxx member that is used for IPLs. For more information about the BPXPRMxx member, refer to *z/OS MVS Initialization and Tuning Reference*.

Editing the aopd.conf Configuration File

Print Interface uses the following attributes in the **aopd.conf** configuration file. Add or edit these attributes to customize Print Interface for your installation. See Appendix B, “Infoprint Server Files” on page 245 for more information about all of the attributes in the configuration file and rules for editing the file.

job-prefix = *prefix*
A prefix that Print Interface uses for all Infoprint Server job IDs. Specify two letters, numbers, or national (@ $ #) characters. Enclose a prefix that contains national characters in single or double quotation marks. The first character *cannot* be numeric.

Print Interface assigns a unique Infoprint Server job ID to each sysout data set that it creates on the JES spool. The Infoprint Server job ID begins with the prefix you specify. The z/OS UNIX **lpstat** and **cancel** commands use the Infoprint Server job ID to display and delete data sets.

z/OS also assigns a unique job ID to each output data set. The z/OS job ID is different from the Infoprint Server job ID. JES operator commands always return the z/OS job ID.

In most cases, the job ID field of the sysout data set on the JES spool contains the Infoprint Server job ID, instead of the z/OS job ID. However, in most cases where Print Interface processes a data set that already has a z/OS job ID associated with it, Print Interface uses the existing z/OS job ID in the job ID field, instead of overriding it with the Infoprint Server job ID.

Default: `job-prefix = PS`

Example: `job-prefix = 'P#'`

Print Interface also uses the **inventory**, **base-directory** , **ascii-codepage**, and **ebcdic-codepage** attributes in the **aopd.conf** configuration file. The Printer Inventory Manager also uses these attributes; see "Creating the Infoprint Server Configuration File (**aopd.conf**)" on page 28 for a description.

Writing a Filter Program

> **Note**
> This section contains programming-interface information.

Print Interface provides support that lets your installation write filter programs to monitor and modify data before Print Interface writes the data to an output data set on the JES spool. A filter can perform the following types of functions:

- Convert data in a document from one data format to another
- Add a header or trailer page to each document
- Add printer instructions to the beginning of each document
- Add, delete, or modify data in a document
- Terminate processing of a document

You can write two types of filters:

- DLL filter: A filter that resides in a dynamic link library (DLL) and is defined by **cfilter.h**.
- UNIX filter: A filter that reads from **stdin** and writes to **stdout**.

For some applications, you might need to write a DLL filter instead of a UNIX filter because a DLL filter can inspect more information about the input data stream and can also perform more functions. For example, only in a DLL filter can you change the attribute that indicates the format of the data stream. If you want to change the data format, therefore, you must write a DLL filter instead of a UNIX filter. See "Writing a DLL Filter" on page 57 and "Writing a UNIX Filter" on page 58 for more information about differences between DLL and UNIX filters.

Alternatives:

- Instead of writing a filter to add header pages and trailer pages to output data sets that IP PrintWay transmits to the target printer, you can write an IP PrintWay Begin Data Set and an IP PrintWay End Data Set exit routine; see "Begin Data Set Exit" on page 204 and "End Data Set Exit" on page 216 for more information.
- Instead of writing a filter to specify printer instructions for IP PrintWay to add to the beginning and end of each data set, the administrator can specify these instructions in a printer definition. Refer to *z/OS Infoprint Server Operation and Administration* for more information.
- Instead of writing a filter to add header and trailer pages to output data sets that PSF for OS/390 processes, you can write a PSF for OS/390 separator page exit. Refer to *PSF for OS/390 & z/OS: Customization* for information.

Filter Options

You can pass filter options and operands to your DLL or UNIX filter. The admininstrator can specify filter options and operands in the printer definition, and the user can specify filter options in the **filter-options** job attribute.

The administrator can use the special **%** operator to pass the value of a job attribute to the filter. For example, if a user specifies a job attribute on the **lp** command, the value of the attribute can be passed to the filter. Use the following format:

%*attribute_name*

where:

% Causes the entire %*attribute-name* substring to be replaced by the value of the attribute.

attribute-name
Is the name of any job attribute, as defined in *z/OS Infoprint Server User's Guide*.

If an attribute value is empty or contains special character or spaces, the value is enclosed in quotation marks and passed to the filter as one argument.

Examples: **Example 1: Using the % operator**

In this example, **%address-text** is specified as an operand for UNIX filter **test_filter** in the printer definition:

```
spawn /usr/local/bin/test_filter %address-text
```

If the **lp** command specifies the following **address-text** job attribute:

```
lp -o "address-text={a1 a2}"
```

Filter **test_filter** is passed these arguments:

```
/usr/local/bin/test_filter "{ a1 a2 }"
```

The filter program receives /usr/local/bin/test_filter as the first argument and { a1 a2 } as the second argument (without quotes).

Example 2: Using the % operator with a filter written in the REXX language

Sometimes the values for job attributes are null. If you use REXX to write your filter, use a **parse** statement like the following one to parse arguments, so that null arguments are properly passed to the program:

```
parse arg arg1 ' ' arg2 ' ' arg3
```

Sometimes the values for job attributes contain embedded spaces. Because the **parse** statement shown above does not correctly receive an argument that contains embedded spaces, use an alternative delimiter such as **/**, to delimit filter options that use the % operator; ensure that the delimiter you select does not have a special meaning to the shell and does not occur in the data. For example, if the administrator uses **/** to delimit filter options in the printer definition as shown:

```
spawn /usr/local/bin/test_filter /%name-text/%document-format/
```

And, if the **lp** command specifies the **name-text** job attribute:

```
lp -o "name-text='mike smith'" text.file
```

The command, after substitution of the attribute values, is:

```
/usr/local/bin/test_filter /"mike smith"/text/
```

You can use the following REXX **parse** statement to parse the arguments:

```
parse arg '/' name_text '/' doc_format '/'
```

For more information about how to specify the filter name in a printer definition, see "Invoking a DLL or UNIX Filter" on page 58 and refer to *z/OS Infoprint Server Operation and Administration*.

Writing a DLL Filter

To assist you in writing a new DLL filter, IBM provides the source code for the **aopfiltr.so** DLL filter in the following files:

- **cfilter.h**, a header file that contains declarations and interface descriptions. Do not change the contents of this file.
- **cfilter.c**, the source code for the **aopfiltr.so** filter.

Both files are in directory **/usr/lpp/Printsrv/samples**. For more information about **aopfiltr.so**, see "Filters Provided With Infoprint Server and Infoprint Server Transforms" on page 59.

To customize the **aopfiltr.so** filter, do the following:

- Modify file **cfilter.c** as necessary, using the comments in the prologs of the file.
- Link the filter as a dynamic link library (DLL), as described in the comments in file **cfilter.c**.

Input to a DLL Filter: A DLL filter has access to the following information:

- Filter options and operands.
- Environment variables.
- Input data in a document; data in text format has already been converted to the printer code page.
- Name of the printer code page used to convert the data.
- Format of the data, as determined by Print Interface or as specified by the job submitter.
- The LPD control file in a format defined by RFC 1179.
- The type of carriage controls in the input file.
- The name of the input file.
- The name of the printer definition for the print request.
- Job attributes. Job attributes can be specified either by the job submitter or in the printer definition. For a list of Infoprint Server job attributes, refer to *z/OS Infoprint Server User's Guide*.

Output from a DLL Filter: A DLL filter can return the following output to Print Interface:

- Modified data.
- Format of the data.
- The type of carriage controls.
- Job attributes. For a list of Infoprint Server job attributes, refer to *z/OS Infoprint Server User's Guide*. Job attributes specified by the job submitter override the same job attributes set by the filter.
- Code page. Text and line-data documents are converted to this code page before the data is passed as input to the filter.
- Return code. If the return code indicates an error, Print Interface ends processing of the document.

Writing a UNIX Filter

A UNIX filter can be a shell executable, such as a C program, REXX exec, shell script, and so on. Print Interface spawns a UNIX filter as a separate process. (**cat** found on all UNIX systems is an example of a UNIX filter.)

The shell script in Figure 12 shows an example of a simple UNIX filter:

```
#!/bin/sh
#  DESCRIPTION:
#    This filter processes EBCDIC (not ASCII).
#    Terminate each line with "carriage return + line feed".
#==============================================================
awk '{ printf "%s\r\n", $0 }';
exit
```

Figure 12. Sample UNIX Filter

Input to a UNIX Filter: A UNIX filter can access the following information:

- Filter options and operands.
- Environment variables.
- The input data in **stdin**; data in text format has already been converted to the printer code page.

Output from a UNIX Filter: A UNIX filter can return the following output to Print Interface:

- Modified data in **stdout**.
- A return code. If the return code indicates an error, Print Interface ends processing of the document and does not allocate the data set on the JES spool.

Invoking a DLL or UNIX Filter

To request that Print Interface invoke a filter, the administrator must specify the filter name and filter options in the printer definition for the target printer. The full path name is required unless the filter is in a directory named in the LIBPATH environment variable (for DLL filters) or in the PATH environment variable (for UNIX filters). For UNIX filters, the administrator must type **spawn** before the filter name.

The administrator specifies the filter in the **Filter** field of the Processing section of a printer definition, or in a Processing component. The administrator can specify different filters for different input data formats. Refer to *z/OS Infoprint Server Operation and Administration* for more detailed information.

Programming Considerations

Consider the following points when coding a filter program:

- You should be an experienced programmer who is aware of overall system implications.
- Do not issue a long-term explicit or implied WAIT in the filter program because it can reduce the throughput of the Print Interface LPD or IPP server.
- The filter program runs in 31-bit addressing mode.
- Code the filter program to be reentrant.
- All filter programs run in problem state.
- Programming exceptions cause Print Interface to abend; an ESTAE is in effect while a filter program is running.

- A DLL filter, in the initialization function, can obtain a work area and pass a pointer to that work area to other functions in the filter program. Using the work area, one function can pass data to another function.

Note: Future Print Interface program maintenance may require that you recompile your filter programs.

Testing Filters

To test your filter, specify the name of the filter in a printer definition and use the **lp** command to print a document to that printer definition. As an alternative, you can use the **filter** command to test your filter; refer to **/usr/lpp/Printsrv/samples/cfilter.h** for a description of the **filter** command.

Modifying Filters

After modifying a DLL filter, you might need to restart the Print Interface LPD or the Print Interface IPP server to cause them to load the new version of the filter. This is necessary only if the LPD or IPP server has already used the old version of the filter with the same name.

To restart the LPD and IPP server, use the **aopstop -d lpd -d ippd** command to stop the LPD and IPP server and then use the **aopstart** command to start them again. Refer to *z/OS Infoprint Server Operation and Administration* for more information about these commands.

You do *not* need to restart the LPD and IPP server when you modify a UNIX filter or if you have written a new filter and tested it only with the **lp** command or the **filter** command.

Filters Provided With Infoprint Server and Infoprint Server Transforms

Infoprint Server provides the following DLL filters in directory **/usr/lpp/Printsrv/lib**:

- **aopfiltr.so**, which converts ASCII line-feed controls (X'0A') 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 and print queues. IBM recommends that the administrator specify the **aopfiltr.so** filter in the printer definition for ASCII printers for text data.
- **aoprform.dll**, which transforms PCL, PostScript, and PDF data remotely using Infoprint Manager for AIX®, Infoprint Manager for Windows NT, or PSF for AIX.
- **lpd_compat.so**, which provides an LPD filter with similar function to the z/OS Communications Server (TCP/IP) LPD.

Infoprint Server Transforms provides the following DLL filters in directory **/usr/lpp/Printsrv/lib**:

- **afp2pcl.dll**, which transforms AFP data and line data to PCL format.
- **afp2pdf.dll**, which transforms AFP data and line data to PDF format.
- **afp2ps.dll**, which transforms AFP data and line data to PostScript format.
- **pcl2afp.dll**, which transforms PCL data to AFP format
- **ps2afp.dll**, which transforms PostScript and PDF data to AFP format
- **sap2afp.dll**, which transforms SAP data to AFP format

Customizing the z/OS UNIX Printing Commands

Print Interface provides enhanced z/OS UNIX shell commands for printing: **lp**, **lpstat**, and **cancel**. Refer to *z/OS Infoprint Server User's Guide* for information about how to use these commands.

The following environment variables affect the behavior of these commands. You can set the environment variables in the **/etc/profile** file so that they apply for all users of the **lp**, **lpstat**, and **cancel** commands. For more information about how to edit the **/etc/profile** file, see "Setting Environment Variables for the Printer Inventory Manager" on page 31.

AOPOPTIONS
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 AOPOPTIONS environment variable before the options you specify on the command line, a user can override the values of this variable. This variable is *optional*; use this environment variable to specify job attributes that are constant for all print jobs. Refer to *z/OS Infoprint Server User's Guide* for the format of the job attributes and values.

AOPPATH
The directory that contains attribute files used by the **lp** command. This variable is *optional*; if you have not created attribute files for the **lp** command, you do not need to set this variable. Refer to *z/OS Infoprint Server User's Guide* for information about attribute files.

MANPATH
The path of directories that contain the man pages. Infoprint Server man pages are available only in English. Add **/usr/lpp/Printsrv/man/En_US** to the values in this variable. This variable is *required*.

Note: Add the new directory path *before* **/usr/man/%L** in the MANPATH variable, so that the Infoprint Server versions of the **lp**, **lpstat**, and **cancel** man pages are displayed.

PRINTER or LPDEST
The default printer for the **lp** command. The value in LPDEST overrides the value in PRINTER.

This variable is *optional*. You can also use Infoprint Serve ISPF panels to define a default printer; see *z/OS Infoprint Server Operation and Administration*. The printer named in either LPDEST or PRINTER overrides the printer named on the ISPF configuration panel.

To use the z/OS UNIX printing commands, you do *not* need to start any daemons in addition to the Printer Inventory Manager daemon.

Customizing the AOPPRINT JCL Procedure

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

Figure 13 shows the AOPPRINT procedure.

```
//*-------------------------------------------------------------------
//* AOPPRINT - OS/390 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 dataset containing environment variables.
//*        Builtin values will work for the default installation.
//*STDENV   DD   DSN=USERID.JCL(ENVVARS),DISP=SHR
```

Figure 13. AOPPRINT JCL Procedure in SYS1.PROCLIB

You can customize the AOPPRINT JCL procedure for your installation in the following ways:

- To change the name of the default printer definition, specify the printer definition name in the PRINTER option on the PROC statement. If you don't change the name, the default is **p1**. The administrator should create a printer definition with the default name.
- To define environment variables used by the Printer Inventory Manager, define environment variables in an STDENV DD statement, as shown in the procedure. If your installation installed Infoprint Server files in the default directory, **/usr/lpp/Printsrv/**, then you do *not* need to specify the LIBPATH, NLSPATH, and PATH environment variables. To define environment variables, specify each variable on a separate line, in the format *variable*=*value*, as follows:

  ```
  LIBPATH=/usr/mylib
  PATH=/usr/mylib
  ```
- If you have not added the Language Environment run-time library (CEE.SCEERUN) or the C++ run-time library (CBC.SCLBDLL) to the system LNKLST, concatenate these data sets to a STEPLIB DD statement.

To use the AOPPRINT procedure, you do *not* need to start any daemons in addition to the Printer Inventory Manager daemon.

Customizing the Print Interface LPD

This section describes how to customize the Print Interface Line Printer Daemon (LPD). You need to customize the Print Interface LPD if you want to:

- Submit print requests to Infoprint Server with commands that use the LPR to LPD protocol, such as:
  - LPR and LPQ commands from MVS, OS/390, z/OS, VM, z/VM, AS/400®, OS/2®, UNIX, and Windows NT/2000
  - **lpr**, **enq** and **qstat** commands from AIX systems
- Submit print requests to Infoprint Server with the Infoprint Port Monitor for Windows from Windows 95/98 and NT/2000 systems.
- Transform data sets that Print Interface does not allocate on the JES spool (for example, data sets that NetSpool allocates on the JES spool and that the z/OS system allocates on the JES spool) and that IP PrintWay transmits to remote printers. For information about how to select the IP PrintWay transform function in a printer definition, refer to the description of the Resubmit for Filtering option in *z/OS Infoprint Server Operation and Administration*.

Note: You do *not* need to customize the Print Interface LPD to print from Windows systems that use the native Windows Server Message Block (SMB) protocol or from any platform that uses Internet Printing Protocol (IPP).

The Print Interface LPD accepts print requests from clients that use the LPR to LPD protocol as defined by RFC 1179, detects the input data stream and transforms data streams, validates that documents can print as requested, allocates jobs on the JES spool, and provides notification of job completion and job status.

This section describes the following customization tasks. Required tasks are required by all installations that want to use the Print Interface LPD; optional tasks are required only if the listed condition applies to your installation.

Task	Condition	See Page:
Customizing z/OS Communications Server (TCP/IP)	Required	62
Editing the aopd.conf Configuration File	Optional: To change the default Infoprint Server configuration	64
Using the LPD Compatibility Filter	Optional: To obtain formatting similar to z/OS TCP/IP LPD	64
Customizing LPR Clients	Optional: To print from a remote client	65

Related tasks: After you customize the Print Interface LPD, the admininstrator and operator must perform the following tasks. Refer to *z/OS Infoprint Server Operation and Administration* for information about these tasks:

- The administrator must create a printer definition in the Printer Inventory for each printer and specify the fields in the printer definition that Print Interface requires.
- The operator must run the **aopstart** command or the AOPSTART procedure to start the Print Interface LPD.

Customizing z/OS Communications Server (TCP/IP)

z/OS Communications Server also provides an LPD, called the TCP/IP LPD, which is the standard LPD on the z/OS system. You must decide whether to run (1) *only* the Print Interface LPD or (2) *both* the TCP/IP LPD and the Print Interface LPD on the z/OS system.

Guideline: We recommend that you run only the Print Interface LPD, without the TCP/IP LPD, for the following reasons:

- The Print Interface LPD provides comparable function to the TCP/IP LPD. The Print Interface LPD is suitable for most clients that currently use the TCP/IP LPD.
- If you run both the Print Interface LPD and the TCP/IP LPD on the same system, you must configure two TCP/IP stacks so that you can use all functions provided by Infoprint Server. See “Running the TCP/IP and Print Interface LPDs on the Same z/OS System” on page 63 for more information.

Running Only the Print Interface LPD on the z/OS System

If you want to run only the Print Interface LPD, and not the TCP/IP LPD, on the z/OS System, follow these steps:

- In order to use all functions that Infoprint Server provides, the Print Interface LPD must listen at port 515. Therefore, you must ensure that the TCP/IP LPD does

not reserve port 515 for its use. See Editing the *hlq*.PROFILE.TCPIP Data Set on page 51 for information about how to remove port 515 from the PORT statement in the *hlq*.PROFILE.TCPIP data set.

- Update the z/OS Communications Server IP start procedure so that the TCP/IP Remote Print Server (LPD) does not start.
- Ask your administrator to create printer definitions in the Printer Inventory with the same name as the printers that are defined with SERVICE statements in the TCP/IP LPD Server configuration data set, *hlq*.SEXAINST(LPDDATA).
- If you want the Print Interface LPD to function comparably to the TCP/IP LPD, specify the Print Interface LPD compatibility filter in these printer definitions. See "Using the LPD Compatibility Filter" on page 64 for information.

Running the TCP/IP and Print Interface LPDs on the Same z/OS System

If you want to run both the Print Interface LPD and the TCP/IP LPD on the same z/OS system, you can either run (1) each LPD on a *separate* TCP/IP stack or (2) both LPDs on the *same* TCP/IP stack.

Guideline: We recommend that you run each LPD on a separate TCP/IP stack so that you can use all of the functions that Infoprint Server provides.

Running each LPD on a separate TCP/IP stack: You need to define multiple TCP/IP stacks so that the Print Interface LPD can run on one stack, while the TCP/IP LPD runs on the other stack. Each TCP/IP stack has its own IP address, or own host name that resolves to an IP address. When the Print Interface LPD and the TCP/IP LPD each runs on its own stack, you can use all of the functions that Infoprint Server provides because you can assign port 515 to both the Print Interface LPD and the TCP/IP LPD.

The Print Interface LPD must run on a z/OS UNIX System Services stack. You must identify the job name associated with the z/OS UNIX TCP/IP stack in the _BPXK_SETIBMOPT_TRANSPORT environment variable. Specify this variable before you start the Print Interface 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, specify `_BPXK_SETIBMOPT_TRANSPORT=TCPIPOE`. You must set variable _BPXK_SETIBMOPT_TRANSPORT in the **aopstart** EXEC as described in "Editing the **aopstart** EXEC" on page 36. If you do not specify this environment variable, TCP/IP uses its standard search mechanism to determine the TCP/IP stack to use.

For more information about how to configure multiple TCP/IP stacks and set the _BPXK_SETIBMOPT_TRANSPORT variable, refer to the following publications:

- *z/OS Communications Server: IP Configuration Guide*
- *z/OS UNIX System Services Planning*.

Running both LPDs on the same TCP/IP stack: The same TCP/IP stack can serve both the Print Interface LPD and the TCP/IP LPD; however, 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 port other than 515 to the Print Interface LPD. To do this, change the port number in the **lpd-port-number** attribute in the **aopd.conf** file.

Limitations: When you run both LPDs on the same TCP/IP stack, you *cannot* use all of the functions that Infoprint Server provides. The following limitations apply if you select this TCP/IP configuration option:

- Remote users cannot use commands such as LPR and **enq** to submit print requests. This is because you *cannot* configure most LPR clients to send print requests to a port other than to the well-known port 515. The Infoprint Port Monitor for Windows, however, is an exception; you can configure the Port Monitor to send print requests to any port.
- IP PrintWay *cannot* use transforms provided by Infoprint Server Transforms to transform data from and to the AFP data format using the IP PrintWay Resubmit for Filtering option. This is because IP PrintWay can only resubmit data for transform (that is, filtering) to Print Interface at port 515.

Editing the aopd.conf Configuration File

Add or edit these attributes in the **aopd.conf** configuration file. See Appendix B, "Infoprint Server Files" on page 245 for more information about all of the attributes in the configuration file and rules for editing the file.

start-daemons = { lpd }
To start the LPD, add **lpd** to the values in this attribute before issuing the **aopstart** command. Enclose the values in braces. By default, the Printer Inventory Manager and the LPD start.

Default: `start-daemons = {lpd}`

lpd-port-number = *nnnn*
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 (see Note 1.) Windows users must specify this port number when they configure the Infoprint Port Monitor for Windows. This attribute is optional.

Rule: This port number must not be reserved in the *hlq*.PROFILE.TCPIP data set. See Editing the *hlq*.PROFILE.TCPIP Data Set on page 51 for more information about the PORT statement.

Recommendation: Assign port 515 to the Print Interface LPD so that you can use all of the functions that Infoprint Server provides. The following limitations apply if you assign a port other than port 515:

- Remote users will not be able to use commands such as LPR and **enq** to submit print requests. This is because you *cannot* configure most LPR clients to send print requests to a port other than to the well-known port 515. The Infoprint Port Monitor for Windows, however, is an exception; you can configure the Infoprint Port Monitor to send print requests to any port.
- IP PrintWay *cannot* use transforms provided by Infoprint Server Transforms to transform data using the IP PrintWay resubmit for filtering function. This is because IP PrintWay always resubmits data to Print Interface at port 515.

Default: `lpd-port-number = 515`

Using the LPD Compatibility Filter

The LPD compatibility filter, **lpd_compat.so**, provides support for some LPD 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 client 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 the following publications for more information:

- *z/OS Infoprint Server Operation and Administration* for information about how to specify the LPD compatibility filter in a printer definition.
- *z/OS Infoprint Server User's Guide* for information about how to specify LPD compatibility filter options in the **filter-options** job attribute.

Customizing LPR Clients

To submit a print request from LPR clients to the Print Interface LPD, a user typically uses a TCP/IP command such as the LPR command. The Infoprint Port Monitor for Windows also uses the LPR protocol to submit a print request to the Print Interface LPD.

You must provide users of LPR commands and the Infoprint Port Monitor for Windows with the following information:

- The IP address or host name of the z/OS system where the Print Interface LPD is running. If you run more than one TCP/IP stack, this is the IP address or host name of the stack where the Print Interface LPD is running.
- The port number at which the Print Interface LPD is listening. The port is the value you specified in **lpd-port-number** attribute in the **aopd.conf** file (default 515).
- The name of the printer definition for the target printer defined in the Printer Inventory. This name is case sensitive.
- 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.
- 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.

For additional information about how to submit jobs from remote systems and how to configure the Infoprint Port Monitor for Windows refer to *z/OS Infoprint Server User's Guide*.

Sending the Control File First

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.

"Increasing the Time-Out Value on an AIX System" on page 66 shows an example of how to configure an AIX print queue to send the control file first.

Increasing the Time-Out Value on an AIX System

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 acknowledgment 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.

On an AIX system, the **rembak** command, which is used by the AIX queuing subsystem to send jobs to remote printers, uses a default time-out value of 90 seconds. IBM recommends that you change this default to a long time-out value when sending files to Infoprint Server, especially when you send large files that need to be transformed. To change the value, specify the **-T** option on the **rembak** command. You can specify the **-T** option in the **/etc/qconfig** file in the **backend** option under the queue device name.

The following example specifies a timeout value of 5 minutes. This example also shows the following options:

- -X option, which lets users send Infoprint Server job attributes on the **enq -o** option.
- -C option, which sends the control file first, before the data.

```
backend = /usr/lib/lpd/rembak -T 5 -X -C
```

Once you have made this change, delete the **/etc/qconfig.bin** file. Then use the following commands to stop and restart the queue daemon:

```
stopsrc -s qdaemon
startsrc -s qdaemon
```

Customizing Print Interface Support for the z/OS SMB Server

Print Interface, in conjunction with support provided by the z/OS SMB server, lets users print from Windows workstations using the standard Windows Server Message Block (SMB) protocol. Users can print to any printer defined in the Printer Inventory. Print Interface validates that the document can print on the target printer, optionally transforms data from one format to another, and allocates output data sets on the JES spool. Print Interface can also provide job status to the z/OS SMB server.

Note: As an alternative to using SMB protocol, users can use the Internet Printing Protocol or the Infoprint Port Monitor for Windows. See Chapter 9, “Installing and Customizing the Infoprint Server Windows Client” on page 237 for a comparison that can help you decide which method to use to print from Windows systems.

This section describes the following task to customize SMB support:

Task	Condition	See Page:
Customizing the z/OS SMB Server	Required	67

Related task: After you customize the z/OS SMB server, the administrator must create a printer definition in the Printer Inventory for each target printer and specify

printer attributes required by Print Interface. Refer to *z/OS Infoprint Server Operation and Administration* for information. The operator does *not* need to restart Infoprint Server.

Customizing the z/OS SMB Server

Before users can print using SMB protocol, you must customize the z/OS SMB server as described in *z/OS Distributed File Service SMB Administration*. You must define a *shared* printer to the z/OS SMB server for each printer definition in the Printer Inventory to which Windows users need to print. When you create a shared printer, consider the following:

- You must specify a name for each shared printer. You can use either the same name as the name of the printer definition or a different name. Using the same name can make it easier to locate the printer in the Printer Inventory.
- You can also specify the printer type when you create a shared printer. The printer type designates the printer driver for the printer, which must reside on the client's system. You can download IBM printer drivers from the IBM Printing Systems Division Web site: http://www.ibm.com/download.html.

  If the target printer is an IBM AFP printer, you can download the IBM AFP printer driver to each workstation and specify the AFP printer driver for Windows. As an alternative, if the administrator has configured printer definitions in the Printer Inventory to use the PCL or PostScript (and PDF) to AFP data transforms, then you can specify a generic PCL or PostScript driver for AFP printers. Refer to *z/OS Infoprint Server Operation and Administration* for information about how to configure printer definitions so that Print Interface uses data transforms.

Customizing the Print Interface IPP Server

Print Interface provides an Internet Printing Protocol (IPP) server, which accepts print requests from any client that uses the IPP protocol. The Print Interface IPP server supports the IPP job attributes sent with the print requests; see Table 5 on page 71 for the job attributes supported by the IPP server. The IPP server can also provide job status to the IPP client. The IPP server does *not*, however, support canceling a print request.

Note: 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.

This section describes the following required and optional tasks to customize the IPP server:

Task	Condition	See Page:
Ensuring Sufficient Memory is Available to Start the IPP Server	Required	68
Editing the aopd.conf Configuration File	Required	68
Setting Environment Variables for the IPP Server	Optional: To change default configuration	69
APF-Authorizing Java Run-Time Libraries	Required	69
Customizing the IPP Workstation Client	Required	70

Related tasks: After you customize the Print Interface IPP server, the administrator and operator must perform the following tasks. Refer to *z/OS Infoprint Server Operation and Administration* for information about these tasks:

- The administrator must create a printer definition in the Printer Inventory for each printer and specify fields in the printer definition that Print Interface requires.
- The operator must run the **aopstart** command or the AOPSTART procedure to start the IPP server daemon.

Ensuring Sufficient Memory is Available to Start the IPP Server

The IPP server requires at least 100 megabytes (MB) of memory. 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:

- 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(104857600)    /* 100*1024*1024 = 100MB (minimum)       */
  ```

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

  ```
  SETOMVS MAXASSIZE=104857600
  ```

  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 104857600.
- 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 100MB 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=100M or higher. If the REGION parameter is not specified, the default region size defined for your installation is used.

Editing the aopd.conf Configuration File

Add or edit the following attributes in the **aopd.conf** configuration file. See Appendix B, "Infoprint Server Files" on page 245 for more information about all of the attributes in the configuration file and rules for editing the file.

ipp-port-number = *nnnn*

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.

Note: This port number must *not* be reserved in the *hlq*.PROFILE.TCPIP data set. See "Customizing z/OS Communications Server" on page 51 for more information.

Default: `ipp-port-number = 631`

start-daemons = { ippd }

To start the IPP server daemon, add **ippd** to any other values in this attribute before issuing the **aopstart** command. Enclose the values in braces. By default, only the Printer Inventory Manager and the LPD start. This attribute is required to start the IPP server.

Default: `start-daemons = {lpd}`

Example: `start-daemons = {lpd ippd}`

Setting Environment Variables for the IPP Server

The following environment variables affect the behavior of the IPP server. If you need to change the default values of these environment variables, specify them in the **aopstart** EXEC. See "Editing the **aopstart** EXEC" on page 36 for more information.

CLASSPATH

The full path names of Infoprint Server IPP files. If you installed Infoprint Server IPP files in the default directory, **/usr/lpp/Printsrv**, you do *not* need to modify this environment variable. If you installed Infoprint Server files in a different directory, specify the following values:

- ***directory*/classes/ipp.jar**
- ***directory*/classes/ippserver.jar**
- ***directory*/classes/ippreal.jar**

where *directory* is the directory that contains the files.

Default:

```
/usr/lpp/Printsrv/classes/ipp.jar:
/usr/lpp/Printsrv/classes/ippserver.jar:
/usr/lpp/Printsrv/classes/ippreal.jar
```

JAVA_HOME

The path used to locate Java 1.3 files. This environment variable is *optional*; however, you must set this environment variable if you did *not* install Java 1.3 files in the default Java directories.

Default: `/usr/lpp/java/IBM/J1.3`

APF-Authorizing Java Run-Time Libraries

To use the IPP server, you must mark Java run-time libraries APF-authorized. To authorize libraries, you can use the sample job AOPJAUTH or the z/OS UNIX **extattr** command.

Note: When maintenance is applied to Java, APF authorization is lost; therefore, you must rerun AOPJAUTH or the **extattr** command after maintenance is applied to Java.

Running AOPJAUTH to Authorize Java Libraries

Infoprint Server provides sample job AOPJAUTH in SYS1.SAMPLIB, which you can use to APF-authorize Java libraries. AOPJAUTH authorizes all Java 1.3 libraries that are installed in default Java directories on your z/OS system.

Guidelines:

- To run AOPJAUTH, you must have at least READ access to the BPX.FILEATTR.APF FACILITY class profile. Also, you must have superuser authority (UID of 0) or access to the BPX.SUPERUSER FACILITY class profile.

- If you did *not* install Java in the default Java directories, edit the AOPJAUTH procedure to specify the directories in which you installed Java libraries.
- If you edit AOPJAUTH, do not modify the PARM parameter for the AOPBATCH program on the EXEC statement. Also, because the STDIN DD statement contains executable z/OS UNIX commands, make sure that these lines do not contain sequence numbers in columns 73 - 80. If you use the ISPF editor to edit AOPJAUTH, set sequence numbers off by typing NUMBER OFF on the command line before typing any data. If sequence numbers already exist, type UNNUM to remove them and set number mode off. See Appendix D, "AOPBATCH Program" on page 271 if you need more information about how to use AOPBATCH.

Using the extattr Command to Authorize Java Libraries

Instead of using AOPJAUTH, you can use the z/OS UNIX **extattr** command to APF-authorize Java libraries. Refer to *z/OS UNIX System Services Command Reference* for information about this command.

If you run Java 1.3, authorize all libraries with an extension of **so** in the following Java directories:

- **/usr/lpp/java/IBM/J1.3/bin**
- **/usr/lpp/java/IBM/J1.3/bin/classic**

Customizing the IPP Workstation Client

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.

The user must first define the 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. The user must also specify the Universal Resource Indicator (URI) of the printer. The URI of any printer defined in the Printer Inventory has the following format:

http://*host*:*port*/servlet/IPPServlet/*printername*

where:

host The host name or IP address of the z/OS system.

port The port number where the IPP server is listening. Specify the same number as specified in the **ipp-port-number** attribute in the **aopd.conf** file. If the **ipp-port-number** attribute is not set, specify 631, which is the default value for the attribute.

servlet/IPPServlet
Qualifiers required by the Print Interface IPP server; these qualifiers are case sensitive.

printername The name of the printer definition in the Printer Inventory. The printer definition name is case sensitive; you must use the same uppercase and lowercase letters as are stored in the Printer Inventory.

Example: If the IP address of the z/OS system is 9.99.1234, the IPP server is listening at port 631, and the name of the printer definition is ″myprinter″, use this URI:

```
http://9.99.1234:631/servlet/IPPServlet/myprinter
```

Notes:

1. Some IPP clients might also support the **ipp** format of the URI. In this case, the URI can have the following format:

   ipp://*host*[:*port*]/servlet/IPPServlet/*printername*

   When you use the **ipp** format for the URI and omit the port, the default is 631, which is also the default port at which the Print Interface IPP server listens.

   Example: If the IP address of the z/OS system is 9.99.1234, the IPP server is listening at port 631, and the name of the printer definition is ″myprinter″, use this URI:

   ```
   ipp://9.99.1234/servlet/IPPServlet/myprinter
   ```

2. For detailed information about how to add an Internet printer to a Windows 2000 workstation, refer to *z/OS Infoprint Server User's Guide*.
3. The Print Interface IPP server cannot receive data that has been encrypted; therefore, the user should ensure that the IPP client does not encrypt data.

An IPP client can specify IPP job attributes with a print request. The Print Interface IPP server supports some of the IPP job attributes that have corresponding Infoprint Server job attributes. Table 5 lists the IPP job attributes that the Print Interface IPP server supports for all printers and the corresponding Infoprint Server job attribute that it sets. Refer to *z/OS Infoprint Server User's Guide* for information about these Infoprint Server job attributes.

Table 5. IPP Attributes and Corresponding Infoprint Server Attributes

IPP Job Attribute	Corresponding Infoprint Server Job Attribute
copies	**copies**
job-name	**title-text**
job-priority	**jes-priority**
requesting-user-name	**owner**[1]

1. JES does not permit the IPP server to allocate data sets on the JES spool with the owner name that is specified in the IPP job attribute; therefore, to make the owner name available in SMF records and in the data set name displayed by SDSF, the IPP server uses the owner name as the job name when allocating data sets on the JES spool.

Customizing SAP R/3

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.

Note: This support means that you can 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.

Depending on where your SAP R/3 application servers run, different Print Interface components handle the print requests and provide different functions:

- An SAP R/3 application server runs on the same z/OS system as Infoprint Server

  When an SAP R/3 application server runs on the same z/OS system as Infoprint Server, the SAP Output Management System (OMS) provided with Print Interface

handles print requests. The SAP OMS lets users cancel jobs and obtain accurate job status. The SAP Callback daemon provided with Print Interface provides immediate notification when documents have completed successfully or unsuccessfully. The Callback daemon also can provide notification to SAP R/3 applications running on other systems, provided that the SAP R/3 spool work process runs on the same z/OS system as Infoprint Server. Figure 4 on page 9 shows this configuration.

- An SAP R/3 application server does not run on the same z/OS system as Infoprint Server

  When no SAP R/3 application server runs on the same z/OS system as Infoprint Server, the Print Interface LPD handles print requests. Users *cannot* receive notification about their print requests or cancel jobs. Users can receive job status; however, the job status is not as accurate as when the SAP R/3 application server runs on the same z/OS system as Infoprint Server and the Infoprint Server OMS is used. Figure 5 on page 11 shows this configuration.

The following table summarizes the customization tasks when an SAP R/3 application server runs on the *same* z/OS system as Infoprint Server:

Task	Condition	See Page:
Defining an SAP User for the Callback Daemon	Required	73
Creating the SAP Callback Daemon Configuration File (**aopsapd.conf**)	Required	73
Setting Environment Variables for the SAP OMS	Required	75
Defining a ROMS and LOMS to the SAP R/3 Application Server	Required	75
Modifying the LOMS to Specify Job Attributes	Optional: To specify Infoprint Server job attributes	76
Creating an SAP R/3 Output Device	Required	77
Customizing the SAP to AFP Transform	Optional: To print SAP OTF and ABAP documents on IBM AFP printers	102

The following table summarizes the customization tasks when an SAP R/3 application server does *not* run on the same system as Infoprint Server. For example, the application server might run on an AIX or Windows NT system.

Task	Condition	See Page:
Creating an SAP R/3 Output Device	Required	77
Customizing the Print Interface LPD	Required	61
Customizing the SAP to AFP Transform	Optional: To print SAP OTF and ABAP documents on IBM AFP printers	102

Related tasks: After you customize Print Interface for printing from SAP R/3, the administrator and operator must perform the following tasks. Refer to *z/OS Infoprint Server Operation and Administration* for information about these tasks:

- The administrator must create a printer definition in the Printer Inventory for each printer. In the printer definition, specify fields that Print Interface requires. If the printer is an IBM AFP printer, also specify the SAP to AFP transform filter in the printer definition.
- If you need the SAP Callback daemon, the operator must stop and restart all Infoprint Server daemons. The SAP Callback daemon, **aopsapd**, starts automatically when the first print request is received from the SAP R/3 Application Server for z/OS.

Defining an SAP User for the Callback Daemon

You must define an SAP user that the Print Interface SAP Callback daemon can use to log on to the SAP R/3 Application Server for z/OS. Define an SAP user to each SAP R/3 application server at which you want to receive job callback notification. To define a user for the Callback daemon, follow these steps:

1. From the SAP R/3 GUI, select **Tools --> Admininstration --> User maintenance --> Users**.
2. The user must be a CPIC type with XOMI permissions to log on the CCMS system management XOM application programming interface. You can provide these authorizations in the S_XMI_XOM_A profile.

Note: If you change the SAP user, make the same changes in the **aopsapd.conf** configuration file, and use the **aopstart** and **aopstop** commands to restart Infoprint Server to pick up the change in the configuration file.

Creating the SAP Callback Daemon Configuration File (aopsapd.conf)

The SAP Callback daemon configuration file, **aopsapd.conf**, contains information that the SAP Callback daemon needs to log on to the SAP R/3 application servers. This file is required.

To create the configuration file, follow these steps:

1. Copy the sample configuration file, **/usr/lpp/Printsrv/samples/aopsapd.conf**, to **/etc/Printsrv/aopsapd.conf**. For example, the following z/OS UNIX command copies the sample configuration file to the default location:

   ```
   cp /usr/lpp/Printsrv/samples/aopsapd.conf  /etc/Printsrv/aopsapd.conf
   ```

   Tip: A sample job is provided in SYS1.SAMPLIB(AOPCPETC) to copy all Infoprint Server sample configuration files from **/usr/lpp/Printsrv/samples** to the **/etc/Printsrv** directory, including the SAP Callback configuration file. Review the comments in the sample job to determine which configuration files your installation requires, and then run the job to create the configuration files.
2. Because this file contains information on how to log on to the SAP R/3 application servers, set the permissions so that only administrators can read the file. Set the owner to the SAP administration ID and set permissions to 0700. Use the **chown** and **chmod** commands to define the owner and set permissions. For example, if the SAP administration ID is **mike**, enter these commands:

   ```
   chown mike /etc/Printsrv/aopsapd.conf
   chmod 0700 /etc/Printsrv/aopsapd.conf
   ```
3. Edit the file for your installation. You can use either the TSO/E OEDIT command or the z/OS UNIX **oedit** command. For example, enter the following z/OS UNIX command:

   ```
   oedit /etc/Printsrv/aopsapd.conf
   ```

4. If you edit this file while Infoprint Server is running, stop and restart all Infoprint Server daemons using these commands:
   a. Use the **aopstop** command to stop all daemons.
   b. Use the **aopstart** command to restart all daemons.
5. If you do *not* create the configuration file in directory **/etc/Printsrv/**, specify the full path name of the configuration file in the **AOPSAPD_CONF** environment variable.

Format of an Entry in the SAP Callback Daemon Configuration File

In the **aopsapd.conf** file, create one entry for each SAP R/3 system that contains an Infoprint Server job callback LOMS. An entry consists of a set of attributes, starting with the **system** attribute and ending with a semicolon. All attributes are required.

*comment*
system *system-name*
    initial-target = *instance-name*
    client = *clientname*
    userid = *userid*
    password = *password*
    language = *language*
;

system *system-name*
: The name of the SAP R/3 system that contains an LOMS for which you want callback notification. The SAP R/3 system can run on any z/OS or non-z/OS system. This attribute must be the first attribute in the entry.

initial-target = *instance-name*
: The name of an SAP R/3 server instance on the SAP R/3 system defined in the **system** attribute. The Print Interface Callback daemon initially contacts this server instance to obtain configuration information about the LOMSs defined on that SAP R/3 system.

client = *clientname*
: The name of the client defined to SAP for use by the Callback daemon.

userid = *userid*
: The user ID defined to SAP for use by the Callback daemon.

password = *password*
: The password defined to SAP for use by the user ID.

language = *language*
: The language used for messages from the SAP R/3 application server.

Note: The client name, user ID, and password must match the values specified when you defined an SAP user for the OMS Callback daemon; see “Defining an SAP User for the Callback Daemon” on page 73.

Example

This example shows two sample entries in the **aopsapd.conf** configuration file.

```
system EZ0
   initial-target = boefcs1_EZ0_00
   client = 000
   userid = mike
   password = xxxxxxx
   language = EN
;
```

```
system EX0
   initial-target = boefcs5_EX0_00
   client = 000
   userid = mike
   password = xxxxxxx
   language = EN
;
```

Syntax Rules

When editing the SAP Callback daemon configuration file, follow these syntax rules:

- End the last attribute in each 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.
- 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.

Setting Environment Variables for the SAP OMS

The following environment variables affect the behavior of the SAP OMS. Set these environment variables in the **aopstart** EXEC. See “Editing the **aopstart** EXEC” on page 36 for information.

AOPSAPD_CONF
: Specify the full path name of the SAP Callback daemon configuration file. If the configuration file is in **/etc/Printsrv/aopsapd.conf**, you do not need to set this environment variable.

  Default: /etc/Printsrv/aopsapd.conf

LIBPATH
: Add the directory that contains the SAP remote function calls DLL (**librfc.dll**).

Defining a ROMS and LOMS to the SAP R/3 Application Server

To define the Infoprint Server Output Management System (OMS) to an SAP R/3 application server, the SAP administrator must define the following entries:

- An SAP Real Output Management System (ROMS)

  The ROMS defines the general capabilities of the OMS and global configuration information.
- One or more SAP Logical Output Management Systems (LOMS)

  An LOMS lets you customize the capabilities for a particular set of SAP output devices.

To define a ROMS and two LOMSs that interface with Infoprint Server, import sample file **/usr/lpp/Printsrv/samples/infoprint.oms** into SAP R/3. This sample file defines a ROMS with this name and description:

- INFOPRINT: IBM Infoprint Server for OS/390

It creates two LOMSs with these names and descriptions:

- IPS001: Infoprint Server with callback: This LOMS provides job callback notification.
- IPS002: Infoprint Server with polling: This LOMS does not provide job callback notification.

To import the file, follow these steps. These steps might be different on your SAP GUI:

1. Log on to the SAP R/3 Application Server for z/OS.
2. Select **Tools** --> **CCMS** --> **Spool** --> **SPAD --> Spool administration**.
3. On the Spool Administration screen, select **Utilities** --> **To output management system** --> **Import**.
4. On the Spool: Saving and Loading of Definitions screen:
   - Type **/usr/lpp/Printsrv/samples/infoprint.oms** in the **Export/Import file name** field.
   - Click the **Import** button.
   - Select the **Execute import** and **Generate log** options.

After you import the ROMS and LOMs, follow these steps to edit them:

1. Edit the INFOPRINT ROMS. For the initialization instance, select the SAP R/3 Application Server for z/OS that runs on the same z/OS system as Infoprint Server.
2. Edit the IPS001 and IPS002 LOMSs. For the tasking target, select the SAP R/3 Application Server for z/OS that runs on the same z/OS system as Infoprint Server.
3. Edit the IPS001 LOMS. For the callback target instance, specify any server instance within the SAP R/3 system.

Modifying the LOMS to Specify Job Attributes

The Infoprint Server LOMS contains **aopoms** commands that perform the SUBMIT, DPOLL, DQUERY, JQUERY and CANCEL OMS commands. If you want to specify Infoprint Server job attributes for print requests, you can modify the **aopoms** command that performs the SUBMIT command. For example, you can specify the **duplex** job attribute to obtain duplex printing on an IBM AFP printer.

The format of the **aopoms** command that performs the SUBMIT command is:

```
aopoms -c submit -d printer -j sap-spoolid -r sap-rmg -s sap-system
        -T sap-callback-target [-t title] [-n copies] [-o options] file ...
```

The options are:

-d *printer*
: Name of the printer definition for the target printer.

-j *sap-spoolid*
: SAP R/3 spool ID to be stored in the job entry.

-n *copies*
: Number of copies to be printed, from 1 to 255.

-o *option*
: One or more attribute value assignments in the format: *attribute=value*. Separate each attribute value assignment with spaces. You can specify any job attributes defined in the *z/OS Infoprint Server User's Guide*.

-r *sap-rmg*
: The SAP R/3 **rmg** ID to be stored in the job entry.

-s *sap-system*
: The name of the SAP R/3 system.

-T *sap-callback-target*
The name of the SAP R/3 server that is the target server for job callbacks.

-t *title* A title for the document. This option is equivalent to Infoprint Server **title** job attribute.

The **aopoms** commands are defined in the LOMS as templates with replacement variables used to supply specific information. The generic form of the **aopoms** SUBMIT command as defined in the Infoprint Server LOMS is:

```
aopoms -c submit -j &EI -r &EG -s &Es -T &ES  -d &P -t""&T"" -n &C &F
```

To specify Infoprint Server job attributes, for example the **jes-priority** job attribute, specify the following **aopoms** command on one line:

```
aopoms -c submit -j &EI -r &EG -s &Es -T &ES -d &P -t ""&T"" -n
&C -o "jes-priority = 4" &F
```

Creating an SAP R/3 Output Device

To print from SAP R/3, the SAP R/3 administrator must create an SAP R/3 output device for each target printer. You might want to create more than one SAP R/3 output device for the same printer. For example, you might want to customize the LOMS differently for each SAP output device, or you might want to associate each SAP output device with a different printer definition for the same target printer. (The administrator can create more than one printer definition in the Printer Inventory for the same printer in order to specify different job characteristics, such as a different form definition name.)

To create an SAP R/3 output device, you need the following information:

- The name of the SAP R/3 Application Server for z/OS.
- The name of the printer definition in the Printer Inventory. This name is case sensitive.
- The data format accepted by the printer, for example, AFP, PCL, or PostScript. The data format determines the device driver you select.
- The name of the LOMS defined to SAP R/3. Default names for Infoprint Server LOMSs are:
  - **Infoprint Server with callback**
    Use this LOMS if you want immediate notification of job events.
  - **Infoprint Server with polling**
    Use this LOMS if you do not need immediate notification of job events. The Callback daemon returns notification only when polled by the SAP R/3 system.
- The host spool access method:
  - Use access method **E** when the SAP application server runs on the same system as Infoprint Server
  - Use access method **U** when the SAP application server runs on a different system from Infoprint Server
- For access method **U**, you also need:
  - The host name or IP address of the z/OS system where Infoprint Server runs.
  - The port number where the Print Interface LPD listens (default port is 515).

The following sections contain instructions for creating an SAP output device. Use the instructions for the correct host spool access method: **E** or **U**. The screen and field names in the instructions might be different on your SAP R/3 GUI.

Creating an SAP Output Device Using Access Method E

If the SAP R/3 application server runs on the same z/OS system as Infoprint Server, follow these steps to create an SAP R/3 output device from the SAP R/3 GUI:

1. On the Spool Administration: Initial Screen, click the **Output devices** button.
2. Enter change mode by pressing F8 or clicking the Change icon (a pencil).
3. On the Spool Administration: List of Output Devices screen, select **Output device --> Create**.
4. On the Spool Administration: Create Output Device screen, fill in these fields:
   - **Output device**: Type a name for the printer.
   - **Short name**: Type a short name.
   - **Device Type**: Select a printer driver that generates a data stream accepted by the printer:
     - For IBM AFP printers, select **SAPGOF E: Generic output format EBCDIC**.
     - For other printers, select either PCL or PostScript.
   - **Spool server**: Select the name of the SAP R/3 application server.
5. On the HostSpoolAccMethod tab, fill in these fields:
   - **Host spool access method**: Select **E: External output management system**
   - **Logical OMS**: Select an Infoprint Server LOMS.
   - **Host printer**: Type the name of the printer definition in the Printer Inventory.
6. Save your changes.

Creating an SAP Output Device Using Access Method U

If the SAP R/3 application server runs on a non-z/OS system, follow these steps to create an SAP R/3 output device from the SAP R/3 GUI:

1. On the Spool Administration: Initial Screen, click the **Output devices** button.
2. Enter change mode by pressing F8 or clicking the Change icon (a red pencil).
3. On the Spool Administration: List of Output Devices screen, select **Output device --> Create**.
4. On the Spool Administration: Create Output Device screen, fill in these fields:
   - **Output device**: Type a name for the printer.
   - **Short name**: Type a short name for the printer.
   - **Device Type**: Select a printer driver that generates a data stream accepted by the printer:
     - For IBM AFP printers, select **SAPGOF E: Generic output format EBCDIC**.
     - For other printers, select either PCL or PostScript.
   - **Spool server**: Select the name of the SAP R/3 application server.
5. On the HostSpoolAccMethod tab, fill in these fields:
   - **Host spool access method**: Select **U: Print using Berkley protocol**.
   - **Logical OMS**: Select an Infoprint Server LOMS.
   - **Host printer**: Type the name of the printer definition in the Printer Inventory.
   - **Destination host**: Type the address of the z/OS system.
   - **Port number**: Type the port where the Infoprint Server LPD listens.
6. Save your changes.

Viewing Job Status and Job Events, Querying Jobs, Canceling Jobs

After you create an SAP R/3 output device, SAP R/3 users can print to an Infoprint Server printer, view job status, view job events, query jobs, and cancel jobs. The following considerations apply when users make the following print requests:

- View current information and all events for a job:
  - If the status indicates the print job has completed, the print job might still remain on the JES spool.
  - If the status indicates the print job is waiting in the host spooler, the position in the queue does not reflect the order in which the job will be printed. For example, jobs not submitted through Infoprint Server might print before the job.
- Display device information:
  - Print jobs submitted to the same printer using a different printer definition name in the Printer Inventory are not listed.
  - The position of the job in the list does not reflect the order in which the job will be printed.
- Cancel a print job

  Users can cancel a print job only if Infoprint Server has not yet selected it for processing.

Customizing the Print Interface Subsystem

The Print Interface subsystem can process output data created by z/OS batch jobs. To request that the Print Interface subsystem process an output data set, the job submitter specifies the name of the Print Interface subsystem in the SUBSYS parameter on the DD JCL statement.

The Print Interface subsystem validates that a data set can print on the printer and calls the data stream transform, or other filter, requested in the printer definition. The subsystem then allocates an output data set on the JES spool using job attributes and JCL parameters specified on the DD JCL statement, the associated OUTPUT JCL statement, and values specified in the Allocation section of the printer definition. Each output data set is placed in a separate JES output subgroup. The output data set can be printed on any printer, including printers controlled by PSF for OS/390 and by IP PrintWay.

The following table summarizes the customization tasks for the Print Interface subsystem. These tasks are required by all installations that want to use the Print Interface subsystem.

Task	Condition	See Page:
Editing the aopd.conf Configuration File	Required	80
Increasing the Maximum Number of Active Transform Daemons	Optional	80

Related tasks: After you customize the Print Interface subsystem, the administrator and operator must perform the following tasks. Refer to *z/OS Infoprint Server Operation and Administration* for information about these tasks:

- The administrator must create a printer definition in the Printer Inventory for each printer, or for each set of related printers, and specify fields in the printer definition that Print Interface and the transforms require.

- The operator must run the **aopstart** command or the AOPSTART procedure to start the Print Interface subsystem daemon.

Editing the aopd.conf Configuration File

Before you can use the Print Interface subsystem, you must add or edit the following attribute in the **aopd.conf** configuration file. See Appendix B, “Infoprint Server Files” on page 245 for more information about all of the attributes in the configuration file and rules for editing the file.

start-daemons = { subd }
To start the Print Interface subsystem daemon, add **subd** to the values in this attribute before you start the Print Interface subsystem daemon. Enclose the values in braces. By default, the Printer Inventory Manager and the LPD start.

Default: `start-daemons = {lpd}`

Example: In this example, the Printer Inventory Manager, the Transform Manager, and the Print Interface subsystem start:

```
start-daemons = {xfd subd}
```

The following attribute in the **aopd.conf** configuration file has special considerations when you use the Print Interface subsystem:

inventory = *inventory*
The name of both the Printer Inventory and the Print Interface subsystem. The name must contain exactly four letters or numbers. The first character must be an uppercase letter; it *cannot* be a lowercase letter or numeric. The job submitter specifies this name in the SUBSYS parameter on the DD JCL statement.

Because this name is also the name of the Printer Inventory, if you change this name after the Printer Inventory Manager has been started, you must stop and restart all Infoprint Server daemons. You must also stop and restart all other components and products that specify this Inventory name in their startup procedures. You must stop and restart NetSpool and IP PrintWay. If the PSF startup procedure specifies the name of the Printer Inventory, you must also stop and restart PSF for OS/390.

Default: `inventory = AOP1`

Increasing the Maximum Number of Active Transform Daemons

If your installation limits the number of transform daemons that can be active at one time, you might need to specify a higher number if jobs that use the Print Interface subsystem take a long time to complete or hang until they are canceled by the operator. For example, the following situations can occur:

- If you limit the number of AFP to PCL transform daemons to one, then a job that writes AFP output to two data sets in the same job step will not complete if both data sets require an AFP to PCL transform daemon.
- If you limit the number of AFP to PCL transform daemons to two, then two jobs that each write AFP output to two data sets in the same job step might not complete if each job reserves an AFP to PCL transform daemon.

Also, other jobs that require the AFP to PCL transform will not be able to complete.

To increase the maximum number of active transforms:

1. Edit the **maximum-active** attributes in the transform configuration file, **aopxfd.conf**.

2. Stop and restart the Transform Manager daemon to pick up the changes.

   Example:

   ```
   aopstop -d xfd
   aopstart
   ```

For more information about the transform configuration file, see “aopxfd.conf -- Infoprint Server Transform Configuration File” on page 249.

Chapter 5. Customizing Infoprint Server Transforms and the Transform Manager

This chapter describes how to customize Infoprint Server Transforms and the Transform Manager component of Infoprint Server. You must customize Infoprint Server Transforms and the Transform Manager in order to use data stream transforms.

See "Infoprint Server Transforms" on page 12 for an overview of Infoprint Server Transforms; see "Transform Manager" on page 13 for an overview of the Transform Manager. Also, refer to *z/OS Infoprint Server User's Guide* for information the capabilities and limitations of each transform.

Infoprint Server Transforms, a separate IBM program product, provides data stream 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 R/3 Release 4.6C (and lower) Output Text Format (OTF) and Advanced Business Application Programming (ABAP) to AFP

Infoprint Server Transforms also provides data stream transforms that convert Advanced Function Presentation (AFP) and line data streams to the following data streams for printing on non-AFP printers or viewing 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:

- Kanji AFP Print feature. This 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.
- Coax Printer Support feature. This feature lets IP PrintWay print to VTAM-controlled printers. If you install this feature, you must enable Infoprint Server Transforms as described in "Installing and Enabling Infoprint Server Transforms" on page 86. The other customization tasks described in this chapter do *not* apply to this feature. See Chapter 8, "Customizing IP PrintWay" on page 163 for the customization tasks required to print to VTAM-controlled printers.

Transforms can be invoked in several ways:

- Print Interface can automatically invoke a transform before it writes data to the JES spool. The administrator specifies the transform in the printer definition.
- IP PrintWay can automatically invoke Print Interface to transform data in batch jobs submitted directly to IP PrintWay with a JCL statement.
- Job submitters can invoke transforms directly using the z/OS UNIX transform commands, such as **pcl2afp** and **ps2afp**.

© Copyright IBM Corp. 1995, 2002

The Transform Manager component of Infoprint Server manages those transforms that are implemented as daemons. Table 6 on page 86 lists the transforms provided by Infoprint Server Transforms and indicates whether or not the Transform Manager manages the daemons.

This chapter describes the following customization tasks. Required tasks are required by all installations that want to use Infoprint Server Transforms. Optional tasks are required only if the listed condition applies to your installation.

Task	Condition	See Page:
Identifying Software Requirements	Required	85
Installing and Enabling Infoprint Server Transforms	Required	86
Customizing the Transform Manager	Optional: To use transforms managed by Infoprint Server Transform Manager	86
Customizing the PCL to AFP Transform	Optional: To use PCL to AFP transform	92
Customizing the PostScript to AFP and PDF to AFP Transform[1]	Optional: To use PostScript to AFP and PDF to AFP transforms	95
Customizing the SAP to AFP Transform	Optional: To use SAP to AFP transform	102
Customizing the AFP to PCL, AFP to PDF, and AFP to PostScript Transforms	Optional: To use AFP to PCL, AFP to PDF, and AFP to PostScript transforms	115

1. Customization for the PostScript to AFP and PDF to AFP transforms is the same because the same transform daemon handles both transforms.

The following table lists other components of Infoprint Server that you might need to customize:

Task	Condition	See Page:
Customizing the Printer Inventory Manager	Required	25
Customizing Print Interface	Optional: To transform data automatically before printing.	49
Customizing IP PrintWay	Optional: To transform data sets submitted directly to IP PrintWay using JCL statements.[1]	163

See especially “Specifying a Default Font and Page Definition for Data Transforms” on page 170.

Related tasks: After you customize the Transform Manager and the transforms, the administrator and operator must perform the following tasks. Refer to *z/OS Infoprint Server Operation and Administration* for information about these tasks.

- If you want Print Interface to automatially transform data before printing, the administrator must specify the required transform filters in the **Filter** field of the printer definitions in the Printer Inventory, along with any filter options. For example, the administrator can specify options to specify the type of AFP images to create and the name of the transform class to use.

Select the **Resubmit for filtering** option in the printer definition if you want Print Interface to transform data submitted directly to IP PrintWay with JCL statements.

- If the transforms you plan to use are managed by the Transform Manager, the operator must restart Infoprint Server to start the Transform Manager.

The general user might need to perform the following tasks. Refer to *z/OS Infoprint Server User's Guide* for information about these tasks:

- The job submitter can specify options that customize the transform. For example, the job submitter can specify the transform class to use. The job submitter specifies transform options either in the **filter-options** job attribute or on the transform command (for example, on the **afp2pdf** command).
- To view a PDF file created by the AFP to PDF transform, the user might need to customize the Adobe Acrobat Reader on his system.

Using Infoprint Manager to transform data: Infoprint Server can transform PCL, PostScript, and PDF data streams to AFP format on an AIX or Windows system running IBM Infoprint Manager. See "Identifying Software Requirements" on page 50 for software requirements. To transform data with Infoprint Manager, you do *not* need to customize Infoprint Server Transforms or the Transform Manager. However, the administrator must specify the remote transform filter, **aoprform**, in the printer definition as described in *z/OS Infoprint Server Operation and Administration*.

Identifying Software Requirements

To use the PCL to AFP, PostScript to AFP, PDF to AFP, or SAP to AFP transform, the following additional IBM software beyond the z/OS base elements is required:

- The Transforms to AFP feature of Infoprint Server Transforms (5697-F51) with PTF UW74473
- PSF V3.1 for OS/390 (5655-B17), or later, is required to print on IBM AFP printers.
- The Kanji AFP Print feature of Infoprint Server Transforms is required to print Japanese DBCS PostScript or PDF documents.
- PTF UW82419 is required to transform SAP R/3 Release 4.6C files to AFP format.

To use the AFP to PCL, AFP to PDF, or AFP to PostScript transform, the following additional IBM software beyond the z/OS base elements is required:

- The Transforms to AFP feature of Infoprint Server Transforms (5697-F51) with PTF UW74473
- One of the following features of Infoprint Server Transforms (5697-F51):
  - AFP to PCL transform feature with PTFs UW74487 and UW74488
  - AFP to PostScript transform feature with PTFs UW74487 and UW74489
  - AFP to PDF transform feature with PTFs UW74487 and UW74490
- AFP Font Collection Version 1 (5648-113) or Version 2 (5648-B33) is required to provide 300-pel raster fonts and outline fonts.
- Sonoran Equivalent Fonts PRPQ 8A5061 (5799-FLK) is required to transform documents that contain Sonoran Serif or Sonoran Sans Serif fonts
- An AFP page definition and form definition are required to format error messages issued by the transforms. Other AFP resources that are not included inline in the documents are also required.

- The following Infoprint Server Transforms PTFs are required to specify the AOP_CUTSHEET, AOP_FLATE, and AOP_PJL environment variables; and to specify the A5 paper size in the AOP_PAPER variable:
  - PTF UW79806 for the AFP to PCL transform
  - PTF UW79807 for the AFP to PostScript transform
  - PTF UW79808 for the AFP to PDF transform
- To use the AOP_ANNOTATIONS environment variable, Infoprint Server Transforms PTF UW86279 for the AFP to PDF transform is required.

Note: If your installation has only 240-pel font libraries installed, you can use the AOXCF30 program provided with Infoprint Server Transforms to scale your 240-pel fonts to 300-pel fonts.

Installing and Enabling Infoprint Server Transforms

To install the Infoprint Server Transforms, use the Program Directory that is provided with your product order. Each feature of the Infoprint Server Transforms product has its own Program Directory.

Before you can use the Transforms to AFP feature or any of the other features, your installation 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)
```

For more information on how to use z/OS dynamic enablement, refer to *z/OS MVS Product Management*.

Customizing the Transform Manager

The Transform Manager component of Infoprint Server manages all transforms that are implemented as daemons. Table 6 lists the transforms provided by Infoprint Server Transforms and indicates whether or not the Transform Manager manages the daemons.

Table 6. Infoprint Server Transforms Managed by Transform Manager

Transform	Managed by the Transform Manager
PCL to AFP	Yes
PostScript to AFP	Yes
PDF to AFP	Yes
SAP OTF and SAP ABAP to AFP	No
AFP to PCL	Yes
AFP to PDF	Yes
AFP to PostScript	Yes

To use transforms that are managed by the Infoprint Server Transform Manager, perform the following customization tasks. Required tasks are required by all

installations that want to use Infoprint Server Transforms. Optional tasks are required only if the listed condition applies to your installation.:

Task	Condition	See Page:
Updating the BPXPRMxx Member of SYS1.PARMLIB	Required	87
Editing the Infoprint Server Configuration File (**aopd.conf**)	Required	87
Creating the Transform Configuration File (**aopxfd.conf**)	Required	87
Setting Environment Variables for the Transform Manager	Optional: To change default location of transform configuration file	91

Updating the BPXPRMxx Member of SYS1.PARMLIB

The Transform Manager uses z/OS UNIX domain sockets. Therefore, you must add the FILESYSTYPE statement in the BPXPRMxx member of SYS1.PARMLIB. Following is an example of the FILESYSTYPE statement:

```
FILESYSTYPE TYPE(UDS) ENTRYPOINT(BPXTUINT)
   NETWORK DOMAINNAME(AF_UNIX)
           DOMAINNUMBER(1)
           MAXSOCKETS(200)
           TYPE(UDS)
```

For more information about the FILESYSTYPE statement, refer to *z/OS UNIX System Services Planning*.

Editing the Infoprint Server Configuration File (aopd.conf)

In order to start the Transform Manager, you must add or edit the **start-daemons** attribute in the Infoprint Server configuration file, **aopd.conf**; see Appendix B, “Infoprint Server Files” on page 245 for the complete format of this configuration file.

start-daemons = { xfd }

Add **xfd** to the values in this attribute 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.

Default: `start-daemons = {lpd}`

Example: In this example, the Printer Inventory Manager, the LPD, the Transform Manager, and the Print Interface subsystem daemons start:

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

Creating the Transform Configuration File (aopxfd.conf)

The transform configuration file contains information that the Transform Manager uses to manage transform daemons. You *must* create this file before you start the Transform Manager.

In the transform configuration file, you can specify options and environment variables that affect the tranform. These options and environment variables apply whether Print Interface invokes the transform automatically or whether the user invokes the transform with a transform command.

To create and edit the transform configuration file, use the following z/OS UNIX commands:

1. Switch to an effective UID of 0:

   ```
   su root
   ```

   To use the **su** command, you must be permitted to the BPX.SUPERUSER FACILITY class profile within RACF.

2. Copy the sample configuration file, **/usr/lpp/Printsrv/samples/aopxfd.conf**, to **/etc/Printsrv/aopxfd.conf**:

   ```
   cp /usr/lpp/Printsrv/samples/aopxfd.conf  /etc/Printsrv/aopxfd.conf
   ```

   If you copy the configuration file to another location, specify the full path name of the file in the AOPXFD_CONF environment variable.

   Tip: You can use the sample job provided in SYS1.SAMPLIB(AOPCPETC) to copy all Infoprint Server sample configuration files, including the transform configuration file, from **/usr/lpp/Printsrv/samples** to the **/etc/Printsrv** directory. Review the comments in the sample job to determine which configuration files your installation requires, and then run the job to create the configuration files. We recommend that a user with an effective UID of 0 run this job.

3. Edit the file using your preferred editor:

   ```
   oedit /etc/Printsrv/aopxfd.conf
   ```

   See “Guidelines” for the syntax rules for editing the file.

4. If you change the configuration file while the Transform Manager is running, restart the Transform Manager to pick up the changes:

   ```
   aopstop -d xfd
   aopstart
   ```

Guidelines:

- For a secure environment, we recommend that you do *not* change the owner or permissions of the sample configuration file. We recommend that the file be owned by UID 0 and be writeable only by a user with a UID of 0. To switch to a UID of 0, you can use the z/OS UNIX **su** command if you are permitted to the BPX.SUPERUSER FACILITY class profile within RACF.
- When you edit this file, follow these syntax rules:
  - End the last attribute in each 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.
  - 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.

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 and environment variables. For each transform class, you can specify a different set of options and 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.

To use a transform class, the administrator 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:

  ```
  pcl2afp.dll -c letter_300
  ```

- The job submitter specifies the transform class on the z/OS UNIX transform command in the **-c** option. For example:

  ```
  pcl2afp -c letter_300 -o myfile.afp myfile.pcl
  ```

- The job submitter specifies the transform class in the **filter-options** job attribute on the **lp** command or AOPPRINT procedure. For example:

  ```
  lp -d myprinter -o "filter-options='-c letter_300'" myfile.pcl
  ```

For more information about how the administrator specifies a transform class as a filter option, refer to *z/OS Infoprint Server Operation and Administration*. For more information about how a job submitter specifies a transform class, refer to *z/OS Infoprint Server User's Guide*.

Format of a Transform Entry

In the **aopxfd.conf** file, create one entry for each transform. 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.

This section shows the general format of a transform entry and defines the attributes in the transform entry that are common for all transforms. To create an entry for a specific transform, also see the section for that transform.

*comment*
transform *transformname*[*_transformclass*]
    start-command = *"daemon* [*option*]...*"*
    [**environment** = {*name* –> *value* [*name* –> *value*]...}]
    [**maximum-idle-time** = *seconds*]
    [**minimum-active** = *number*]
    [**maximum-active** = *number*]
;

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`

To specify the **letter_300** transform class on the **pcl2afp** command, type:

```
pcl2afp -c 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.

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 of environment variables with the same names that were set when the **aopstart** command was issued.

Example: `environment = {AOP_RESOURCE_PATH –> /usr/lpp/Printsrv/ps2afp}`

Note: 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. You can specify a job name of 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*.

maximum-idle-time = *seconds*

The number of seconds before the Transform Manager shuts down an idle transform daemon and system resources are freed. The number you specify must be greater than 0.

This attribute is optional; however, to avoid having many transform daemons active, consider specifying either this attribute or the **maximum-active** attribute, or both.

Default: An idle transform daemon is not shut down.

minimum-active = *number*

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 = *number*

The maximum number of transform daemons that the Transform Manager keeps active. The number must be greater than 0 and greater than or equal to the number specified in **minimum-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.

Recommendations:

- If you use the Print Interface subsystem in your installation, do not specify too low a value because jobs that use the subsystem and write multiple data sets might not be able to complete.
- To avoid having many transform daemons active, consider specifying either **maximum-active** or **maximum-idle-time**, or both.

Default: No maximum number; transform daemons are started when needed.

Example

This example shows one sample entry in the **aopxfd.conf** configuration file 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 as a filter option when the transform is invoked.

When the Transform Manager starts, it starts one transform daemon for the PCL to AFP transform. The Transform Manager keeps no more than two transform daemons 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 active.

```
transform pcl2afp
     start-command = "pcl2afpd"
     environment = { _BPX_JOBNAME -> PCL2AFPD }
     minimum-active = 1
     maximum-active = 2
     maximum-idle-time = 300   # 5 minutes
;
```

Setting Environment Variables for the Transform Manager

The following environment variable affects the behavior of the Transform Manager. If you need to set these environment variable, set it in both of the following locations:

- In the **aopstart** EXEC: The Transform Manager, and other Infoprint Server daemons, use environment variables specified in this file. See "Editing the **aopstart** EXEC" on page 36 for information about how to edit this file.
- In the **/etc/profile** file: The z/OS UNIX transform commands, such as **ps2afp**, use environment variables specified in this file. See "Setting Environment Variables for the Printer Inventory Manager" on page 31 for information about how to edit this file.

AOPXFD_CONF
: Full path 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.

MANPATH
: The path of directories that contain the man pages. Infoprint Server man pages are available only in

English. Add **/usr/lpp/Printsrv/man/En_US** to the values in this variable. This variable is *required*.

Note: Add the new directory path *before* **/usr/man/%L** in the MANPATH variable, so that the Infoprint Server versions of the **lp**, **lpstat**, and **cancel** man pages are displayed.

Guideline: Specify the MANPATH environment variable in the **/etc/profile** file if you have not done so already so that users can view the man pages for the z/OS UNIX transform commands.

Customizing the PCL to AFP Transform

To use the PCL to AFP transform provided by Infoprint Server Transforms, perform the following customization task:

Task	Condition	See Page:
Editing the Transform Configuration File (**aopxfd.conf**)	Required	92

Editing the Transform Configuration File (aopxfd.conf)

The transform configuration file contains information that the Transform Manager uses to manage the PCL to AFP transform and also other transforms. See "Creating the Transform Configuration File (**aopxfd.conf**)" on page 87 for more information about this file.

If this configuration file does not already exist, create it before you start the Transform Manager. If you change the configuration file while the Transform Manager is started, stop and restart the Transform Manager to pick up the changes.

Specifying 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. Also see "Transform Classes" on page 88 for more information about how to use transform classes.

Format of a PCL to AFP Transform Entry

In the **aopxfd.conf** file, create one transform entry for the PCL to AFP transform. If you want to define transform classes, create a separate transform entry for each transform class.

transform pcl2afp[_*transformclass*]
  start-command = "pcl2afpd [-m *nnn* {**K**|**M**}]"
  [**environment = {***name* **–>** *value* [*name* **–>** *value*]... **}**]
  [**maximum-idle-time =** *seconds*]

[**minimum-active** = *number*]
[**maximum-active** = *number*]
;

transform pcl2afp[_*transformclass*_**]**
The name of a transform and an optional transform class. Specify from 1 to 63 characters, including letters, numbers, or special characters. **transform** indicates the beginning of an entry.

pcl2afp
The name of the PCL to AFP transform provided by Infoprint Server Transforms.

transformclass
The name of an optional transform class. To use this transform class, the class must be specified when the transform is invoked. Specify a transform class if you want to transform PCL data for printers with different resolutions or paper sizes.

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`

To specify the **letter_300** transform class on the **pcl2afp** command, type:

```
pcl2afp -c letter_300
```

start-command = ″**pcl2afpd [-m** *nnn* **{K|M}]**″
This attribute names the transform daemon and option. Enclose the value in single or double quotation marks if you specify the option.

pcl2afpd
The name of the PCL to AFP 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.

-m *nnn***{K|M}**
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 on.

Use the following algorithm or refer to the following table to determine the amount to initially specify; 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)

The following table shows the amount of storage for different paper sizes and resolutions (240, 300, and 600 pixels per inch), using the previous algorithm:

Paper Size	240	300	600
Letter	2.0M	3.2M	12.6M
Legal	2.6M	4.0M	16.1M
A4	2.1M	3.3M	13.1M

Example: `start-command = "pcl2afpd -m 4M"`

Default: 6M

environment = {*name* **->** *value* **[** *name* **–>** *value***]... }**
Environment variables that define the environment for the PCL to AFP transform. Enclose the environment variables in braces. The values in these environment variables override values of environment variables with the same names that were set when the **aopstart** command was issued.

The following environment variables are optional:

Environment Variable	Meaning
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`

Example: `environment = {AOP_RESOLUTION –> 300}`

maximum-idle-time = *seconds*
See “Format of a Transform Entry” on page 89 for a description.

minimum-active = *number*
See “Format of a Transform Entry” on page 89 for a description.

maximum-active = *number*
See “Format of a Transform Entry” on page 89 for a description.

Example

This example shows two sample entries in the **aopxfd.conf** configuration file:

- 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.

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.

```
 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
;
```

Tip: For additional examples, see the sample configuration file, **/usr/lpp/Printsrv/samples/aopxfd.conf**.

Customizing the PostScript to AFP and PDF to AFP Transform

To use the PostScript to AFP and PDF to AFP transform provided by Infoprint Server Transforms, perform the following customization tasks:

Task	Condition	See Page:
Ensuring Sufficient Memory is Available to Start the Transform	Required	96
Establishing Security for the PostScript and PDF to AFP Transform	Required	97
Editing the Transform Configuration File (**aopxfd.conf**)	Required	98
Customizing PostScript and PDF to AFP Transform Resource Files	Optional: To change default files	101

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 (1) on the **ps2afp** and **pdf2afp** commands and (2) in the **filter-options** job attribute when submitting a print request.

Refer to the following publications for more information:

- *z/OS Infoprint Server Operation and Administration* for information about how to specify the transform DLL and transform options in a printer definition.
- *z/OS Infoprint Server User's Guide* for information about the **ps2afp** and **pdf2afp** commands and the transform options.

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.

Ensuring Sufficient Memory is Available to Start the Transform

The transform requires at least 42 megabytes (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:

- 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 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. 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 **aopstart** 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. See "Format of a PostScript and PDF to AFP Transform Entry" on page 98 for more information.

Establishing Security for the PostScript and PDF to AFP Transform

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 to RACF.

   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 to RACF 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. Refer to *z/OS UNIX System Services Planning* for complete information about how to define z/OS UNIX users to RACF.

   The user profile must have an OMVS segment. Its UID must (1) *not* be 0 and (2) *not* be the default UID, which is defined in the BPX.DEFAULT.USER profile in the RACF 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, type 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, type on the z/OS UNIX command line:

```
ls -l /usr/lpp/Printsrv/bin/ps2afpd
```

Assuming that you assigned user NOBODY as the owner of **ps2afpd**, output from the **ls** command should look like the following:

```
-rwsr-x--- 1 NOBODY ... ...
```

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 Security Server RACF Security Administrator's Guide*. For more information about the **chown**, **chmod**, and **ls** commands, refer to *z/OS UNIX System Services Command Reference*.

Editing 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 "Creating the Transform Configuration File (**aopxfd.conf**)" on page 87 for more information about this file.

If this configuration file does not already exist, create it before you start the Transform Manager. If you change the configuration file while the Transform Manager is started, stop and restart the Transform Manager to pick up the changes.

Specifying Transform Classes for the PostScript and PDF to AFP Transform

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 resouce directory in the AOP_RESOURCE_PATH environment variable. See "Transform Classes" on page 88 for more information about how to use transform classes.

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*]
;

transform ps2afp[_*transformclass*]

This attribute names the transform and an optional transform class. It must be the first attribute in the entry.

ps2afp

The name of the PostScript and PDF to AFP transform provided by Infoprint Server Transforms.

transformclass

The name of an optional transform class. Specify from 1 to 63 characters, including letters, numbers, or special characters. To use this transform class, the class must be specified when the transform is invoked. Specify a transform class if you want to transform PostScript or PDF data using different transform resource files.

Default: If this parameter is omitted, this entry is used only if no class is specified when the transform is invoked.

Example: `transform ps2afp_no_font_substitution`

To use this transform class on the **ps2afp** command, type:

```
ps2afp -c no_font_substitution
```

start-command = "ps2afpd -i *initializationfile***[-M** *nnnnn***{K|M}]"**

This attribute names the transform daemon, the initialization file, and the amount of memory to use. Enclose the value in single or double quotation marks. This attribute is required.

ps2afpd

The name of the PostScript and PDF to AFP transform daemon. 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.

-i *initializationfile*

The name of the initialization file. 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

Example: `start-command = "ps2afpd -i ps2afpe.ps.bin"`

-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 "Ensuring Sufficient Memory is Available to Start the Transform" on page 96 for information about how to specify the region size.

Default: 32M

Example: `start-command = "ps2afpd -i ps2afpe.ps.bin -M 32M"`

environment = {*name* **–>** *value* **[** *name* **–>** *value***]... }**
Environment variables that define the transform environment for the PostScript and PDF to AFP transform. Enclose the environment variables in braces.

The values in these environment variables override environment variables with the same name that were set in the **aopstart** EXEC.

The following environment variable is optional:

Environment Variable	Meaning
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

Example: `environment = {AOP_RESOURCE_PATH –> /etc/Printsrv/ps2afp}`

maximum-idle-time = *seconds*
See "Format of a Transform Entry" on page 89 for a description.

minimum-active = *number*
See "Format of a Transform Entry" on page 89 for a description.

maximum-active = *number*
See "Format of a Transform Entry" on page 89 for a description.

Example

This example shows two sample entries in the **aopxfd.conf** configuration file.

- 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 do 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 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.

```
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
;
```

Customizing PostScript and PDF to AFP 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 a transform resource file, 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. Identify 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 14 turns font substitution on.

```
%!
turnFontSubstitutionOn
```

Figure 14. 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.

Customizing the SAP to AFP Transform

To use the SAP to AFP transform, perform the following customization tasks:

Task	Condition	See Page:
Setting Environment Variables for the SAP to AFP Transform	Optional: To change default location of files	103
Installing Multibyte Conversion Tables for the SAP to AFP Transform	Optional: To transform DBCS data	103
Customizing SAP to AFP Configuration Files	Optional: To change default files	104

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 (1) on the **sap2afp** command and (2) in the **filter-options** job attribute when submitting a print request.

Refer to the following publications for more information:

- *z/OS Infoprint Server Operation and Administration* for information about how to specify the transform DLL and transform options in a printer definition.
- *z/OS Infoprint Server User's Guide* for information about the **sap2afp** command and the transform options.

Setting Environment Variables for the SAP to AFP Transform

The following environment variable affects the behavior of the SAP to AFP transform. If you need to set this environment variable, set it both in the **aopstart** EXEC (see "Editing the **aopstart** EXEC" on page 36 for information) and in the **/etc/profile** file.

AOP_SAP2AFP_RESOURCES
: Path where resources used by the SAP to AFP transform are located. If you installed Infoprint Server Transforms resources in the default directory, **/usr/lpp/Printsrv/sap2afp**, you do not need to set this environment variable.

  Default: /usr/lpp/Printsrv/sap2afp

Installing Multibyte Conversion Tables for the SAP to AFP Transform

In order to transform and print SAP R/3 data streams that contain multibyte special characters, such as data streams produced by Asian versions of the SAP R/3 system, you must install binary conversion tables for the code pages used by the SAP to AFP transform in the **/usr/lib/nls/locale/uconvTable** directory.

The z/OS UNIX **uconvdef** command can create the required conversion tables. Infoprint Server Transforms provides the following source files in the **/usr/lpp/Printsrv/sap2afp** directory, which you should use as input to the **uconvdef** command . These source files define the mapping between UCS-2 and multibyte code sets:

IBM-932-SAP2AFP.ucmap
: Defines the mapping for the SAP R/3 8000 (Japanese) system code page. The **sap2afp** transform uses code page IBM-932-SAP2AFP as the **iconv** source code page.

IBM-1030-SAP2AFP.ucmap
: Defines the mapping for the IBM-1030-SAP2AFP code page. This mapping is based on IBM-1027.ucmap, with the addition of the 11 box drawing characters. To use the IBM-1030-SAP2AFP code page, customize the **80000000.tab** configuration file.

IBM-1031-SAP2AFP.ucmap
: Defines the mapping for the IBM-1031-SAP2AFP code page, which the **sap2afp** transform uses as the default output code page. This mapping is based on IBM-290.ucmap, with the addition of the 11 box drawing characters. You can change the output code page in the **80000000.tab** configuration file, described in "xxxx0000.tab Configuration File" on page 114.

To create binary tables in the **/usr/lib/nls/locale/uconvTable** directory, use the following z/OS UNIX commands:

1. Switch to an effective UID of 0:

   ```
   su root
   ```

   To use the **su** command, you must be permitted to the BPX.SUPERUSER FACILITY class profile within RACF.
2. Change to the **/usr/lib/nls/locale/uconvTable** directory:

   ```
   cd /usr/lib/nls/locale/uconvTable
   ```
3. Create binary conversion tables:

```
uconvdef -f /usr/lpp/Printsrv/sap2afp/IBM-932-SAP2AFP.ucmap -v IBM-932-SAP2AFP
uconvdef -f /usr/lpp/Printsrv/sap2afp/IBM-1030-SAP2AFP.ucmap -v IBM-1030-SAP2AFP
uconvdef -f /usr/lpp/Printsrv/sap2afp/IBM-1031-SAP2AFP.ucmap -v IBM-1031-SAP2AFP
```

4. Set the **p** extended attribute for the binary conversion tables:

   ```
   extattr +p IBM-*-SAP2AFP
   ```

5. Change the access permissions:

   ```
   chmod 644 IBM-*-SAP2AFP
   ```

For more information about these commands, refer to *z/OS UNIX System Services Command Reference*.

Customizing SAP to AFP Configuration Files

The SAP to AFP transform, **sap2afp**, uses the following configuration files:

- **barcode.tab**
- **defcp.tab**
- **fonts.tab**
- **image.tab**
- **pagedef.tab**
- **printer.tab**
- ***xxxx*0000.tab**

Infoprint Server Transforms provides default configuration files in the **/usr/lpp/Printsrv/sap2afp** directory. If you choose to put the SAP to AFP configuration files in a different directory, then identify the new directory in environment variable AOP_SAP2AFP_RESOURCES. All of the configuration files must be in the same directory.

To customize a configuration file, do the following:

1. Create a new directory.

   IBM recommends that you do *not* change files that reside in the **/usr/lpp/Printsrv** directory. Therefore, create a new directory, for example, you can create directory **/etc/Printsrv/sap2afp**.

2. Copy all files to the new directory or use symbolic links; edit the files.

   All configuration files must in the same directory; however, if you do not want to customize some files, you can use symbolic links to the **/usr/lpp/Printsrv/sap2afp** directory for those files. When editing the files, follow these rules:

   - All tables can contain comments and empty lines. Start comments with **//** in column 1. Comments and empty lines are ignored by the **sap2afp** transform.
   - All table entries are case-sensitive.

3. Identify the new directory that contains the configuration files in the AOP_SAP2AFP_RESOURCES environment variable.

barcode.tab Configuration File

The **barcode.tab** configuration file contains a table that maps bar code names used by SAP to the matching bar codes defined by the AFP Bar Code Object Content Architecture (BCOCA).

You can specify the following keywords and values in file **barcode.tab**:

BarCode = *name*

This required keyword specifies the OTF bar code names (SAPBARCODE parameter of the BC OTF command). The maximum length of this field is 8 bytes. It can contain any value.

Type = *nnn*
This required keyword specifies the AFP bar code type as defined in *IBM Data Stream and Object Architectures: Bar Code Object Content Architecture Reference* for the **Barcode Type** parameter of the **Barcode Data Descriptor** structured field. This value can be any decimal value between 0 and 999. The **sap2afp** transform does not verify that this value maps to a valid hexadecimal barcode type.

Mode=*nnn*
This required keyword specifies the modifier byte within the **Barcode Modifier** parameter of the **Barcode Data Descriptor** structured field. Any numeric value between 0 and 999 is accepted and not verified.

Flag = [0 | 128]
This required keyword controls the printing of the Human Readable Interface (HRI) character. Specify one of the following values:

0 Causes the HRI character to be printed

128 Causes the HRI character not to be printed

ModWidth = *nnn*
This optional keyword specifies the width of the narrowest defined bar code element (bar or space) in mils (that is, thousandths of an inch – 1/1000). Allowed values are 1– 254.

For bar code types that explicitly specify the module width, such as POSTNET and RM4SCC, this field is ignored.

Default: The printer default ModWidth

Ratio = *nnn*
This optional keyword specifies the ratio of the width of wide bar code elements to narrow bar code elements, when only two different size elements exist, that is, for a two-level bar code type. Express the ratio as a percentage. Allowed values are 100 – 500; however, the value should usually be between 200 and 300. For example, 200 represents a ratio of 2 to 1; 250 represents a ratio of 2.5 to 1.

Specify a value that is appropriate for the bar code type and for your printer. Many bar codes do not support ratio values, and other bar codes require that the ratio value fall within a specific range. If you specify a ratio value that is not appropriate for the printer, error messages are issued at printing time.

For bar code types that explicitly specify the module width, such as POSTNET and RM4SCC, this field is ignored.

Default: The default ratio for the bar code type.

Note: Verify any values you enter in this table. The **sap2afp** transform does not verify values, and invalid values might result in errors.

Figure 15 on page 106 shows the default **barcode.tab** configuration file.

```
// Barcode table
// Format : Barcode=ARTNR Type=017 Mode=002
// BarCode = 3OF9     Type = 001 Mode = 001 Flag = 000
// BarCode = 3OF91    Type = 001 Mode = 002 Flag = 000
// BarCode = MSI      Type = 002 Mode = 002 Flag = 000
// BarCode = UPCA     Type = 003 Mode = 000 Flag = 000
// BarCode = UPCE     Type = 005 Mode = 000 Flag = 000
// BarCode = UPC2     Type = 006 Mode = 000 Flag = 000
// BarCode = UPC5     Type = 007 Mode = 000 Flag = 000
// BarCode = EAN8     Type = 008 Mode = 000 Flag = 000
// BarCode = EAN13    Type = 009 Mode = 000 Flag = 000
// BarCode = IND25    Type = 010 Mode = 002 Flag = 000
// BarCode = MAT25    Type = 011 Mode = 002 Flag = 000
// BarCode = INTER    Type = 012 Mode = 002 Flag = 000
// BarCode = CODA     Type = 013 Mode = 002 Flag = 000
// BarCode = C128     Type = 017 Mode = 002 Flag = 000
// BarCode = EAN2     Type = 022 Mode = 000 Flag = 000
// BarCode = EAN5     Type = 023 Mode = 000 Flag = 000
// BarCode = POST     Type = 024 Mode = 002 Flag = 000
// BarCode = AUFNR    Type = 001 Mode = 001 Flag = 128
BarCode = ARTNR    Type = 017 Mode = 002 Flag = 000
BarCode = AUFNR    Type = 017 Mode = 002 Flag = 000
BarCode = BARCLVS  Type = 001 Mode = 001 Flag = 000
BarCode = ARTNR    Type = 017 Mode = 002 Flag = 000
BarCode = AUFNR    Type = 017 Mode = 002 Flag = 000
BarCode = BARCLVS  Type = 001 Mode = 001 Flag = 000
BarCode = BC_C128B Type = 017 Mode = 002 Flag = 128
BarCode = BC_CD39  Type = 001 Mode = 001 Flag = 128
BarCode = BC_CD39C Type = 001 Mode = 002 Flag = 128
BarCode = BC_EAN13 Type = 009 Mode = 000 Flag = 128
BarCode = BC_EAN8  Type = 008 Mode = 000 Flag = 128
BarCode = BC_EANH  Type = 017 Mode = 002 Flag = 128
BarCode = BC_I25   Type = 012 Mode = 001 Flag = 128
BarCode = BC_I25C  Type = 012 Mode = 002 Flag = 128
BarCode = BC_MSI   Type = 002 Mode = 001 Flag = 128
BarCode = BC_MSIC  Type = 002 Mode = 002 Flag = 128
BarCode = BC_MSIC1 Type = 002 Mode = 003 Flag = 128
BarCode = BC_MSIC2 Type = 002 Mode = 005 Flag = 128
BarCode = BC_PSN5  Type = 024 Mode = 000 Flag = 128
BarCode = BC_PSN9  Type = 024 Mode = 001 Flag = 128
BarCode = KUNAUNR  Type = 017 Mode = 002 Flag = 000
BarCode = KUNAUPS  Type = 017 Mode = 002 Flag = 000
BarCode = MBBARC   Type = 017 Mode = 002 Flag = 000
BarCode = MBBARC1  Type = 008 Mode = 000 Flag = 000
BarCode = RSNUM    Type = 017 Mode = 002 Flag = 000
BarCode = RSPOS    Type = 017 Mode = 002 Flag = 000
BarCode = RUECKNR  Type = 017 Mode = 002 Flag = 000
```

Figure 15. Default **barcode.tab** *Configuration File*

defcp.tab Configuration File

The **defcp.tab** configuration file contains a table that maps the Open Systems EBCDIC 1047 code page to the code page associated with the ABAP coded fonts specified in the **pagedef.tab** file.

Figure 16 on page 108 shows the default **defcp.tab** file. The table in this file maps to output code page T1DBBASE for use with coded fonts GT2B, GT5B, GT8B, and GB2B, which are specified in the default **pagedef.tab** file.

Table 7 on page 107 shows how the **sap2afp** transform maps ABAP box characters (called frame characters in SAP documentation) to the corresponding code point in

the output code page. The transform maps the ABAP characters to hex values X'00' through X'0A'. The table in file **defcp.tab** then maps these hex values to a code point. For example, \G0 and #462 are ABAP characters for the upper left box corner (called ″down right″ in SAP documentation). The transform maps these characters to X'00'. The table in file **defcp.tab** maps X'00' to **ac**, which is the code point for an upper left box corner in code page T1DBBASE.

Table 7. How SAP ABAP Box Characters Map to Code Points

ABAP Frame Characters	GOF ABAP Frame Characters	Description in SAP Documentation	Description in defcp.tab	Mapped Hex Value	Code Point (See Note)
\G0	#462	down right	upper left box corner	00	ac
\G1	#464	up right	lower left box corner	01	ad
\G2	#463	down left	upper right box corner	02	ae
\G2	#456	up left	lower right box corner	03	b6
\G4	#460	horizontal	center box bar hoizontal	04	be
\G5	#461	vertical	center box bar vertical	05	df
\G6	#466	vertical right	left middle box side	06	bc
\G7	#467	vertical left	right middle box side	07	bd
\G8	#469	up horizontal	middle box bottom	08	b9
\G9	#468	down horizontal	middle box top	09	b8
\G:	#470	vertical horizontal	box intersection	0A	b7

Note: The code points shown in this column assume the T1DBBASE output code page. You can change the code points in the **defcp.tab** file.

```
// ABAP/Line Data Code Page
//
// Maps codepoints from codepage "Open Systems 1047" + box characters (0 - a)
//                    to codepage "T1DBBASE (Belgium/Swiss/Luxembourg-DBBase)"
//
// NOTE:  Codepage TIDBBASE chosen because it has box characters.
//
// ABAP box characters mapped to 0-a by sap2afp transform:
//   0 = upper left box corner
//   1 = lower left box corner
//   2 = upper right box corner
//   3 = lower right box corner
//   4 = center box bar horizontal
//   5 = center box bar vertical
//   6 = left middle box side
//   7 = right middle box side
//   8 = middle box bottom
//   9 = middle box top
//   a = box intersection
// Format : Ebcdic=Ebcdic
0 = ac
1 = ad
2 = ae
:
fd = fd
fe = fe
ff = 40
```

Figure 16. Default **defcp.tab** *Configuration File*

Asian versions of SAP R/3 use special coding to represent double-byte special characters; therefore, the **defcp.tab** file must specify the SAP R/3 system code page and the output code page conversion table. The following attributes in the table specify the input and output code pages:

CONVERT FOR=*xxxx*
: Specifies the R/3 system code page, where *xxxx* is one of the following values:
  8000 Japanese
  8300 Chinese – traditional/Taiwan
  8400 Chinese – simplified/PR China
  8500 Korean

CONVERT TO=*yyyyyyyyyy*
: Specifies the output code page conversion table, for example, IBM-1031-SAP2AFP.

Figure 17 on page 109 shows the Japanese conversion table in file **defcp.tab.japan**. To use this file, rename it to **defcp.tab**.

```
// defcp.tab
//-----------------------------------------------------------------------
// This file is used internally by sap2afp, as well as to convert ABAP
// data to linedata.
// SAP2AFP is configured to use IBM-932 as the iconv "From" value.  This table
// sets the "To" value to IBM-1031.
CONVERT FOR=8000
CONVERT TO=IBM-1031-SAP2AFP
```

Figure 17. Sample **defcp.tab.japan** *Configuration File*

fonts.tab Configuration File

The **fonts.tab** configuration file contains a table that maps the fonts used in the SAP OTF data stream to AFP fonts. The following font families are predefined with SAP R/3:

Font Family	Font
COURIER	Courier
HELVE	Helvetica
LETGOTH	Letter Gothic
LNPRINT	Line Print
TIMES	Times New Roman
OCRA	Optical Character Recognition A
OCRB	Optical Character Recognition B
JPMINCHO	Heisei Mincho
DBMINCHO	Heisei Mincho
DBGOTHIC	Heisei Gothic

Note: JPMINCHO and DBMINCHO are different names for the same font.

A font named **BARCODE** must be defined for the **HRI** character of a bar code.

The following attributes in the **fonts.tab** configuration file set the format of the fonts you use to print with R/3:

DefCodePage Specifies the default SBCS code page used if no **FC** OTF command is given or if the requested font is not found in the **fonts.tab** table.

DefCharSet Specifies the default SBCS character set used if no **FC** OTF command is given or if the requested font is not found in the **fonts.tab** table.

DBDefCodePage
Specifies the default DBCS code page used if no **FC** OTF command is given or if the requested font is not found in the **fonts.tab** table.

DBDefCharSet
Specifies the default DBCS character set used if no **FC** OTF command is given or if the requested font is not found in the **fonts.tab** table.

SBDefCodePage
Specifies the default SBCS code page used for half-width characters in DBCS fonts if no **FC** OTF command is given or if the requested font is not found in the **fonts.tab** table.

SBDefCharSet
Specifies the default SBCS character set used for half-width characters in DBCS fonts if no **FC** OTF command is given or if the requested font is not found in the **fonts.tab** table.

Font
Describes the font family (**FONTFAMILY** parameter of the **FC** OTF command). The maximum size is 8 bytes and content is not verified.

Size
Specifies the font size in 1/10 of a point (**FONT SIZE** parameter of the **FC** OTF command). The value must be numeric and is not verified.

Type
Defines the font type (**BOLD** and **ITALIC** parameter of the **FC** OTF command). **Type=0** is normal, **Type=1** is italic, **Type=2** is bold, and **Type=3** is italic bold. Any other value is invalid.

CodePage
Specifies the code page and requires a valid AFP code page name (8 bytes). The value is not verified. An invalid name can result in an error message.

CharSet
Specifies the AFP font character set. The content is not verified.

CodedFont
Specifies the AFP coded font. The content is not verified.

DB
Specifies whether the font is an SBCS font (**DB=0**) or a DBCS font (**DB=1**).

SBCodePage
Specifies the single-byte code page used for half-width characters in DBCS fonts. The value is not verified. An invalid name can result in an error message.

SBCharSet
Specifies the AFP single-byte font character set used for half-width characters in DBCS fonts. The content is not verified.

SBCodedFont
Specifies the AFP single-byte coded font used for half-width characters in DBCS fonts. The content is not verified.

If an SBCS font that matches the **Font**, **Size**, and **Type** values is not found, the code page and font character set from the **DefCodePage** and **DefCharSet** attributes are used and a warning message is displayed. If a DBCS font that matches the **Font**, **Size**, and **Type** values is not found, the code pages and font character sets from the **DBDefCodePage**, **SBDefCodePage**, **DBDefCharSet**, and **SBDefCharSet** attributes are used and a warning message is displayed.

The SAP device types **IBMAFP** and **IBMAFP3** support the predefined font families. These font families are also supported as IBM Expanded Core Fonts, or IBM Chinese, Japanese, and Korean (CJK) Fonts.

For ABAP listings, use the Letter Gothic Latin1 font provided with the AFP Font Collection V2 (program number 5648-B33). If you choose not to install the AFP Font Collection and plan to use the Gothic Text fonts provided with the IBM Compatibility Fonts, adjust the font names in the **pagedef.tab** configuration file.

Figure 18 on page 111 and Figure 19 on page 111 show sections of the **fonts.tab** configuration file.

```
// Fonts table
// Format : Font=COURIER Size=070 Type=0 CodePage=T1V10500 CharSet=C0420070
//          DefCodePage = T1V10500
//          DefCharSet  = C0420000
DefCodePage = T1V10500
DefCharSet  = C0420000
DBDefCodePage = T10300
DBDefCharSet  = CZJHMN
SBDefCodePage = T1H01027
SBDefCharSet  = CZJHMN
// Courier
Font=COURIER  Size=070 Type=0 CodePage=T1V10500 CharSet=C0420070 DB=0
Font=COURIER  Size=070 Type=1 CodePage=T1V10500 CharSet=C0430070 DB=0
Font=COURIER  Size=070 Type=2 CodePage=T1V10500 CharSet=C0440070 DB=0
Font=COURIER  Size=070 Type=3 CodePage=T1V10500 CharSet=C0450070 DB=0
Font=COURIER  Size=080 Type=0 CodePage=T1V10500 CharSet=C0420080 DB=0
Font=COURIER  Size=080 Type=1 CodePage=T1V10500 CharSet=C0430080 DB=0
Font=COURIER  Size=080 Type=2 CodePage=T1V10500 CharSet=C0440080 DB=0
Font=COURIER  Size=080 Type=3 CodePage=T1V10500 CharSet=C0450080 DB=0
Font=COURIER  Size=100 Type=0 CodePage=T1V10500 CharSet=C0420000 DB=0
Font=COURIER  Size=100 Type=1 CodePage=T1V10500 CharSet=C0430000 DB=0
Font=COURIER  Size=100 Type=2 CodePage=T1V10500 CharSet=C0440000 DB=0
Font=COURIER  Size=100 Type=3 CodePage=T1V10500 CharSet=C0450000 DB=0
Font=COURIER  Size=120 Type=0 CodePage=T1V10500 CharSet=C04200B0 DB=0
Font=COURIER  Size=120 Type=1 CodePage=T1V10500 CharSet=C04300B0 DB=0
Font=COURIER  Size=120 Type=2 CodePage=T1V10500 CharSet=C04400B0 DB=0
Font=COURIER  Size=120 Type=3 CodePage=T1V10500 CharSet=C04500B0 DB=0
Font=COURIER  Size=140 Type=0 CodePage=T1V10500 CharSet=C04200D0 DB=0
Font=COURIER  Size=140 Type=1 CodePage=T1V10500 CharSet=C04300D0 DB=0
Font=COURIER  Size=140 Type=2 CodePage=T1V10500 CharSet=C04400D0 DB=0
Font=COURIER  Size=140 Type=3 CodePage=T1V10500 CharSet=C04500D0 DB=0
Font=COURIER  Size=200 Type=0 CodePage=T1V10500 CharSet=C04200J0 DB=0
Font=COURIER  Size=200 Type=1 CodePage=T1V10500 CharSet=C04300J0 DB=0
Font=COURIER  Size=200 Type=2 CodePage=T1V10500 CharSet=C04400J0 DB=0
Font=COURIER  Size=200 Type=3 CodePage=T1V10500 CharSet=C04500J0 DB=0
```

Figure 18. Header and Courier Portions of Default **fonts.tab** *Configuration File*

```
// Japanese Gothic
Font=DBGOTHIC Size=060 Type=0 CodePage=T10300 CharSet=CZJHKG DB=1 SBCodePage=T1H01027 SBCharSet=CZJHKG
Font=DBGOTHIC Size=080 Type=0 CodePage=T10300 CharSet=CZJHKG DB=1 SBCodePage=T1H01027 SBCharSet=CZJHKG
Font=DBGOTHIC Size=100 Type=0 CodePage=T10300 CharSet=CZJHKG DB=1 SBCodePage=T1H01027 SBCharSet=CZJHKG
Font=DBGOTHIC Size=120 Type=0 CodePage=T10300 CharSet=CZJHKG DB=1 SBCodePage=T1H01027 SBCharSet=CZJHKG
Font=DBGOTHIC Size=140 Type=0 CodePage=T10300 CharSet=CZJHKG DB=1 SBCodePage=T1H01027 SBCharSet=CZJHKG
Font=DBGOTHIC Size=160 Type=0 CodePage=T10300 CharSet=CZJHKG DB=1 SBCodePage=T1H01027 SBCharSet=CZJHKG
Font=DBGOTHIC Size=180 Type=0 CodePage=T10300 CharSet=CZJHKG DB=1 SBCodePage=T1H01027 SBCharSet=CZJHKG
```

Figure 19. Heisei Gothic Portion of Default **fonts.tab** *Configuration File*

image.tab Configuration File

The **image.tab** configuration file contains a table that defines values used to print image data. It contains the following parameters:

DEFRES Specifies the default resolution used for printing image data if you do not specify a resolution in the printer definition and the job submitter does not specify a resolution in the **-r** option on **lp** or **sap2afp** command. Valid values are **240**, **300**, **480**, and **600**.

Width Specifies the width of the dither matrix.

Height Specifies the height of the dither matrix.

Cell Specifies the values for the dither matrix.

Transform Specifies 256 grayscale correction values.

Infoprint Server Transforms provides the following **image.tab** files. To use one of these files, rename it to **image.tab**:

- **image.tab** - same as **image.tab.85lr**
- **image.tab.141ap** - 141 lines/inch AppleWriter grayscale emulation
- **image.tab.141dt** - 141 lines/inch Xerox DocuTech grayscale emulation
- **image.tab.141lr** - 141 lines/inch linear gamma correction
- **image.tab.85ap** - 85 lines/inch AppleWriter grayscale emulation
- **image.tab.85dt** - 85 lines/inch Xerox DocuTech grayscale emulation
- **image.tab.85lr** - 85 lines/inch linear gamma correction

Figure 20 shows the default **image.tab** configuration file.

```
// image.tab
//
// This file is used to determine the dither matrix and the
// grayscale correction values.
// Define the default output resolution
DEFRES = 600
// Define the dither matrix
Width = 78
Height = 78
Cell =
 15  63 184 219 249 240 158 109  98  54  82 118 133 113  75  64
 87 104 166 215 234 229 179  62  38   4  15  59 186 220 246 240
166 123  87  61  65 108 133 118  78  64 100 105 167 219 234 230
   ⋮
 28  73 193 223 253 247 199  76  31   8  20  93 148 156 203 178
136 126 144 186 201 155 151  90  43  24  28  75 199 222 254 251
196  79  31   7
//
// Grayscale correction values
//
Transform =
0
1
2
⋮
248
252
255
```

Figure 20. Default **image.tab** *Configuration File*

pagedef.tab Configuration File

The **pagedef.tab** configuration file contains a table that maps SAP R/3 Format names to the names of AFP form definitions, AFP page definitions, and fonts. The **sap2afp** transform uses the form definition name when it transforms either SAP OTF or ABAP data; however, the **sap2afp** transform uses the page definition and font names only for SAP ABAP data.

Every SAP R/3 document to be printed is associated with an SAP R/3 Format, for example the X_65_80 Format for ABAP data or the LETTER format for OTF data. You can also define user Formats to SAP R/3. If you define a user Format, you must add it to the **pagedef.tab** file.

A form definition lets you specify electronic overlays, portrait or landscape presentation mode, the paper source, simplex or duplex printing, and so on. By default, the **sap2afp** transform uses the F1A10111 form definition, which specifies no electronic overlays, portrait presentation, the primary paper source, and duplex printing. You can change the form definition in the **pagedef.tab** file to another form definition that PSF for OS/390 provides. Refer to *PSF for OS/390 & z/OS: User's Guide* for information about form definitions that PSF for OS/390 provides.

For example, to print in the landscape orientation you could define new user Formats to SAP R/3 (for example ZLAND for ABAP data and Z_65_255 for OTF data), and then map the new user Formats to form definition F1C10110 in the **pagedef.tab** file. F1C10110 is a form definition with landscape presentation mode specified provided by PSF for OS/390.

File **pagedef.tab** contains the following parameters:

Paper Specifies the name of the SAP R/3 format. This is the value of the OTF print option parameter **PJPAPER**. This value is not verified.

FormDef Specifies the name of the form definition to be used for printing both OTF and ABAP reports. This value is not verified; however, an error results if PSF cannot find the form definition during printing.

PageDef Specifies the name of the page definition to be used for printing ABAP reports. This value is not verified; however, an error results if PSF cannot find the page definition during printing.

FontNorm Specifies the normal coded font used for printing ABAP data. The value is not verified.

FontBold Specifies the bold coded font used for printing ABAP data. The value is not verified.

Figure 21 on page 114 shows the default **pagedef.tab** configuration file.

```
// PageDef table
//-----------------------------------------------------------------------------
// This table allows users to configure resource selection based on a
// Format in the SAP system.  Thus a format, say ZCHECK, could be set-up
// to use a Formdef that would invoke an Overlay for checks.
//
// Although all paper types are defined in the same format, conventions
// on the SAP system should be observed as follows:
//   + ABAP user defined formats should be in the form Z_ROWS_COLUMNS or
//     Z_ROWS_COLUMNS_EXTRA-ID.  An example of a user defined ABAP format
//     would be Z_65_132_MYABAP.
//   + Formats for OTF in the SAP system are formats listed that do not
//     start with a X (i.e. LEGAL).  An example of a user defined OTF format
//     would be ZMYFORMAT.
//   + Do NOT copy ABAP formats as OTF formats or OTF formats as ABAP formats!
//
// Resource selection is done as follows:
//   + For ABAP data, all entries (FormDef, PageDef, FontNorm, FontBold)
//     are used.
//   + For OTF data, only the FormDef value is used.  However, the other
//     field values must be defined.
//-----------------------------------------------------------------------------
// Format: Paper=X_65_132 FormDef=F1A10111 PageDef=P1V06683 FontNorm=GT2B
//         FontBold=GB2B
// Notes:
// ======
// 1) the pagedefs apply to letter size paper.
//    A4 paper size may require modifying the PageDef field.
// 2) Z_INCH12 is a sample for a user specification
// 3) A format (Paper) must not exceed 16 characters.
//-----------------------------------------------------------------------------
// ABAP Formats
Paper=X_65_132 FormDef=F1A10111 PageDef=P1V06683 FontNorm=GT2B FontBold=GB2B
Paper=X_44_120 FormDef=F1A10111 PageDef=P1V06683 FontNorm=GT2B FontBold=GB2B
Paper=X_58_170 FormDef=F1A10111 PageDef=P1V06683 FontNorm=GT8B FontBold=GT8B
Paper=X_65_255 FormDef=F1A10111 PageDef=P1V06683 FontNorm=GT2B FontBold=GT2B
Paper=X_65_80  FormDef=F1A10111 PageDef=P1C09182 FontNorm=GT2B FontBold=GB2B
Paper=X_90_120 FormDef=F1A10111 PageDef=P1C09182 FontNorm=GT5B FontBold=GT5B
Paper=X_PAPER  FormDef=F1A10111 PageDef=P1C09182 FontNorm=GT2B FontBold=GB2B
// OTF Formats
Paper=DINA3    FormDef=F1A10111 PageDef=P1V06683 FontNorm=GT2B FontBold=GB2B
Paper=DINA4    FormDef=F1A10111 PageDef=P1V06683 FontNorm=GT2B FontBold=GB2B
Paper=DINA5    FormDef=F1A10111 PageDef=P1V06683 FontNorm=GT2B FontBold=GB2B
Paper=EXECUTIV FormDef=F1A10111 PageDef=P1V06683 FontNorm=GT2B FontBold=GB2B
Paper=INCH4    FormDef=F1A10111 PageDef=P1V06683 FontNorm=GT2B FontBold=GB2B
Paper=INCH6    FormDef=F1A10111 PageDef=P1V06683 FontNorm=GT2B FontBold=GB2B
Paper=INCH7    FormDef=F1A10111 PageDef=P1V06683 FontNorm=GT2B FontBold=GB2B
Paper=INCH8    FormDef=F1A10111 PageDef=P1V06683 FontNorm=GT2B FontBold=GB2B
Paper=INCH11   FormDef=F1A10111 PageDef=P1V06683 FontNorm=GT2B FontBold=GB2B
Paper=INCH12   FormDef=F1A10111 PageDef=P1V06683 FontNorm=GT2B FontBold=GB2B
Paper=LEGAL    FormDef=F1A10111 PageDef=P1V06683 FontNorm=GT2B FontBold=GB2B
Paper=LETTER   FormDef=F1A10111 PageDef=P1V06683 FontNorm=GT2B FontBold=GB2B
Paper=LINE_21  FormDef=F1A10111 PageDef=P1V06683 FontNorm=GT2B FontBold=GB2B
Paper=LINE_22  FormDef=F1A10111 PageDef=P1V06683 FontNorm=GT2B FontBold=GB2B
//Sample of a added user-defined pagedef entry:
//Paper=Z_INCH12 FormDef=F1A10111    PageDef=P1SAPPD  FontNorm=GT2B FontBold=GB2B
```

Figure 21. Default **pagedef.tab** *Configuration File*

xxxx0000.tab Configuration File

The following configuration files contain tables that map characters of an individual SAP code page into an AFP code page.

Table 8. SAP Code Page Configuration Files

File Name	Input Code Page	Description	Output Code Page	Description
00000000.tab	T1000819	Latin-1 ISO ANSI 8-bit (ISO8859–1)	T1V10500	International #5
01200000.tab	T1000819	Latin-1 ISO ANSI 8-bit (ISO8859–1)	T1V10500	International #5
11000000.tab	T1000819	Latin-1 ISO ANSI 8-bit (ISO8859–1)	T1V10500	International #5
11000000.tab.japan [1]	T1000819	Latin-1 ISO ANSI 8-bit (ISO8859–1)	T1V10281	Japan (Latin)-CECP
40010000.tab	T1000876 [2]	OCR-A ASCII	T1000892	OCR-A
40020000.tab	T1000877 [2]	OCR-B ASCII	T1000893	OCR-B
80000000.tab	IBM-932-SAP2AFP	Japanese	IBM-1031-SAP2AFP	Japanese

1. To produce output in the T1V10281 Japan (Latin) code page, rename file **11000000.tab.japan** to **11000000.tab**.
2. The input OCR-A and the OCR-B code pages are equivalent to ISO-8859-1 with the special characters **hook**, **fork**, and **chair**.

Asian versions of SAP R/3 use special coding to represent double-byte special characters. Therefore, the ***xxxx*0000.tab** file must indicate the R/3 system code page and the output code page. The following parameters in the table specify the input and output code pages:

CONVERT FOR=*xxxx*
: Specifies the R/3 system code page, where *xxxx* is one of the following values:
  8000 Japanese
  8300 Chinese – traditional/Taiwan
  8400 Chinese – simplified/PR China
  8500 Korean

CONVERT TO=*yyyyyyyyyy*
: Specifies the output code page conversion table, for example IBM-1031-SAP2AFP.

Figure 22 shows the default **80000000.tab** configuration file used for the R/3 system code page 8000 (Japanese).

```
// Japanese
CONVERT FOR=8000
CONVERT TO=IBM-1031-SAP2AFP
```

Figure 22. Default **80000000.tab** *Configuration File*

Customizing the AFP to PCL, AFP to PDF, and AFP to PostScript Transforms

To use the AFP to PCL, AFP to PDF, or AFP to PostScript transform, perform the following customization tasks:

Task	Condition	See Page:
Customizing JES to Specify No Blank Truncation	Required	53

Task	Condition	See Page:
Editing the Transform Configuration File (**aopxfd.conf**)	Required	116
Customizing the Font-Mapping Table	Optional: To customize default font mapping	133
Scaling 240-Pel to 300-Pel Fonts using the AOXCF30 Program	Optional: If you have only 240-pel raster fonts and the transform requires 300-pel raster fonts.	135

Editing 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 "Creating the Transform Configuration File (**aopxfd.conf**)" on page 87 for more information about this file.

If this configuration file does not already exist, create it before you start the Transform Manager. If you change the configuration file while the Transform Manager is started, stop and restart the Transform Manager to pick up the changes.

Specifying Transform Classes

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, or AFP resource libraries. For example, you can define one transform class for color printers, and another for monochrome printers. 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/aopxfd.conf**, shows how to define different transform classes. Also see "Transform Classes" on page 88 for more information about how to use transform classes.

Specifying 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 the paper trays installed in your printers. 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 the installed paper trays. You can specify from one to ten positional tray IDs. The position (one through ten) 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 1 1 2"
```

If the user specifies 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 9 summarizes the valid paper names and their dimensions.

You can specify from one to ten positional paper names. The position (one through ten) 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 the user selects AFP input tray 2, then the transform formats the output for legal size paper. If the user selects input tray 3, then the transform formats the output for `letteree` size paper. If the user selects 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.

The user can specify an AFP tray ID in the **input-tray-number** job attribute, the INTRAY JCL parameter, or in the AFP form definition. The user 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.

Table 9 summarizes the names and dimensions of the paper that the transforms support.

Table 9. Paper Names and Dimensions

Paper Name	Paper Size	Printable Area Origin	Printable Area Size	Logical Page Origin	Logical Page Size (PCL only)	PCL Paper Type ID
a3	3507,4960	50,50	3407,4860	71,0	3365,4960	27
a4	2480,3507	50,50	2380,3407	71,0	2338,3507	26
a4ee	2480,3507	0,0	2480,3507	0,0	2480,3507	26
a5	1748,2480	50,50	1648,2380	71,0	1606,2480	25
c5	1913,2704	50,50	1813,2604	71,0	1771,2704	91
com10	1237,2850	50,50	1137,2750	75,0	1087,2850	81
dl	1299,2598	50,50	1199,2498	71,0	1157,2598	90
executive	2175,3150	50,50	2075,3050	75,0	2025,3150	1
legal	2550,4200	50,50	2450,4100	75,0	2400,4200	3
letter	2550,3300	50,50	2450,3200	75,0	2400,3300	2

Table 9. Paper Names and Dimensions (continued)

Paper Name	Paper Size	Printable Area Origin	Printable Area Size	Logical Page Origin	Logical Page Size (PCL only)	PCL Paper Type ID
letteree	2550,3300	0,0	2550,3300	0,0	2550,3300	2
monarch	1162,2250	50,50	1062,2150	75,0	1012,2250	80

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. See the description of the AOP_PAPER variable in "AFP to PCL, AFP to PostScript, or AFP to PDF Transform Entry" on page 123 for a list of the paper sizes in inches and millimeters.
- **Printable Area Origin**: The X and Y coordinates of the top left corner of the printable area in relation to the physical medium.
- **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 medium.
- **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.

Note: Use the **a4ee** and **letteree** paper names when the printer is configured for edge-to-edge printing.

If the user specifies 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. The user 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.

Specifying 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.

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.

Recommendations:

- For the AFP to PDF and the AFP to PostScript transforms, specify an outline font because outline fonts provide higher quality output for printing and viewing.
- For the AFP to PCL transform, specify a raster font because this transform requires raster fonts.

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.

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

The transforms use search hierarchies to select resources when they are specified in more than one location. These hierarchies are described in the following sections.

Hierarchy for Selecting a Form Definition: 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.

Notes:

1. If form definition name **dummy** is specified, the transform uses the first inline form definition.
2. 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.

Hierarchy for Selecting a 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.

Notes:

1. If page definition name **dummy** is specified, the transform uses the first inline page definition.
2. 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.

Note: JES can supply a default page definition for jobs submitted directly to IP PrintWay with JCL statements 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 JES2 PRT*nnnnn* statement for the IP PrintWay FSA, and do *not* specify the NIFCB parameter in the JES2 PRINTDEF statement.
- In JES3, specify `PDEFAULT=FCB` in the JES3 DEVICE statement for the IP PrintWay FSA.

See “Defining the IP PrintWay Functional Subsystem” on page 169 for more information.

Hierarchy for Selecting a Font: 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.

Note: JES can supply a default font for jobs submitted directly to IP PrintWay with JCL statements 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 JES2 PRT*nnnnn* statement.
- In JES3, specify `PDEFAULT=CHARS` in the JES3 DEVICE statement for the IP PrintWay FSA.

See “Defining the IP PrintWay Functional Subsystem” on page 169 for more information.

Hierarchy for Selecting a Font for Transform Error Messages: 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.

Specifying 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 as 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.

You can specify system resource libraries in the following environment variables:

AOP_FONTLIB
This variable identifies the system font libraries.

AOP_FORMDEFLIB
This variable identifies the system form definition libraries.

AOP_OVERLAYLIB
This variable identifies the system overlay libraries.

AOP_PAGEDEFLIB
This variable identifies the system page definition libraries.

AOP_PAGESEGLIB
This variable 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

Note: 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 AOPOPER group.) See “Customizing Startup and Shutdown Commands and Procedures” on page 35 for information about the AOPSTART procedure provided in SYS1.PROCLIB.

The transforms use hierarchies to search AFP resource libraries, as described in the following section.

Hierarchy for Searching AFP 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

The 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.

To map fonts, these transforms use an internal font-mapping table. This table supports the IBM AFP Font Collection Expanded Core Fonts, which are available in both raster and outline formats. This font table can be customized for your installation; see “Customizing the Font-Mapping Table” on page 133 for information.

Each transform performs different font mapping, as follows:

- **AFP to PCL Transform**: Maps outline fonts to raster fonts

  This transform can map outline fonts to equivalent 300-pel raster fonts. The AFP to PCL transform can use *only* 300-pel raster fonts. 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.

  Recommendation: Map outline fonts to raster fonts because this transform requires raster fonts.
- **AFP to PostScript Transform**: Maps AFP raster to AFP outline

  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. 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.

  Recommendation: Map raster fonts to outline fonts because outline fonts provide superior print quality at higher printer resolutions.
- **AFP to PDF Transform**: Maps AFP raster to AFP outline

  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. 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.

  Recommendation: Map raster fonts to outline fonts because outline fonts provide superior viewing quality.

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. Consider the following when selecting this value:

- **AFP to PCL transform**: We recommend that you enable font mapping to map outline fonts to raster fonts because this transform requires raster fonts.
- **AFP to PostScript transform**: We recommend 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.
- **AFP to PDF transform**: We recommend 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.

Note: 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.

AOP_OUTLINES (for AFP to PDF transform only)

This variable controls 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.

See the description of each of these variables for more information.

AFP to PCL, AFP to PostScript, or AFP to PDF Transform Entry

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

If you want to define transform classes, create a separate transform entry for each transform class.

transform afp2*ccc*[_*transformclass*]
  start-command = afp2*ccc***d**
  [**environment = {***name* **–>** *value* [*name* **–>** *value*]... **}**]
  [**maximum-idle-time =** *seconds*]
  [**minimum-active =** *number*]
  [**maximum-active =** *number*]
;

transform afp2*ccc***[**_*transformclass***]**

Name of the transform and, optionally, the name of the transform class. **transform** indicates the beginning of a transform entry.

afp2*ccc*

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

transformclass
The name of an optional transform class. Specify from 1 to 63 characters, including letters, numbers, or special characters.

Default: No transform class is defined.

Example: `transform afp2pcl_us`

To use this transform class on the **afp2pcl** command, type:

```
afp2pcl -c us
```

start-command = afp2*ccc*d
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. This attribute is required. Valid values are:

afp2pcld The AFP to PCL transform daemon

afp2pdfd The AFP to PDF transform daemon

afp2psd The AFP to PostScript transform daemon

environment = {*name* –> *value* [*name* –> *value*]... }
Environment variables that define the transform environment for the transform. Enclose the environment variables in braces. The values in these environment variables override environment variables with the same name that were set when the **aopstart** command was issued. Enclose the value in single or double quotation marks if the value contains special characters or spaces.

The following environment variables are optional:

Name **Value**

AOP_ANNOTATIONS
Indicates whether the AFP to PDF transform is to produce annotations in the PDF file. The AFP to PDF transform can create an annotation on each page. The annotation contains printing instructions such as ″Normal duplex Front side″. For more information about annotations, see the online help provided with Adobe Acrobat reader. This variable applies only to the AFP to PDF transform; other transforms ignore this variable. Valid values are:

yes The AFP to PDF transform produces annotations.

no The AFP to PDF transform does not produce annotations.

Default: AOP_ANNOTATIONS –> yes

Example: `environment = {AOP_ANNOTATIONS –> no}`

AOP_CHARS The default coded font. 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 one to four 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 six-character names, not counting the X0 or XZ prefix. For these fonts, use the four-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.

Recommendations:

- You might want to specify the same font as the resident font in the AFP printer to which output is usually printed.
- For the AFP to PDF and the AFP to PostScript transforms, specify an outline font because outline fonts provide higher quality output for printing and viewing.
- For the AFP to PCL transform, specify a raster font because this transform requires raster fonts.

Default: AOP_CHARS –> 60d9

Note: The transforms prefix X0 to this font.

Example: `environment = {AOP_CHARS –> 60d8}`

AOP_COLOR Indicates whether the transform is to produce color output. Specify **yes** if the printer supports color. This variable applies only to the AFP to PCL and AFP to PostScript transforms. The AFP to PDF transform ignores this variable and always produces output in color. Valid values are:

yes The transform produces color output.

no The transform performs color simulation.

Default: AOP_COLOR –> no

Example: `environment = {AOP_COLOR –> yes}`

AOP_CUTSHEET

Indicates whether the transform is to prepare the output for printing on a cutsheet printer. Valid values are:

yes The output is to be printed on a cutsheet printer. Therefore, the transform uses the cutsheet specification in the form definition to determine whether or not to send medium orientation

information to the printer. Refer to the description of the CUTSHEET command in *IBM Page Printer Formatting Aid: User's Guide* for more information about the cutsheet specification in the form definition.

Tip: Select this option if your output is incorrectly printing in the down direction on a cutsheet printer.

no The output is not to be printed on a cutsheet printer. Therefore, the transform always sends medium orientation information to the printer.

Default: AOP_CUTSHEET –> no

Example: `environment = {AOP_CUTSHEET –> no}`

AOP_FLATE Indicates whether the AFP to PDF transform is to use the Adobe Flate compression algorithm to compress the output. 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:

yes The transform produces compressed output.

no The transform does not produce compressed output; however throughput is higher.

Default: AOP_FLATE –> yes

Example: `environment = {AOP_FLATE –> 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.
  - Specify AFP outline font libraries if data to be transformed refers to outline fonts.
- **AFP to PDF transform:**
  - Specify AFP 300-pel raster font libraries if *either* (1) you specify a raster font in the AOP_CHARS environment variable *or* (2) 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* (1) data to be transformed references outline fonts *or* (2) you specify AOP_FONTMAP –> yes to map raster to outline fonts.

- **AFP to PostScript transform:**
  – Specify AFP 300-pel raster font libraries if *either* (1) you specify a raster font in the AOP_CHARS environment variable *or* (2) 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* (1) data to be transformed references outline fonts *or* (2) you specify AOP_FONTMAP –> yes to map raster to outline fonts.

Default: AOP_FONTLIB –> ″sys1.font300 sys1.fontoln″

Examples:

```
environment={AOP_FONTLIB –>
    "sys1.font300 sys1.fontoln inst.font300"}
environment={AOP_FONTLIB –> "sys1.font300"}
```

AOP_FONTMAP

Indicates 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.

Recommendation: Enable font mapping for all transforms. See “Mapping AFP Raster and AFP Outline Fonts” on page 122 for more information.

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_PAGEDEFLIB 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_PAGEDEF

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_PAGEDEF –> p1p08682

Example: `environment = {AOP_PAGEDEF –> p1q09182}`

AOP_PAGEDEFLIB

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_PAGEDEFLIB –> ″sys1.pdeflib″

Example: `environment = {AOP_PAGEDEFLIB –> "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. Valid paper names and their sizes are:

Paper Name	Size
a3	292.25 x 413.25 mm
a4	210 x 297 mm
a4ee	210 x 297 mm
a5	148.3 x 210 mm
c5	162 x 229 mm (6.48 x 9.16 in.)
com10	4.125 x 9.5 in. (104.8 x 241.3 mm)
dl	110 x 220 mm (4.4 x 8.8 in)
executive	7.25 x 10.5 in (185 x 267 mm)
legal	8.5 x 14.0 in (216 x 356 mm)
letter	8.5 x 11.0 in (216 x 279 mm)
letteree	8.5 x 11.0 in (216 x 279 mm)
monarch	3.875 x 7.5 in. (98.4 x 190.5 mm)

Note: Use the **a4ee** and **letteree** paper names when the printer is configured to print edge-to-edge.

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 letter"}`

AOP_PJL Indicates whether the printer accepts all PCL 5 commands, including PJL commands. All printers that support PCL 5 support PJL commands. Some printers that support only PCL 4 do not support PJL commands. Sometimes, a printer that does not support PJL commands prints a smiley face where a PJL command occurs in the data stream or prints the PJL command on the first page.

This environment variable applies only to the AFP to PCL transform; the AFP to PostScript and AFP to PDF transforms ignore this variable. Valid values are:

yes The printer accepts PJL commands.

no The printer does not accept PJL commands; therefore, the AFP to PCL transform does not generate PJL commands.

Default: AOP_PJL –> yes

Example: `environment = {AOP_PJL –> no}`

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 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 0 2″

Example: `environment = {AOP_TRAYID –> "1 4 1 1 1 1 1 1 1 2"}`

Example

This example shows two sample entries in the **aopxfd.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 daemons active at any time. The Transform Manager shuts down any transform daemon that is idle and frees system resources after 5 minutes. The Transform Manager keeps at least one transform daemon of the first type active.

```
transform afp2pcl
   start-command = afp2pcld
   min-active = 1
   max-active = 2
   maximum-idle-time = 300 # 5 minutes
   environment = {
      AOP_FONTLIB -> "sys1.font300"
      AOP_PAGEDEFLIB -> "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 1 2"
      AOP_PAPER ->
         "letter legal letteree letter letter letter letter letter letter letter"
      AOP_COLOR -> no
      AOP_CUTSHEET -> yes
      AOP_PJL -> yes
      AOP_MSGPAGEDEF -> p1p08682
      AOP_MSGFORMDEF -> f1cp0110
      AOP_PAGEDEF -> p1p08682
      AOP_FORMDEF -> f1cp0110
      AOP_CHARS -> 60d9
      AOP_FONTMAP -> yes
   }
;
transform afp2pcl_eu
   start-command = afp2pcld
   min-active = 0   max-active = 2
   maximum-idle-time = 300 # 5 minutes
   environment = {
      AOP_FONTLIB -> "sys1.font300"
      AOP_PAGEDEFLIB -> "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 1 2"
      AOP_PAPER -> "a4 a3 a4ee a4 a4 a4 a4 a4 a4 c5"
      AOP_COLOR -> no
      AOP_CUTSHEET -> yes
      AOP_PJL -> yes
      AOP_MSGPAGEDEF -> p1q09182
      AOP_MSGFORMDEF -> f1cp0110
      AOP_PAGEDEF -> p1q09182
      AOP_FORMDEF -> f1cp0110
      AOP_CHARS -> 60d9
      AOP_FONTMAP -> yes
   }
;
```

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.

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.

Character Set Names of Raster Font	Character Set Name of Outline Font	Value in Font-Mapping 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 134 for more information. Figure 23 on page 135 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:

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 usermod.

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 3-6 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 23 on page 135 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 23. Sample Font-Mapping Table — SYS1.SAMPLIB(AOXFONTS)

Scaling 240-Pel to 300-Pel Fonts using the AOXCF30 Program

The AOXCF30 font-conversion program provided with Infoprint Server Transforms lets you scale (that is, convert) your 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.

You must scale 240-pel fonts to 300-pel fonts if you do not already have 300-pel fonts and either of these conditions apply:

- You use the AFP to PCL transform. This transform requires 300-pel fonts.
- You use the AFP to PostScript or AFP to PDF transform, documents to be transformed reference raster fonts, and you do not map raster fonts to outline fonts.

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 the AOXCF30 program to scale them again. Because IBM Core Interchange raster fonts are shipped in both 240-pel and 300-pel resolutions, you do not need to convert them. Outline fonts are resolution-independent and do not need to be converted.

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.

Using the AOXCF30 Program

To use the AOXCF30 program, follow these steps:

1. Define a new font library for 300-pel fonts if one does not already exist. SYS1.FONT300 is the recommended name for your 300-pel system font library.

You can use the existing 240-pel font library as a model; however, the 300-pel font library requires approximately 30% more space than the 240-pel font library.

2. Create a list of 240-pel fonts that require conversion.

   Only the character set members, whose file names start with `C0`, are converted with AOXCF30; however, you can also specify coded font members, whose file names start with `X0`, and code page members, whose file names start with `T0`; the coded fonts and code pages are copied to the output file without conversion.

3. Modify the JCL supplied in SYS1.SAMPLIB(AOXCF30J).

   Specify an entry in the JCL for each font member to be converted. The sample JCL is shipped with two dummy entries:

   ```
   //C0XXXXXX EXEC PROC=SCALE,NAME=C0xxxxxx
   //C0YYYYYY EXEC PROC=SCALE,NAME=C0yyyyyy
   ```

   Duplicate these statements for each font that is to be converted.

   For example, specify:

   ```
   C0A055A0  EXEC  PROC=SCALE,NAME=C0A055A0
   C0A055B0  EXEC  PROC=SCALE,NAME=C0A055B0
   ```

   Figure 24 on page 137 shows a portion of SYS1.SAMPLIB(AOXCF30J).

4. Submit the JCL to run the AOXCF30 program.

   This program can use a large amount of CPU resources, especially if you run it on a smaller S/370 processor. To run the AOXCF30 program, you should have a minimum region size of 5 MB.

DD Statements Used by the AOXCF30 Program

Do not convert the Bar Code or Optical Character Recognition fonts (Program Number 5688-021). The conversion process usually distorts the font, making it unreadable by an optical scanning device. Check all the JCL for this font before running the AOXCF30 program.

The AOXCF30 program uses the following DD statements:

DD Name	Description
SYSPRINT	Program log and error message output.
SYSUT1	Input font library, including data set and member name
SYSUT2	Output font library, including data set and member name
FONTB300	Box character source library, which is supplied with the AOXCF30 program.

Parameters Used by the AOXCF30 Program

You can specify the following parameters to the AOXCF30 program in the PARM1 parameter.

Parameter	Description
BOX	Indicates that substitution of box characters will be performed (default).
NOBOX	Indicates that substitution of box characters will *not* be performed.

Box character substitution is performed in conjunction with fixed-pitch fonts. When the BOX parameter is specified, the AOXCF30 program replaces box characters in the font with characters from the supplied AOX.SAOXCF30 library. To disable this

substitution, specify the NOBOX parameter. For example, specify NOBOX when you scale custom built fonts, such as logo fonts.

Sample JCL for the AOXCF30 Program

Figure 24 shows JCL provided in SYS1.SAMPLIB(AOPCF30J).

```
//*******************************************************************
//*  LICENSED MATERIALS - PROPERTY OF IBM                           *
//*  5697-B51                                                       *
//*  (C) COPYRIGHT IBM CORP. 2000                                   *
//*  AFP TRANSFORM (C) COPYRIGHT I-DATA INTERNATIONAL 1990,2000     *
//*                                                                 *
//*  AOXCF30 PROGRAM                                                *
//*                                                                 *
//*  CONVERTS SBCS AND DBCS BOUNDED BOX FONTS FROM 240 TO 300 DPI   *
//*  USING A PROC INVOCATION OF THE AOXCF30 FONT SCALING PROGRAM.   *
//*  THE SUPPLIED PROC STATEMENTS ARE PROVIDED AS A SAMPLE ONLY, AND *
//*  SHOULD BE TAILORED TO MATCH INSTALLATION REQUIREMENT FOR       *
//*  SCALING OF FONTS TO 300 DPI.                                   *
//*                                                                 *
//*  BOX DRAWING CHARACTERS LOCATED IN FONTS CAN BE REPLACED USING  *
//*  THE BOX PARAMETER, AND SUPPLIED BOX DRAW CHARACTERS (FONTB300) *
//*                                                                 *
//*  PARM='BOX' USE SUPPLIED BOX CHARACTER SET (DEFAULT)            *
//*  PARM='NOBOX' DO NOT USE THE SUPPLIED BOX CHARACTER SET         *
//*                                                                 *
//* S2      EXEC PROC=SCALE,NAME=C0XXXXXX,PARM='BOX'                *
//*                            (FONT IMAGE MEMBER NAME)             *
//*                                                                 *
//*                                                                 *
//*******************************************************************
//*
//SCALE     PROC HLQ=AOX,NAME=,PARM1=BOX
//S1        EXEC PGM=AOXCF30,REGION=5120K,PARM='&PARM1'
//SYSPRINT  DD SYSOUT=*
//*    SYSUT1 IS THE INPUT 240 DPI FONT LIBRARY
//SYSUT1    DD DISP=SHR,DSN=SYS1.FONTLIBB(&NAME.)
//*    SYSUT2 IS THE OUTPUT 300 DPI FONT LIBRARY
//SYSUT2    DD DISP=SHR,DSN=SYS1.FONT300(&NAME.)
//FONTB300  DD DISP=SHR,DSN=&HLQ..SAOXCF30
//          PEND
//*
//*
//C0XXXXXX  EXEC PROC=SCALE,NAME=C0xxxxxx
//C0YYYYYY  EXEC PROC=SCALE,NAME=C0yyyyyy
```

Figure 24. Sample JCL to Run the AOXCF30 Font-Scaling Program — SYS1.SAMPLIB(AOXCF30J)

AOXCF30 Return Codes

The AOXCF30 program detects each of the errors listed below and places the return code for it in register 15. The return code prints on the screen when you use the AOXCF30 EXEC. When an error occurs, the font-conversion program might create an incomplete output file, which you should discard.

Code	Description
4	The AOXCF30 program was unable to locate the box character member. Ensure that the FONTB300 DD name correctly specifies the SAOXCF30 library.
20	The AOXCF30 program was unable to open the output data set, or an I/O error occurred while writing to the output data set.

28	The AOXCF30 program was unable to open the input data set.
32	The font named in the input data set is not valid. Ensure that the input font member is a valid 240-pel bounded box.
44	An unexpected end-of-file occurred on the input data set. Ensure that the input data is not corrupted.
88	Insufficient storage was available for program execution.
100	The AOXCF30 program was unable to write to the SYSPRINT data set.

Chapter 6. Customizing the SNMP Subagent

This chapter describes how to customize the SNMP subagent component of Infoprint Server. You must customize the SNMP subagent to monitor printer characteristics and printer status for printers controlled by PSF for OS/390 that are either not TCP/IP-attached or lack internal SNMP agents. For example, you can monitor channel-attached printers.

The Infoprint Server SNMP subagent works in conjunction with PSF for OS/390, the z/OS Communications Server SNMP agent, and an SNMP manager application, such as IBM Network Printer Manager for the Web. See "SNMP Subagent" on page 19 for an overview of the SNMP subagent and how it fits into your z/OS system.

This chapter describes the following customization tasks. All tasks are required if you want to use the SNMP subagent.

Task	Condition	See Page:
Identifying Software Requirements	Required	139
Customizing and Starting the z/OS Communications Server SNMP Agent	Required	140
Editing the **aopd.conf** Configuration File	Required	141
Enabling SNMP Reporting in the Printer Inventory	Required	141
Customizing the SNMP Manager	Required	142

The following table lists other components of Infoprint Server that you must customize:

Task	Condition	See Page:
Customizing the Printer Inventory Manager	Required	25

Related task: After you customize the SNMP subagent, the operator must perform the following tasks. Refer to *z/OS Infoprint Server Operation and Administration* for information about how to start the z/OS SNMP agent and the Infoprint Server SNMP subagent.

- Start the z/OS SNMP agent if it is not already started.
- Restart Infoprint Server to start the SNMP subagent.
- Start each PSF printer at least once after enabling SNMP reporting in the Printer Inventory and before defining PSF printers to an SNMP manager.

Identifying Software Requirements

To use the SNMP subagent of Infoprint Server, the following additional IBM software beyond the z/OS base elements is required:

- IBM Network Printer Manager (NPM) for the Web or an SNMP manager with equivalent function. The SNMP manager must be able to communicate with an SNMP agent that supports the general printer MIB (defined in RFC 1759), and the SNMP manager must support multiple printers defined with one IP address (the IP address of the z/OS system).

© Copyright IBM Corp. 1995, 2002

The NPM for the Web server runs on a Windows NT workstation and requires the following software:

- Windows NT Version 3.51 or Version 4.0, with TCP/IP protocol network software installed
- A Web server running under Windows NT
- Java Virtual Machine software Version 1.1

The NPM Graphical User Interface (GUI) can run on the same system as the NPM server or a different system. The NPM GUI requires a Java 1.1 enabled Web browser, which can be either Netscape Communicator 4.04 (or later) or Microsoft Internet Explorer 4.0 (or later).

Customizing and Starting the z/OS Communications Server SNMP Agent

You must configure and start the z/OS Communications Server (CS) for SNMP agent before you start the Infoprint Server SNMP subagent. For information about how to configure and start the z/OS CS SNMP agent, refer to *z/OS Communications Server: IP Configuration Reference*.

Following are some considerations related to configuring and starting the z/OS CS SNMP agent:

- The community-name function of SNMP can provide a higher-level of security for your installation; however, IBM does *not* recommend that you use the community-name function with the Infoprint Server SNMP subagent because the Infoprint Server SNMP subagent does not let administrators change printer characteristics.

  If your installation wants to use the community-name function, then you must specify the *same* community name to (1) the z/OS CS SNMP agent, (2) the TCP/IP SNMP subagent, (3) the Infoprint Server SNMP subagent, and (4) the SNMP manager:

  - Specify the community name to the z/OS CS SNMP agent on the **-c** parameter when you start the SNMP agent. The z/OS CS SNMP agent also lets you specify a community name in the SNMPD.CONF or PW.SRC file; however, IBM recommends that you do *not* create an SNMPD.CONF file or a PW.SRC file unless you want to restrict the community name to use by a certain group of IP addresses (of SNMP managers). In this case, create a PW.SRC file, and do not specify the **-c** parameter when you start the z/OS CS SNMP agent.
  - Specify the community name to the TCP/IP SNMP subagent in the COMMUNITY keyword of the SACONFIG statement in the *hlq*.PROFILE.TCPIP data set.
  - Specify the community name to the Infoprint Server SNMP subagent in the **snmp-community** statement in the **aopd.conf** configuration file.
  - See the online help for the NPM Manager for the Web for information about how to specify the community name.
- Provide SNMPTRAP.DEST information to the z/OS CS SNMP agent. SNMPTRAP.DEST information defines the IP addresses of the SNMP managers to which the z/OS CS SNMP agent sends traps. When you provide trap information, the z/OS SNMP agent can notify the SNMP manager of printer alert conditions more quickly.

- Do *not* reserve ports 161 and 162 for the z/OS CS SNMP agent in the *hlq*.PROFILE.TCPIP data set. See "Customizing z/OS Communications Server" on page 51 for more information.

Editing the aopd.conf Configuration File

You must customize the Infoprint Server **aopd.conf** configuration file before starting the SNMP subagent. Add or edit these statements in the configuration file. See Appendix B, "Infoprint Server Files" on page 245 for more information about the configuration file.

snmp-community = *name*

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 z/OS 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`

start-daemons = {snmpd}

To start the SNMP subagent with the **aopstart** command, add **snmpd** to the values in this statement. Enclose the values in braces. By default, only the Printer Inventory Manager and the LPD start; therefore, this statement is required to start the SNMP subagent.

Default: `start-daemons = {lpd}`

Enabling SNMP Reporting in the Printer Inventory

The administrator can enable SNMP reporting for a PSF-controlled printer in the FSA definition for the PSF printer in the Printer Inventory. By default, SNMP reporting is disabled. To enable SNMP reporting, the administrator must select the **SNMP reporting** field on the ISPF panel used to add or edit the FSA definition.

If a TCP/IP-connected printer contains an internal SNMP agent, IBM recommends that you define the printer directly to the SNMP manager and not enable SNMP reporting in the FSA definition. When you define a printer directly to the SNMP manager, the administrator can also view printer statistics and change some printer characteristics. Refer to the documentation for your TCP/IP-enabled printers to determine if they have internal SNMP agents.

The administrator should enable SNMP reporting in the Printer Inventory and also start the PSF printers before defining the PSF-contolled printers to the SNMP manager.

Refer to *PSF for OS/390 & z/OS: Customization* for information about how to customize PSF to use information in the Printer Inventory and enable SNMP reporting. Refer to *z/OS Infoprint Server Operation and Administration* for information about how to add or edit an FSA definition.

Customizing the SNMP Manager

To monitor PSF-controlled printers, you must install and use an SNMP manager. The IBM Network Printer Manager (NPM) for the Web contains the necessary support to display the status of a print system (such as a z/OS system) that contains multiple printers. When you use NPM for the Web, administrators can use a Java-enabled version of Netscape Navigator or Microsoft Internet Explorer to monitor printers; the NPM server runs on a Windows NT system. NPM for the Web limits the number of printers you can monitor within each print system; see the NPM online help for the maximum number of printers NPM allows.

To download the latest version of NPM for the Web visit the IBM Printing Systems Division Web site: http://www.ibm.com/printers/download.html. Use the NPM README file and online help for information about how to install and use NPM.

Following are some considerations related to customizing NPM for the Web:

- To add the z/OS print system to NPM, you need to know the IP address (in dotted-decimal notation) of the z/OS host and the community name that you defined to the z/OS SNMP agent and to the SNMP subagent. If you have not defined a community name, use the default **public**.
- Before you add the z/OS print system to NPM, you must (1) start the z/OS SNMP agent and SNMP subagent, (2) enable SNMP reporting for PSF printers in the Printer Inventory, and (3) start or restart the PSF printers for which you enabled SNMP reporting.

  Any PSF printer can be inactive when you define the z/OS print system, provided the printer has been started at least once after SNMP reporting was enabled in the Printer Inventory. If you do not start *any* PSF printers after enabling SNMP reporting, NPM displays a status of NOT SUPPORTED. If this situation occurs, cycle the NPM server.
- When you add the z/OS print system to NPM, NPM automatically discovers each PSF printer for which SNMP reporting is enabled in the Printer Inventory. NPM lists the FSA definition name for the PSF printers it discovers. If a PSF printer is not in the list, start or restart the PSF printer and cycle the NPM server.
- If you enable SNMP reporting for a PSF printer in the Printer Inventory after adding the z/OS print system to NPM, start or restart the PSF printer and cycle the NPM server.
- If you disable SNMP reporting for a PSF printer in the Printer Inventory after adding the z/OS print system to NPM, cycle the NPM server.

Chapter 7. Customizing NetSpool

This chapter describes the NetSpool component of Infoprint Server. You must customize NetSpool if you want to print from VTAM applications such as CICS and IMS.

NetSpool automatically intercepts print requests from VTAM applications, converts the input SCS or 3270 data streams to either line data streams or PCL data streams, and allocates output data sets on the JES spool. See “NetSpool” on page 14 for an overview of NetSpool and how it fits into your system.

This chapter describes the following customization tasks. Required tasks are required by all installations that want to use NetSpool. Optional tasks are required only if the listed condition applies to your installation.

Task	Condition	See Page:
Customizing JES to Specify No Blank Truncation	Required	143
Allocating and Initializing the Message Log	Optional: To record messages in a log	144
Customizing the NetSpool Startup Procedure	Required	144
Changing LE Run-time Options	Optional: To change default options	150
Writing NetSpool Exit Programs	Optional: To add, modify, or delete transparent data in SCS data	150

The following table lists other components of Infoprint Server that you might need to customize:

Task	Condition	See Page:
Customizing the Printer Inventory Manager	Required	25
Customizing IP PrintWay	Optional: To print data to remote printers in your TCP/IP network	163

Related tasks: After you customize NetSpool, the administrator and operator must perform the following tasks. Refer to *z/OS Infoprint Server Operation and Administration* for information about these tasks:

- The administrator must create or update the printer definition for each printer in the Printer Inventory and specify printer attributes required by NetSpool.
- The administrator must define each NetSpool printer logical unit (LU) to VTAM, and modify VTAM resource definitions if necessary.
- The operator must start NetSpool and vary the NetSpool printer LUs active.

Customizing JES to Specify No Blank Truncation

JES2 and JES3, by default, remove EBCDIC blank characters from the end of each record on the JES spool in order to save spool space. We recommend that you request no blank truncation for those output classes in which NetSpool allocates output data sets. This is because some data streams, for example PCL data

© Copyright IBM Corp. 1995, 2002

streams in ASCII representation and line data streams in EBCDIC representation that contain transparent data, might not print properly.

NetSpool allocates output data sets in the output classes specified in the CLASS fields of the printer definitions. If no output class is specified in a printer definition, NetSpool allocates output data sets in the JES default output class, typically class A.

To request no blank truncation, specify the following JES parameters:

- BLNKTRNC=NO on the JES2 OUTCLASS statement. The default is BLNKTRNC=YES.
- TRUNC=NO on the JES3 SYSOUT statement. The default is TRUNC=YES; however, you can change the default in the TRUNC parameter of the JES3 BUFFER statement.

For example, if NetSpool can allocate data sets in output class A and P, specify BLNKTRNC=NO (in JES2) and TRUNC=NO (in JES3) for both classes A and P.

For more information, refer to the following publications:

- *z/OS JES2 Initialization and Tuning Reference* for information about the JES2 OUTCLASS statement
- *z/OS JES3 Initialization and Tuning Reference* for more information about the JES3 SYSOUT statement

Allocating and Initializing the Message Log

NetSpool can write informational and error messages in a NetSpool message-log data set. The NetSpool message-log data set is an optional data set that 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.

To use the NetSpool message log, you must allocate space for the data set and initialize it. You can 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 requires the following specifications:

- 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

Secondary space is unnecessary, because the ANFMFILE program formats all available space. To change the size of the NetSpool message-log data set, rerun the ANFMFILE program.

Note: The ANFMFILE program produces a D37 abend when it completes processing.

Customizing the NetSpool Startup Procedure

Before starting NetSpool, you must create a startup 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

The following table summarizes the customization tasks related to setting up the startup procedure. Required tasks are required by all installations; optional tasks are required only if the listed condition applies to your installation.

Task	Condition	See Page:
Determining the Number of NetSpool Programs to Start	Required	145
Using the Japanese National Language Version	Optional: To receive messages in Japanese	145
Using an Extended MCS Console	Optional: To receive unsolicited messages at an extended console	146
Creating a User ID to Assign to the NetSpool Startup Procedure	Required	146
Defining the NetSpool Startup Procedure to RACF	Required	146
Editing the NetSpool Startup Procedure	Optional: To change default settings	147

Determining the Number of NetSpool Programs to Start

You can run multiple instances of NetSpool simultaneously, each instance running in a different address space. You might want to run multiple instances for the following reasons:

- 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

Using the Japanese National Language Version

A Japanese National Language Version (NLV) is available for NetSpool. If you ordered the Japanese NLV, a new message data set is provided in a separate FMID. To install the message data set associated with the language feature, refer to *z/OS Program Directory*.

To receive messages in Japanese, specify the LANGUAGE parameter in the EXEC statement, using the last three characters in the name of the message data set. See “Editing the NetSpool Startup Procedure” on page 147 for more information about the LANGUAGE parameter.

If the language feature is installed, you can still use the English message data set by either removing the LANGUAGE parameter from the EXEC statement or by specifying LANGUAGE=ENU and starting NetSpool again.

Be certain your console supports the 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.

You must configure an extended MCS console on TSO/E to view messages in Japanese. Refer to *z/OS TSO/E Customization* for information about customizing TSO/E. Messages issued by NetSpool are already translated into Japanese.

Using an Extended MCS Console

When using extended MCS consoles, messages issued in response to NetSpool operator commands are routed to the console that issued the command. NetSpool sends unsolicited messages to an extended MCS console when the CONSNAME parameter in the startup procedure specifies the name of an extended MCS console.

Creating a User ID to Assign to the NetSpool Startup Procedure

The user ID assigned to the NetSpool startup procedure must be defined to RACF with access to z/OS UNIX.

Steps for creating a RACF user ID with access to z/OS UNIX:

1. Create a user ID with an OMVS segment, a home directory, and a default group that has an OMVS segment with a group identifier (GID).

   You can select any name for the user ID. Use the RACF ADDUSER command to create a new user ID; use the RACF ALTUSER command to alter an existing user ID to give it access to z/OS UNIX. For the default group, you can select any group that has an OMVS segment and a GID. The user ID must be connected to the default group.

   For example, the following RACF command defines user ID AOPSTC with a default group of AOPOPER. For *user-identifier*, specify an integer that is different from other UIDs in your installation. The NOPASSWORD operand defines the user ID as a *protected* user ID; protected user IDs cannot log on to the z/OS system, and they cannot be revoked due to incorrect password attempts.

   ```
   ADDUSER AOPSTC OMVS(UID(user-identifier) HOME('/u/aopstc')
      PROGRAM('/bin/sh')) DFLTGRP(AOPOPER) NOPASSWORD
   ```

2. Create the home directory specified in the ADDUSER command. The following z/OS UNIX commands switch to an effective UID of 0 and create a home directory for user ID AOPSTC with the required permissions:

   ```
   su 0
   cd /u
   mkdir -m 0700 aopstc
   chown -R AOPSTC /u/aopstc
   ```

   To use the **su** command, you must be permitted to the BPX.SUPERUSER profile in the FACILITY class within RACF.

3. Give the user ID RACF update access to the NetSpool message-log data set. If the name of the data set is USER.MSGFILE, then the following RACF command gives update access to user ID AOPSTC:

   ```
   PERMIT USER.MSGFILE ACCESS(UPDATE) ID(AOPSTC)
   ```

Defining the NetSpool Startup Procedure to RACF

You must define the NetSpool startup procedure to the RACF STARTED profile in the FACILITY class. If you want to use the started procedure table (ICHRIN03) instead, refer to *z/OS Security Server RACF Security Administrator's Guide* for more information.

The following RACF commands define the APIJPJCL procedure to RACF. This example assumes that the user ID you defined for use with this procedure is AOPSTC and that AOPSTC is connected to group AOPOPER.

```
RDEFINE STARTED APIJPJCL.* STDATA(USER(AOPSTC) GROUP(AOPOPER))
SETROPTS RACLIST(STARTED) REFRESH
```

Editing the NetSpool Startup Procedure

NetSpool provides a startup procedure in SYS1.PROCLIB (APIJPJCL). You can copy and modify this procedure for your installation.

Steps for Editing the NetSpool Startup Procedure

1. Copy the SYS1.PROCLIB(APIJPJCL) procedure to a procedure library that is known to JES. For information about procedure libraries, refer to *z/OS JES2 Initialization and Tuning Guide* and *z/OS JES3 Initialization and Tuning Guide*.
2. Edit the APIJPJCL procedure if any of the following conditions apply to your installation:
   - If you did not add the Language Environment run-time library, CEE.SCEERUN, to the system LNKLST, add the SCEERUN data set to the STEPLIB DD statement.
   - If you did not add the C++ run-time library, CBC.SCLBDLL, to the system LNKLST, add the SCLBDLL data set to the STEPLIB DD statement.
   - If you use a NetSpool exit and the library that contains the exit is not in the standard z/OS search order, add the library to the STEPLIB DD statement.
   - If you want to start NetSpool LU classes other than class 1, specify the classes in the LUCLASS parameter of the PROC statement.
   - If the Printer Inventory name is not the default name, AOP1, specify the inventory name in the INV parameter of the PROC statement.
   - If you do not use the default name for the NetSpool message-log data set, change the data set name in the MSGFILE parameter of the PROC statement.
   - If you want to receive NetSpool messages in Japanese, specify the LANGUAGE parameter in the EXEC statement.
   - If you want to receive NetSpool unsolicited messages on an alternate console, specify the CONSNAME parameter in the EXEC statement.

   This procedure uses symbolic parameters with default values for some of the parameters that you might want to change for your installation. Refer to *z/OS MVS JCL Reference* for more information on symbolic parameters.
3. If NetSpool is running, restart it to pick up the changes. Refer to *z/OS Infoprint Server Operation and Administration* for information about how to stop and start NetSpool.

APIJPJCL Procedure

Figure 25 on page 148 shows the NetSpool startup procedure provided in SYS1.PROCLIB.

```
//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 25. NetSpool Startup Procedure — SYS1.PROCLIB(APIJPJCL)

The JCL statements and parameters that you can include in your NetSpool startup procedure are:

label **PROC**
Marks the beginning of the procedure. The sample PROC statement in Figure 25 specifies default values for symbolic parameters coded in the sample. If you plan to use the sample procedure, modify these default values to suit your installation.

label **EXEC PGM=APIPPAAA,REGION=***size***,TIME=NOLIMIT, PARM='LUCLASS=(***class***[,...]),INV=***inventory***[,LANGUAGE=***language_id***] [,CONSNAME=***console_name***][,TRACE=ON]'**
Invokes the NetSpool program.

PGM=APIPPAAA
The NetSpool program entry point.

REGION=*size*
The maximum amount of virtual storage NetSpool can use. Refer to *z/OS MVS JCL Reference* for more information about specifying region size.

TIME=NOLIMIT
Specify TIME=1440 or TIME=NOLIMIT to prevent TIMEOUT abends.

PARM=
Specify a maximum of 100 characters for the value of the PARM parameter, as described in *z/OS MVS JCL Reference*.

LUCLASS=(*class***[,...])**
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 *z/OS 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 *z/OS Infoprint Server Operation and Administration*.

This parameter is required.

INV=*inventory*
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=*console_name*
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. You can abbreviate the CONSNAME parameter as follows: **CON=***console_name*.

Translated NetSpool messages can be displayed correctly when CONSNAME specifies the name of an extended MCS console that is configured for the language of the translated messages.

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=*language_id*
The language that NetSpool is to use for messages.

Note: You can abbreviate this parameter as follows: **LNG=***language_id*

NetSpool appends the *language_id* to the prefix APIMT to determine the name of the message data set to load. If you specify Japanese, 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.)

Example:

```
LNG=JPN
```

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.

STEPLIB DD DSN=*dsname***,DISP=SHR**
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 any of the IBM-supplied exit programs.

Also, concatenate the LE and C++ run-time libraries (CEE.SCEERUN and CBC.SCLBDLL) if you did not add them to the system LNKLST.

APIMMSG DD DSN=*dsname***,DISP=SHR**
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. See “Allocating and Initializing the Message Log” on page 144 for more information.

TRACEOPT DD DSN=*dsname***,DISP=SHR**
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.

Changing LE Run-time Options

Figure 26 shows the run-time options for the LE environment that NetSpool establishes.

```
TRAP(OFF)
POSIX(ON)
ALL31(ON)
ENVAR("_CEE_ENVFILE=//DD:STDENV")
RPTOPTS(ON)
RPTSTG(ON)
MSGFILE(,,,,ENQ)
STACK(8K,64K,ANY,KEEP)
HEAP(8K,8K,,FREE)
ANYHEAP(8K,8K)
THREADHEAP(8,4K,ANY,FREE)
BELOWHEAP(8,4K,FREE)
LIBSTACK(8,0)
```

Figure 26. LE Run-time Options for NetSpool

To change any of the LE run-time options, follow these steps:

1. Create a CEEUOPT object file. You can use sample job CEE.SCEESAMP(CEEWUOPT) to create the CEEUOPT.
2. Create an SMPE usermod to link-edit the CEEUOPT object file with NetSpool module SYS1.LINKLIB(APIIOPN). Place the CEEUOPT text deck first when you link-edit it with module APIIOPN.
3. Put the new module into a link library and add that library to the NetSpool startup procedure with a STEPLIB or JOBLIB DD statement.

Refer to the following publications for more information:

- For information about LE run-time options, refer to *z/OS Language Environment Programming Guide*.
- For information about how to create a CEEUOPT file and link edit it with another module, refer to *z/OS Language Environment Customization*.

Writing NetSpool Exit Programs

Note
This section contains programming-interface information.

NetSpool provides the following exits, which let you customize how NetSpool processes SCS and 3270 data streams:

- The Beginning of File exit for SCS data streams (APIPPTD1)

  NetSpool calls this exit each time it creates a new output data set.
- The Beginning of File exit for 3270 data streams (APIUBF3)

  NetSpool calls this exit each time it creates a new output data set.

- The Transparent Data Control exit for SCS data streams (APIPPTD2)

  NetSpool calls this exit when it finds the Transparent Data control (TRN) in the input data stream.
- The Graphic Escape exit for SCS and 3270 data streams (APIUGEX)

  NetSpool calls this exit when it finds a Graphic Escape SCS code, X'08', in the input data stream.

Programming Considerations

Consider the following points when coding exit programs:

- You should be an experienced programmer and aware of overall system implications.
- Do not issue an explicit or implied WAIT in an exit, because it puts NetSpool into a WAIT state.
- The exits can run in either 24-bit or 31-bit addressing mode. 31-bit addressing is recommended, to allow more printer LUs to run in an address space.
- Use only the control block passed to the exit. Code references to fields in the control block using the names associated with each field in the DSECT control block. Do not use absolute displacements, because these displacements can change from one release to another.
- Code the exit to be reentrant for improved performance; however, if you replace an exit, you must stop and restart NetSpool to pick up the changed exit. If the exit is non-reentrant, you can replace the exit while NetSpool is running.
- The exit programs run in supervisor state in key 8.
- An ESPIE is in effect while the exit programs are running. Programming exceptions cause the printer session and the application session to terminate.

Notes:

1. If you are migrating from a release prior to OS/390 V2R8, be sure to relink your exits with the AC(1) attribute and place them in an APF-authorized library. You do not need to reassemble them.
2. Future NetSpool program maintenance might require that you reassemble these exits.

APIPP1X2 Control Block

When the exits are called, register 1 contains the address of the APIPP1X2 control block, mapped by DSECT S2STG, shown in Figure 27 on page 152.

```
| Offset  (Hex) Type     Length  Name      Description
|
|      0    (0) Structure     *  S2STG     Exit parameter area
|      0    (0) Character     5  S2EYEC    Set to 'S2STG'
|      5    (5)               1  *         Reserved
| ==========================================================================================
| The following flags are input fields to all exits. The exits must not change these fields.
| ==========================================================================================
|      5    (5) Unsigned      1  S2FLAGS   Flags for LU type and formatting option
|               ...1 ....        S2LU1     Set for LU1; not set for LU0 and LU3
|               .... ...1        S2FPCL    Set when NetSpool converts to PCL
| ==========================================================================================
| The following fields are input to APIPPTD1 and APIPPTD2. The exits must not change these fields.
| ==========================================================================================
|                                          Horizontal Page Format:
|      6    (6) Signed        2  S2COL       Next column number in output
|      8    (8) Signed        2  S2LM        Left margin column number
|     10   (0A) Signed        2  S2RM        Right margin column number
|     12   (0C) Signed        2  S2MPP       Maximum presentation position (line length)
|                                          Vertical Page Format:
|     14   (0E) Signed        2  S2LINE      Next output line number
|     16   (10) Signed        2  S2TM        Top margin line number (channel 1)
|     18   (12) Signed       22  S2C2T12     Channel 2-12 line numbers, 2 bytes per channel
|     40   (28) Signed        2  S2BM        Bottom margin line number
|     42   (2A) Signed        2  S2MPL       Maximum page length
| ==========================================================================================
| The following fields are input to all exits. The exits must not change these fields.
| ==========================================================================================
|     44   (2C) Character     8  S2PLUNAM  LU name of the application generating
|                                           the print request - primary LU (PLU)
|     52   (34) Character     8  S2LPTNAM  Name of the printer LU - secondary LU (SLU)
|     60   (3C) Character     8  S2PRTCLS  Classes of the printer LU.
|                                          Each bit represents a class, from 1 to 64:
|                                          Value   Class
|               1... ....                    1    Class 1
|               .1.. ....                    1    Class 2
|               .... ...1                    1    Class 8
|                                          This pattern repeats for 7 more bytes.
|                                          More than one bit can be set at a time.
| ==========================================================================================
| The following field can be set by exit APIPPTD1.
| ==========================================================================================
|     68   (44) Character     1  S2DBCSM   Number of bytes to allocate per DBCS character:
|                                          Value    Meaning
|                                           0     Two positions per character (default)
|                                           1     One position per character
|     69   (45)               1  *         Reserved
| ==========================================================================================
| The following fields can be set by exits APIPPTD1 and APIPPTD2.
| ==========================================================================================
|     70   (46) Signed        2  S2NLNUM   New line number for output
|     72   (48) Signed        2  S2NCNUM   New column number for output
```

Figure 27. Format of APIPP1X2 Control Block (Part 1 of 2)

```
| ==========================================================================================
| The following field is input to exits APIPPTD2 and APIUGEX. The exits must not change this field.
| ==========================================================================================
|     76   (4C) Character   275  S2TPDATA  APIPPTD2: Transparent data found in input data:
|                                            1 byte: X'35'
|                                            1 byte: Length of data
|                                            * bytes: Transparent data
|                                          APIUGEX: Graphic Escape character in EBCDIC:
|                                            1 byte: In EBCDIC
|                                            1 byte: In printer code page
|    351  (15F)               5  *         Reserved
| ==========================================================================================
| The following fields can be set by all exits.
| ==========================================================================================
|    356  (164) Signed        4  S2WRKLEN  Length of data in S2WRKARA
|    360  (168) Character   275  S2WRKARA  Data returned by exit to include in output
|                                           APIPPTD1: SCS controls or transparent data
|                                           APIPPTD2: Modified transparent data and SCS controls
|                                           APIUBF3:  Line data or PCL data to add
|                                           APIUGEX:  Line data or PCL data to replace the Graphic
|                                                      Escape character sequence X'08XX'
```

Figure 27. Format of APIPP1X2 Control Block (Part 2 of 2)

Beginning of File Exit for SCS Data Streams (APIPPTD1)

The Beginning of File exit for SCS data streams (APIPPTD1) receives control when NetSpool starts a new output data set for SCS input data streams. You can use this exit to add up to 275 bytes of data to the start of a data set or to close the VTAM session with the printer.

NetSpool calls the same exit for all LU type 1 sessions. However, your exit can perform different functions based on the following input fields in the APIPP1X2 control block:

- S2FLAGS: Type of output data stream NetSpool creates for this printer— a line or PCL data stream
- S2LPTNAM: NetSpool printer name, also known as the secondary LU (SLU) name
- S2PLUNAM: LU name of the VTAM application, also known as the primary LU (PLU) name
- S2PRTCLS: LU classes to which the NetSpool printer belongs

You can use this exit to add SCS controls, printable text, and PCL commands. Follow these guidelines when you add data:

- You can add any SCS controls that NetSpool supports. Refer to the appendixes in *z/OS Infoprint Server User's Guide* for a summary of the SCS controls that NetSpool supports.
- If the printer is controlled by IP PrintWay, you can add text data and any PCL commands that your printer supports as transparent data. Code text data in ASCII representation, and precede text and PCL commands with an SCS Transparent (TRN) control — X'35' followed by a 2–byte length field. Either NetSpool or IP PrintWay removes the SCS TRN control before transmitting the text and PCL commands, unchanged, to the printer.
- If you add transparent data in a TRN control, you might need to adjust the column and line number of the next print position. By default, when NetSpool finds a TRN control, NetSpool increases the column and line number at which

the next character prints by the length of the transparent data, as specified in the TRN control. In your exit, you can change the column and line number where the next character after the transparent data prints. To assist in calculating the desired line and column numbers, NetSpool provides the exit with the column and line number of the next print position before any transparent data is added.

- Any PCL commands that you add in this exit override any PCL commands specified in the IP PrintWay **Document header** field in the printer definition. They also override any PCL commands that NetSpool automatically generates as a result of PCL conversion values specified in the printer definition, such as the print density or line density.

Input to APIPPTD1

At input to the Beginning of File exit (APIPPTD1), the contents of the registers are:

Register 1 Address of the APIPP1X2 control block

Register 13 Address of an 18-word save area for saving the caller's registers

Register 14 Return address

The APIPP1X2 control block is described in Figure 27 on page 152. The following fields are valid at entry:

S2FLAGS Type of conversion and VTAM session:

- **S2LU1** Set for VTAM LU1 session; not set for VTAM LU0 or LU3 sessions.
- **S2PCL** Set when NetSpool will convert the input data stream to a PCL data stream.

S2COL The column number of the next print position

S2LM Left margin column number

S2RM Right margin column number

S2MPP Maximum presentation position (line length)

S2LINE The line number of the next print position

S2TM Top margin line number (channel 1)

S2C2T12 Channel 2-12 line numbers

S2BM Bottom margin line number

S2MPL Maximum page length

S2PLUNAM LU name of the application generating the print request (PLU)

S2LPTNAM Name of the printer LU that NetSpool is processing (SLU)

S2PRTCLS Logical unit classes to which the printer LU belongs. Each bit in this field represents a class, with the high order-bit of the leftmost byte representing class 1. See Figure 27 on page 152 for a description of the bits.

Output From APIPPTD1

At return from the Beginning of File exit (APIPPTD1), restore the contents of all registers, and place one of the following codes in register 15 to tell NetSpool what action to take:

Code	Meaning
1	Add the data from S2WRKARA to the output data set.
2	Do *not* add data from S2WRKARA to the output data set.
4	Close the VTAM session for this printer LU.
8	Do *not* call exit APIPPTD2 for any transparent data found in the data stream for this output data set. Do *not* add data from S2WRKARA to the output data set.
9	Do *not* call exit APIPPTD2 for any transparent data found in the data stream for this output data set. Add the data from S2WRKARA to the output data set.

Also, set the following fields in the APIPP1X2 control block.

S2DBCSM The number of character positions NetSpool is to allocate in an output line for each double-byte character.

Value	Meaning
0	Allocate two print positions for each DBCS character. This is the default value set by NetSpool.
1	Allocate only one print position for each DBCS character.

S2NCNUM The column number of the next print position after the transparent data. If this field is zero, NetSpool determines the column number.

S2NLNUM The line number of the next print position after the transparent data. If this field is zero, NetSpool determines the line number.

S2WRKLEN The length of the data in the work area, S2WRKARA.

S2WRKARA The transparent-data control (TRN) and data to be added to the data stream. You can also add other SCS controls that NetSpool supports. Valid controls are described in *z/OS Infoprint Server User's Guide*.

Examples

1. To add no data to the beginning of the data set, set return code 2 or 8 in register 15.
2. To add data to the beginning of the data set, do the following:
   - Put the data in the S2WRKARA field.
   - Put the length of the data in S2WRKARA into the S2WRKLEN field.
   - Set return code 1 or 9 in register 15.

Beginning of File Exit for 3270 Data Streams (APIUBF3)

The Beginning of File exit for 3270 data streams (APIUBF3) receives control when NetSpool starts a new output data set for 3270 input data streams. You can use this exit to add up to 275 bytes of data to the start of some or all data sets.

NetSpool calls the same exit for all LU type 0 and LU type 3 sessions. However, your exit can perform different functions based on the following input fields in the APIPP1X2 control block:

- S2FLAGS: Type of output data stream NetSpool creates for this printer— a line or PCL data stream
- S2LPTNAM: NetSpool printer name, also known as the secondary LU (SLU) name
- S2PLUNAM: LU name of the VTAM application, also known as the primary LU (PLU) name
- S2PRTCLS: LU classes to which the NetSpool printer belongs

You can use this exit to add printable text and PCL commands. Follow these guidelines when you add data:

- If NetSpool converts the input data stream to a line data stream:
  - If the printer accepts EBCDIC line data (such as an IBM AFP printer controlled by PSF for OS/390), you can add text. Code the text data in EBCDIC representation, and precede the data with a carriage control character.
  - If the printer is controlled by IP PrintWay, you can add text and any PCL commands that your printer supports. Code text data in ASCII representation, and precede the text and PCL commands with a X'35' character and a 2–byte length field. IP PrintWay removes the X'35' and length field before transmitting the text and PCL commands, without change, to the printer.
- If NetSpool converts the input data stream to a PCL data stream:
  - You can add text and any PCL commands that your printer supports. Code text data in ASCII representation. NetSpool and IP PrintWay transmit the text and PCL commands to the printer without change.
  - If you add text, be sure to add a line termination control, such as a Carriage Return/New Line control sequence or a Form Feed control, after the text data to ensure that subsequent text starts either on a new line or page.
- Any PCL commands that you add in this exit override PCL commands specified in the IP PrintWay **Document header** field in the printer definition. They also override any PCL commands that NetSpool automatically generates as a result of PCL conversion values specified in the printer definition, such as print density or line density values.

Input to APIUBF3

At input to the Beginning of File exit (APIPPTD1), the contents of the registers are:

Register 1 Address of the APIPP1X2 control block

Register 13 Address of an 18-word save area for saving the caller's registers

Register 14 Return address

The APIPP1X2 control block is described in Figure 27 on page 152. The following fields are valid at entry:

S2FLAGS Type of conversion and VTAM session:

- **S2LU1** Set for VTAM LU1 session; not set for VTAM LU0 or LU3 sessions.
- **S2PCL** Set when NetSpool will convert the input data stream to a PCL data stream.

S2PLUNAM LU name of the application generating the print request — primary LU (PLU)

S2LPTNAM Name of the printer LU that NetSpool is processing — secondary LU (SLU)

S2PRTCLS Logical unit classes to which the printer LU belongs. Each bit in this field represents a class, with the high order-bit of the leftmost byte representing class 1. See Figure 27 on page 152 for a description of the bits.

Output From APIUBF3

At return from the Beginning of File exit (APIUBF3), restore the contents of all registers, and place one of the following return codes in register 15 to tell NetSpool what action to take:

Code	Meaning
1	Add the data from S2WRKARA to the output data set.
2	Do *not* add data from S2WRKARA to the output data set.

Also, set the following fields in the APIPP1X2 control block:

S2WRKLEN Length of work area S2WRKARA.

S2WRKARA Data and PCL commands to be added to the output data stream.

Examples

1. To add no data to the beginning of the data set, set return code 2 in register 15.
2. To add data to the beginning of the data set, do the following:
   - Put the data in the S2WRKARA field.
   - Put the length of the data in S2WRKARA into the S2WRKLEN field.
   - Set return code 1 in register 15.

Transparent Data Control Exit (APIPPTD2)

NetSpool calls the Transparent Data Control exit (APIPPTD2) whenever NetSpool encounters the Transparent Data control (TRN) in an SCS input data stream. You can use this exit to inspect the transparent data and leave it unchanged, modify it, or delete it.

NetSpool calls the same exit for all VTAM sessions. However, your exit can perform different functions based on the following input fields in the APIPP1X2 control block:

- S2FLAGS: Type of output data stream NetSpool creates for this printer— a line or PCL data stream
- S2LPTNAM: NetSpool printer name, also known as the secondary LU (SLU) name
- S2PLUNAM: LU name of the VTAM application, also known as the primary LU (PLU) name
- S2PRTCLS: LU classes to which the NetSpool printer belongs

When you modify the transparent data, you can also add other SCS controls in addition to the TRN control. The SCS controls supported by NetSpool are described in *z/OS Infoprint Server User's Guide*. The total number of bytes of modified data that you can include is 275. NetSpool does not call the Transparent Data Control exit (APIPPTD2) again for any transparent data included by this exit.

In order to format lines and pages, NetSpool keeps track of the next output print position, that is, the column number and line number at which the next character is to print. For transparent data, NetSpool increases the column and line number by the length of the transparent data, as specified in the TRN control.

The APIPPTD2 exit lets you change the column and line number of the next print position *after* the transparent data. To assist in calculating the desired line and column numbers, NetSpool provides the exit with the column and line number of the next print position, *before* the transparent data. NetSpool also provides the exit with other page-formatting values.

Input to APIPPTD2

At input to the Transparent Data Control exit (APIPPTD2), the contents of the registers are:

Register 1 Address of the APIPP1X2 control block

Register 13 Address of an 18-word save area for saving the caller's registers

Register 14 Return address

The APIPP1X2 control block is described in Figure 27 on page 152. The following fields are valid at entry:

S2FLAGS Type of conversion and VTAM session:

- **S2LU1** Set for VTAM LU1 session; not set for VTAM LU0 or LU3 sessions.
- **S2PCL** Set when NetSpool will convert the input data stream to a PCL data stream.

S2COL The column number of the next print position

S2LM Left margin column number

S2RM Right margin column number

S2MPP Maximum presentation position (line length)

S2LINE The line number of the next print position

S2TM Top margin line number (channel 1)

S2C2T12 Channel 2-12 line numbers

S2BM Bottom margin line number

S2MPL Maximum page length

S2PLUNAM LU name of the application generating the print request (PLU)

S2LPTNAM Name of the printer LU that NetSpool is processing (SLU)

S2PRTCLS Logical unit classes to which the printer LU belongs. Each bit in this field represents a class, with the high-order bit of the leftmost byte representing class 1. See Figure 27 on page 152 for a description of the bits.

S2TPDATA Transparent data found in the input data stream, containing the TRN control and transparent data

Output from APIPPTD2

At return from the Transparent Data Control exit (APIPPTD2), restore the contents of registers 1 through 14, and place one of the following codes in register 15 to tell NetSpool what action to take:

Code	Meaning
1	Modify the transparent data; S2WRKARA contains the modified data.
2	Do *not* modify the transparent data.
4	Close the VTAM session for this printer LU.
8	Do *not* call exit APIPPTD2 again for any additional transparent data found in this data set. Do *not* modify the transparent data.
9	Do *not* call exit APIPPTD2 again for any additional transparent data found in this data set. Modify the transparent data; S2WRKARA contains the modified data.

Also, set the following fields in the APIPP1X2 control block.

S2NCNUM The column number of the next print position after the transparent data. If this field is zero, NetSpool determines the column number.

S2NLNUM The line number of the next print position after the transparent data. If this field is zero, NetSpool determines the line number.

S2WRKLEN The length of the data in the work area, S2WRKARA. Set this field to zero to delete the transparent data.

S2WRKARA The modified TRN control and transparent data. You can also include other SCS controls that NetSpool supports. Valid controls are described in *z/OS Infoprint Server User's Guide*.

Note: Do *not* change the TRN control or data in the S2TPDATA field.

Examples

1. To include the transparent data passed to this exit, unmodified, in the output data stream, set code 2 or 8 in register 15.
2. To modify the transparent data in the output data stream, do the following:
   - Put the TRN control and the modified transparent data in the S2WRKARA field.
   - Put the length of the TRN control and transparent data into S2WRKARA in the S2WRKLEN field.
   - Set code 1 or 9 in register 15.
3. To delete the transparent data passed to this exit from the output data stream, do the following:
   - Set the S2WRKLEN field to zero, to indicate that no data is in S2WRKARA.
   - Set code 1 or 9 in register 15.

Graphic Escape Exit (APIUGEX)

The Graphic Escape exit (APIUGEX) for both SCS and 3270 data streams receives control when NetSpool finds a Graphic Escape (GE) character sequence in the input data stream. A GE character sequence begins with a one-byte code, X'08', followed by the GE printable character. You can use this exit to replace the Graphic Escape character sequence with another character or sequence of characters.

If you do not modify the default Graphic Escape exit, NetSpool simply removes the GE code (X'08') and prints the next character in the font used to print the rest of the data set. For example, for GE character sequence X'08C1', letter A prints. A GE code typically means, however, that the printer should print the next character in an alternate font. Therefore, if any of your VTAM applications generate GE characters,

you might want to write an exit that replaces GE characters with a different character in the same font or the same character in the printer's alternate font. The sample Graphic Escape exit illustrates how to how to replace GE characters with the same character in the alternate font.

NetSpool calls the same Graphic Escape exit for all VTAM sessions. However, your exit can perform different functions based on the following input fields in the APIPP1X2 control block:

- S2FLAGS: Type of output data stream NetSpool creates for this printer — a line or PCL data stream
- S2LPTNAM: NetSpool printer name, also known as the secondary LU (SLU) name
- S2PLUNAM: LU name of the VTAM application, also known as the primary LU (PLU) name
- S2PRTCLS: LU classes to which the NetSpool printer belongs

When you replace a GE character sequence, follow these guidelines:

- When NetSpool creates a PCL data stream, code the replacement characters in ASCII representation.
- When NetSpool creates a line data stream, code the replacement characters in EBCDIC representation.
- Do *not* include the X'08' GE code in the replacement character sequence.

Input to APIUGEX

At input to the Graphic Escape exit (APIUGEX), the contents of the registers are:

Register 1 Address of the APIPP1X2 control block

Register 13 Address of an 18-word save area for saving the caller's registers

Register 14 Return address

The APIPP1X2 control block is described in Figure 27 on page 152. The following fields are valid at entry:

S2FLAGS Type of conversion and VTAM session:

- **S2LU1** Set for VTAM LU1 session; not set for VTAM LU0 or LU3 sessions.
- **S2PCL** Set when NetSpool will convert the input data stream to a PCL data stream.

S2PLUNAM LU name of the application generating the print request — primary LU (PLU)

S2LPTNAM Name of the printer LU that NetSpool is processing — secondary LU (SLU)

S2PRTCLS Logical unit classes to which the printer LU belongs. Each bit in this field represents a class, with the high order-bit of the leftmost byte representing class 1. See Figure 27 on page 152 for a description of the bits.

S2TPDATA First byte: The Graphic Escape (GE) character

found in the input data stream; that is, the character after GE code X'08'. This character is usually in EBCDIC representation.

Second byte: When NetSpool creates a PCL data stream, this byte contains the equivalent character from the printer's code page. This character is usually in ASCII representation. When NetSpool creates a line data stream, this byte contains X'00'.

Output From APIUGEX

At return from the Graphic Escape exit (APIUGEX), restore the contents of all registers, and place one of the following return codes in register 15 to tell NetSpool what action to take:

Code	Meaning
1	Add the replacement data in S2WRKARA to the output data set.
2	Do *not* add data in S2WRKARA to the output data set.

Also, set the following fields in the APIPP1X2 control block:

S2WRKLEN Length of work area S2WRKARA.

S2WRKARA Data to replace the Graphic Escape character sequence.

Examples

1. To keep the Graphic Escape character, set return code 2 in register 15.
2. To replace the Graphic Escape character, do the following:
   - Put the replacement data in the S2WRKARA field.
   - Put the length of the data in S2WRKARA into the S2WRKLEN field.
   - Set return code 1 in register 15.

Sample Exits and Macro

NetSpool provides sample exits and a macro, written in assembler language. The sample exits are in SYS1.SAMPLIB; the macro is in SYS1.MACLIB:

- APIJPTD1, a sample Beginning of File exit for SCS data streams
- APIJPTD2, a sample Transparent Data Control exit
- APIJUBF3, a sample Beginning of File exit for 3270 data streams
- APIJUGEX, a sample Graphic Escape exit
- APIPP1X2, a macro that contains parameter area S2STG and is used in assembling the exits.

You can either view the samples online or print them. You can then modify, assemble, and install the exit programs to provide your own versions of the exits. When you link-edit the samples, change the names to:

- APIPPTD1 for the Beginning of File exit for SCS data streams
- APIPPTD2 for the Transparent Data Control exit
- APIUBF3 for the Beginning of File exit for 3270 data streams
- APIUGEX for the Graphic Escape exit

Object code for the sample exits is not provided.

Installing the Exits

To install the exits, follow these steps to compile and link-edit the exits:

- Ensure that the name of the file is correct. See "Sample Exits and Macro" for the required names.

- Specify the RENT link-edit attribute for improved performance; however, if you specify the RENT attribute, you must stop and restart NetSpool to pick up changes to the exit.
- Specify the AC(1) link-edit attribute and put the exit in an APF-authorized library.

Identify the library containing the exits in either:

- A STEPLIB statement in the NetSpool startup procedure. To provide a different exit program for each instance of NetSpool, use a unique STEPLIB for each startup procedure.
- A library concatenated to LNKLST. If you place an exit in a LNKLST library, then you must use the same exit program for each instance of NetSpool.

Modifying the Exits

If you coded the exit to be reentrant and specified the RENT link-edit attribute, you must stop and restart NetSpool to pick up the new exit. If you did not specify the RENT link-edit attribute, you can replace the exit while NetSpool is running; however, IBM recommends that you code the exit to be reentrant for improved performance.

Chapter 8. Customizing IP PrintWay

This chapter describes how to customize the IP PrintWay component of Infoprint Server. See “IP PrintWay” on page 16 for an overview of IP PrintWay and how this component fits into your system.

You must customize IP PrintWay if you want to transmit output data sets from the JES spool to the following types of remote printers and e-mail destinations:

- TCP/IP-attached ASCII printers or print servers. The printers or print servers must support one of the following TCP/IP transmission protocols:
  - LPR to LPD
  - Direct socket printing
  - Internet Printing Protocol (IPP)
- VTAM-controlled printers. The printers must be defined to VTAM as one of the following types of logical units: LU0 (DSC/DSE), LU1 (SCS), or LU3 (DSC/DSE).
- Electronic mail (e-mail) destinations, that is, one or more e-mail addresses.

IP PrintWay uses a different transmission protocol to transmit data to each type of printer and to e-mail destinations. The administrator can select one of the following transmission protocols in the printer definition: LPR, direct sockets, IPP, VTAM, and e-mail.

This chapter describes the following customization tasks. Required tasks are required by all installations that want to use IP PrintWay. Optional tasks are required only if the listed condition applies to your installation.

Task	Condition	See Page:
Identifying Software Requirements	Required	165
Customizing z/OS Communications Server	Required	165, 51
Defining the IP PrintWay Functional Subsystem	Required	169
Updating the BPXPRMxx Member of SYS1.PARMLIB	Optional: To start multiple IP PrintWay FSAs	178
APF-Authorizing SYS1.IMAGELIB	Optional: To format data using FCBs	178
Allocating and Initializing the Transmission-Queue Data Set	Required	178
Restricting Access to the Transmission-Queue Data Set	Optional: To limit access to the data set	180
Allocating and Initializing the IP PrintWay Message-Log Data Set	Required	180
Customizing the IP PrintWay Startup Procedure	Required	181
Changing LE Run-time Options	Optional: To change default options	185
Creating a VTAM APPL Resource Definition	Optional: To print to VTAM-controlled printers	185
Creating FSS and FSA Definitions in the Printer Inventory	Optional: To print to VTAM-controlled printers or to change default configuration options	186

© Copyright IBM Corp. 1995, 2002

Task	Condition	See Page:
Writing IP PrintWay Exit programs	Optional: To customize IP PrintWay processing	189
Customizing PSF for OS/390 for Sharing Network Printers	Optional: To share printers with PSF for OS/390	236

If you have already customized IP PrintWay to use one transmission protocol, and you now want IP PrintWay to use another protocol (such as the e-mail protocol), you might need to perform additional customization tasks. The following table summarizes the customization tasks that are specific to each IP PrintWay transmission protocol. Required tasks are required by all installations that want to use the transmission protocol; optional tasks apply only to some installations.

	Protocol					
Task	LPR	Direct sockets	IPP	VTAM	E-mail	See Page:
Install additional software.	No	No	No	Required	Required	165
Customize sendmail.	No	No	No	No	Required	167
Specify environment variables.	No	No	Optional	No	Optional	181
Create a VTAM APPL definition, and specify the APPL name in an FSS definition.	No	No	No	Required	No	185

The following table lists other components of Infoprint Server that you might need to customize:

Task	Condition	See Page:
Customizing the Printer Inventory Manager	Required	25
Customizing Print Interface and the LPD	Optional: To transform data from one format to another using Infoprint Server Transforms.[1] [2]	51 and 61
Customizing Infoprint Server Transforms	Optional: To transform data from one format to another using Infoprint Server Transforms.[1]	83
Enabling Infoprint Server Transforms	Optional: To print to VTAM-controlled printers using the Coax Printer Support feature of Infoprint Server Transforms.	86

1. This task is not required to use the Coax Printer Support feature of Infoprint Server Transforms.
2. Whenever you add a new class to the classes in which Print Interface can allocate data sets, be sure to specify no blank truncation for that class as described in “Customizing JES to Specify No Blank Truncation” on page 53.

Related tasks: After you customize IP PrintWay, the administrator and operator must perform the following tasks. Refer to *z/OS Infoprint Server Operation and Administration* for information about these tasks.

- The administrator must create a printer definition in the Printer Inventory for each target printer and specify attributes required by IP PrintWay. If you want to transform data, also specify attributes required by the transforms. If users can specify the IP addresses of the target printers on JCL or in job attributes, you do not need to create a printer definition for each printer.
- The administrator must ensure that VTAM-controlled printers are defined to VTAM.
- The operator must start IP PrintWay.

Identifying Software Requirements

The following additional IBM software beyond the z/OS base elements is required:

- To print to VTAM-controlled printers:
  - The Coax Printer Support feature of the Infoprint Server Transforms product, with Infoprint Server Transforms PTF UW74922
  - To print binary data (such as PCL data) unchanged or to specify an end-of-file control:
    - Infoprint Server PTFs UW82899 and UW82940
    - Infoprint Server Transforms PTF UW82927
  - To use the delete form feed function:
    - Infoprint Server PTF UW85178
    - Infoprint Server Transforms PTF UW85179
- To send output to e-mail destinations: Infoprint Server PTFs UW85325 and UW85327.
- To obtain the user ID of the job submitter in IP PrintWay exits when you use the resubmit for filtering function: Infoprint Server PTF UW87646.
- To use the lower TCP/IP connection timeout value of 30 seconds, Infoprint Server PTF UW88108.
- To print to IPP-enabled printers, Java is no longer required.

Customizing z/OS Communications Server

To print to TCP/IP-attached printers or to send output to e-mail destinations, you must customize the TCP/IP component of z/OS Communications Server. For information about how to customize TCP/IP, refer to *z/OS Communications Server: IP Configuration Guide* and *z/OS Communications Server: IP Configuration Reference*.

The following table describes tasks related to TCP/IP customization. Required tasks are required by all installations that customize TCP/IP. Optional tasks are required only if the listed condition applies to your installation.

Task	Condition	See Page:
Customizing the TCPIP.DATA Data Set	Optional: To specify the TCP/IP name or high-level qualifier	166
Customizing the PROFILE.TCPIP Data Set	Optional: To change TCP/IP configuration	167

Task	Condition	See Page:
Customizing z/OS UNIX Sendmail	Optional: To send output to e-mail destinations	167

Customizing the TCPIP.DATA Data Set

The TCPIP.DATA data set contains the following statements that are used by IP PrintWay:

Statement **Meaning**

TCPIPJOBNAME

Specifies the name of the TCP/IP program in your installation. If you have several TCP/IP programs installed with different names, you can specify the name of the TCP/IP program that IP PrintWay is to use in this statement or in the IP PrintWay FSS definition in the Printer Inventory; see "Creating FSS and FSA Definitions in the Printer Inventory" on page 186 for information about FSS definitions. If you specify a TCP/IP name in both places, 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, the name of the TCP/IP program must be TCPIP, which is the TCP/IP default name.

DATASETPREFIX

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

If you create a unique TCPIP.DATA data set for use by IP PrintWay, either (1) identify that data set in the IP PrintWay startup procedure with a SYSTCPD DD statement, or (2) use the jobname of the IP PrintWay startup procedure as the high-level qualifier of the TCPIP.DATA data set name.

To find the TCPIP.DATA data set to use, IP PrintWay follows the following search sequence. IP PrintWay reads each instance of TCPIP.DATA it can find, in order, until it finds values for each TCP/IP configuration statement (TCPIPDATA and DATASETPREFIX). It uses the first value it finds for each statement. If an allocation fails, the data set does not exist, or the data set is not available, IP PrintWay searches the next data set in the sequence.

1. The data set referenced by the SYSTCPD DD statement in the IP PrintWay startup procedure
2. A data set named *jobname*.TCPIP.DATA, where *jobname* is the jobname of the IP PrintWay startup procedure
3. A data set named SYS1.TCPPARMS(TCPDATA)
4. A data set named *tcphlq*.TCPIP.DATA, where *tcphlq* is the value in a DATASETPREFIX statement found earlier in the search sequence or the system default value of `TCPIP`

For information about the TCPIP.DATA data set, refer to *z/OS Communications Server: IP Configuration Reference*

Customizing the PROFILE.TCPIP Data Set

When you customize TCP/IP, you can customize the PROFILE.TCPIP data set. See "Customizing z/OS Communications Server" on page 51 for information about how to customize this data set for IP PrintWay and also for other components of Infoprint Server.

Customizing z/OS UNIX Sendmail

When the e-mail protocol is selected in a printer definition, IP PrintWay uses z/OS UNIX sendmail to prepare and send e-mails to the recipients listed in the printer definition. Sendmail is a mail transfer agent provided with z/OS Communications Server that provides enhanced SMTP support. Sendmail version 8.8.7 runs on z/OS V1R2.

If you have not already customized z/OS UNIX sendmail, you must do so before IP PrintWay can send output to e-mail destinations. For information about how to customize sendmail, refer to the following publications and Web site:

- *z/OS Communications Server: IP Configuration Guide* describes how to customize sendmail and the other components of z/OS Communications Server that sendmail requires.
- The industry-accepted publication *Sendmail*, published by O'Reilly & Associates, Inc. (ISBN 1–56592–222–0), describes the sendmail configuration file (**sendmail.cf**), how to set up sendmail aliases files, how to ensure security, and how to debug problems with your sendmail configuration.
- http://www.sendmail.org, a Web site maintained by the Sendmail Consortium as a resource for the freeware version of sendmail.

Guidelines: Follow these guidelines when you customize sendmail:

- Sendmail is highly dependent on the Domain Name Server (DNS); therefore, be sure to set up this component correctly.
- Customize the **Timeout** option in the sendmail configuration file to suit your installation. The Timeout option specifies how long sendmail waits before sending a warning that e-mail could not be delivered to a remote system and how long sendmail waits before bouncing e-mail. Bounced e-mail is undeliverable e-mail.
- Start sendmail as a daemon so that sendmail can receive messages. The sendmail **-bd** option starts sendmail as a daemon, and the **-q** option specifies how often sendmail is to look in its queue to process pending mail.

  Example: The following commands switch to an effective UID of 0 and start sendmail as a daemon that checks its mail queue every minute:

  ```
  su
  /usr/sbin/sendmail  -bd  -q1m
  ```

  To use the **su** command, you must be permitted to the BPX.SUPERUSER profile in the FACILITY class within RACF.

  If you want sendmail to start automatically when the z/OS system is initialized, specify the **sendmail** command in the **/etc/rc** file.
- In the sendmail aliases file, you can create alias names to represent a list of real e-mail addresses. When you create an alias, you should specify the user ID that owns the list in the **owner-***alias* statement. The user ID who owns the list receives notification about bounced e-mails. If you do not specify the user ID that owns the alias name, then sendmail sends notification of bounced e-mails to the user ID assigned to the IP PrintWay startup procedure.

- Configure the z/OS system so that a firewall does not prevent users from sending e-mails. Also if you want to receive replies from e-mails or sendmail notifications about undeliverable e-mails, configure your z/OS system so that it can receive e-mails from outside the firewall.
- If you do not install sendmail in the default directory, **/usr/sbin**, specify the full path name of the sendmail command in the AOPMAILER environment variable in the IP PrintWay startup procedure; see "Editing the ANFWPROC Startup Procedure" on page 182 for information.

Tip: Verify that you have configured sendmail correctly before the administrator selects the e-mail protocol in a printer definition. To verify that sendmail is configured correctly, run the **sendmail** command directly from the z/OS UNIX command line. The **sendmail** command is described in *z/OS Communications Server: IP User's Guide and Commands*. Following are some tests you can perform:

- **Send a file to a local user:** Run the following command to send file `myfile` to yourself:

  ```
  /usr/sbin/sendmail youruserID <myfile
  ```

  Result: The e-mail should be delivered to you and you should receive no error messages from sendmail. To view the mail sent to your user ID, run the z/OS UNIX **mail** command:

  ```
  mail
  ```

- **Send a file to a remote address:** To verify that firewalls do not prevent sending e-mails, send a file to a remote user on a domain that is not within your installation:

  ```
  /usr/sbin/sendmail username@domain <myfile
  ```

  Result: The e-mail should be delivered to the remote address and you should receive no error messages from sendmail.

- **Send a file to an alias name:** To verify that the sendmail aliases file is configured correctly, send a file to a sendmail alias name:

  1. To edit the aliases file, use the **su** command to switch to an effective UID of 0 and then use your favorite editor:

     ```
     su
     oedit /etc/aliases
     ```

     To use the **su** command, you must be permitted to the BPX.SUPERUSER profile in the FACILITY class within RACF.
  2. To create alias `testlist`, type the following lines in the sendmail aliases file, **/etc/aliases**:

     ```
     testlist: youruserID
     owner-testlist: youruserID
     ```
  3. Run the sendmail **newaliases** command to update the sendmail aliases file:

     ```
     /usr/sbin/newaliases
     ```
  4. Run the **sendmail** command to send file `myfile` to alias name `testlist`:

     ```
     /usr/sbin/sendmail testlist <myfile
     ```

  Result: The e-mail should be delivered to you and you should receive no error messages from sendmail.

Defining the IP PrintWay Functional Subsystem

IP PrintWay operates as a JES functional subsystem (FSS). A functional subsystem is an extension of JES that runs in its own address space. For more information about JES functional subsystems, refer to one of the following publications:

- *z/OS JES2 Initialization and Tuning Guide*
- *z/OS JES3 Initialization and Tuning Guide*

Within the IP PrintWay FSS, you define one or more IP PrintWay functional subsystem applications (FSAs), which use the support facilities of the FSS to communicate with JES. Several IP PrintWay FSAs can run in the same FSS. The following sections describe planning decisions to consider before coding the JES2 or JES3 initialization statements and writing a startup procedure.

Determining How Many Functional Subsystems to Define

You do not need to create more than one IP PrintWay FSS; however; for improved throughput and more efficient use of system resources, you can define up to 2000 FSSs.

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. Note that when multiple IP PrintWay FSSs share a transmission-queue data set, you must also 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. See “Allocating and Initializing the Transmission-Queue Data Set” on page 178 for a sample job for allocating the transmission-queue data set.

Determining How Many Functional Subsystem Applications to Define

You do not need to define more than one or two IP PrintWay FSAs because one PrintWay FSA can transmit data sets to many printers or e-mail destinations. However, for improved throughput and more efficient use of system resources, you might want to define several FSAs per FSS. JES allows you to define up to 64 FSAs per FSS.

Recommendations:

- Define at least two IP PrintWay FSAs so that when one FSA is waiting for a printer to respond, the second IP PrintWay FSA can print another document on a different printer.
- Do not define more than 35 FSAs per FSS.
- Determine how many jobs that need to print concurrently (the burst rate), and specify at least that many FSAs. For example, some installations with low print volume can drive as many as 600 printers with 5 or 6 FSAs.

Related task: If you create more than one IP PrintWay FSA, you might need to increase the number of active z/OS UNIX processes allowed on your system. See “Updating the BPXPRMxx Member of SYS1.PARMLIB” on page 178 for information.

Determining Work-Selection Criteria

You define work-selection criteria for each IP PrintWay FSA during JES initialization. These criteria can determine which output data sets each FSA selects from the JES spool. You specify the work-selection criteria on the WS parameter of either the JES2 PRT*nnnnn* statement or the JES3 DEVICE 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: output class, form name, and destination name. Some considerations related to work-selection criteria are:

- 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 JCL parameter. The reason for this is that JES does *not* use the host name or IP address specified as an OUTPUT JCL parameter when determining whether an output data set meets the work-selection criteria defined for an FSA.
- In a JES3 environment, specify form as a work-selection criterion if your installation wants IP PrintWay to use the form name specified in a JCL statement to select any printer definitions in the Printer Inventory. The reason for this is that JES3 passes the form name specified as a JCL parameter to the IP PrintWay FSA only if form is a JES work-selection criterion. Although you specify form as a work-selection criterion, the IP PrintWay FSA selects jobs with all form names for processing.
- In a JES3 environment, specify FCB as a work-selection criterion if you want IP PrintWay to perform FCB processing. The reason for this is that JES3 passes the FCB name specified as a JCL parameter to the PrintWay FSA only when FCB is a JES work-selection criterion.

Specifying a Default Font and Page Definition for Data Transforms

When you define an IP PrintWay FSA to JES, you can specify a default UCS and a default FCB. JES supplies the default UCS and FCB to IP PrintWay when an output data set does not specify the UCS or FCB parameter on the OUTPUT JCL statement. IP PrintWay uses the default UCS name as the default font name if the output data set does not specify the CHARS parameter on the OUTPUT JCL statement. IP PrintWay uses the default FCB name as the default page definition name (in addition to the FCB name) if the output data set does not specify the PAGEDEF parameter.

You can specify a default font and page definition to JES in the following ways:

- JES2:
  - Specify a default font in (1) the NIUCS parameter on the JES2 PRINTDEF statement or (2) the UCS parameter on the JES2 PRT*nnnnn* statement for the IP PrintWay FSA. If you don't specify a default font, JES2 uses GF10 as the default font.
  - Specify a default page definition in (1) the NIFCB parameter on the JES2 PRINTDEF statement or (2) the FCB parameter on the JES2 PRT*nnnnn* statement for the IP PrintWay FSA.
- JES3:
  - Specify a default font in (1) the CHARS parameter on the JES3 OUTSERV statement or (2) the CHARS parameter on the JES3 DEVICE statement for the IP PrintWay FSA. If you do not specify a default font, JES3 uses font GS10 as the default font.
  - Specify a default page definition in (1) the CARRIAGE parameter on the JES3 OUTSERV statement or (2) the CARRIAGE parameter on the JES3 DEVICE statement for the IP PrintWay FSA. If you don't specify a default page definition, JES3 uses 6 as the default FCB name. (IP PrintWay adds a P1 prefix, so the default page definition is P16.)

When you select the **Resubmit for filtering** option in a printer definition, IP PrintWay passes the default font and page definition received from JES to the AFP

to PCL, AFP to PDF, and AFP to PostScript transforms. This means that, when you specify a default font and page definition to JES, the transforms do *not* use the font and page definition the administrator specifies in the printer definition or the font and page definition you specify in the transform configuration file.

If you want the transforms to use a font and page definition specified in a printer definition or in the transform configuration file, then do the following:

- JES2:
  - To prevent JES2 from supplying a default page definition name, (1) do *not* specify the NIFCB parameter in the PRINTDEF statement and (2) do *not* specify the FCB parameter in the PRT*nnnnn* statement.
  - To prevent JES2 from supplying a default font, specify `UCS=0` in the PRT*nnnnn* statement. When UCS=0, JES2 ignores any font specified in the NIUCS parameter.
- JES3:
  - To prevent JES3 from supplying a default page definition name, specify `PDEFAULT=FCB` on the DEVICE statement.
  - To prevent JES3 from supplying a default font specify `PDEFAULT=CHARS` on the DEVICE statement.
  - To prevent JES3 from supplying a default font or FCB, specify `PDEFAULT=(CHARS,FCB)` on the DEVICE statement.

See “PRTnnnnn Statement” on page 172 and “DEVICE Statement” on page 176 for more information about these statements.

See “Hierarchy for Selecting a Page Definition” on page 119 for information about how the transforms select a page definition; see “Hierarchy for Selecting a Font” on page 120 for information about how the transforms select a font.

Defining the IP PrintWay Functional Subsystem in JES2

To define a functional subsystem for IP PrintWay in a JES2 environment, code these JES2 initialization statements:

- One FSS(*fss_name*) statement to define each IP PrintWay functional subsystem (FSS)
- One PRT*nnnnn* statement to define each functional subsystem application (FSA) under control of the FSS

Also, these JES2 initialization statements have considerations for IP PrintWay:

- **SPOOLDEF:** Specify TRKCELL=5 on the JES2 SPOOLDEF statement for improved performance. Also, specify TRKCELL=YES on the PRT*nnnnn* statement.
- **OUTCLASS:** Specify BLNKTRNC=NO on the JES2 OUTCLASS statements for all output classes that IP PrintWay processes. This prevents JES from removing EBCDIC blanks at the end of each record on the JES spool, which JES does by default to save spool space. Some data streams, for example PCL, PostScript, and PDF data streams might not print properly if JES removes blanks. The JES default is BLNKTRNC=YES.
- **PRINTDEF:** The JES2 PRINTDEF statement lets you specify a default FCB and page definition in the NIFCB parameter and a default font in the NIUCS parameter. See “Specifying a Default Font and Page Definition for Data Transforms” on page 170 for more information.

Sample JES2 Initialization Statements

Figure 28 shows sample JES2 statements for one FSS named PRINTWAY and one associated FSA named PRT123.

```
FSS(PRINTWAY)   PROC=ANFWPROC,AUTOSTOP=YES
PRT123          CLASS=E,FSS=PRINTWAY,MODE=FSS,PRESELCT=YES,
                START=NO,TRKCELL=YES,UCS=0,WS=(Q)
```

Figure 28. Sample JES2 Initialization Statements. These statements are commonly found in the initialization-deck member of SYS1.PARMLIB.

The following section describes the JES2 initialization statements shown in Figure 28, including some additional parameters that you might want to specify for an IP PrintWay FSS and FSA.

Note: These sample statements do not include all possible JES parameters. For a complete list of JES parameters and detailed descriptions of each of the parameters, refer to *z/OS JES2 Initialization and Tuning Reference*.

FSS(fss_name) Statement

The FSS(*fss_name*) initialization statement is optional but recommended. If you omit it, JES2 generates a default FSS when an FSA is started.

Each FSS must have a unique name, specified as a one- to eight-character name. In the example, the FSS name is PRINTWAY. This FSS name must match the FSS name in the PRT*nnnnn* statement for each associated FSA.

FSS statement parameters are:

PROC=*procedure_name*
: Specifies the name of the procedure for starting this FSS. Different FSS statements can refer to the same startup procedure. See “Customizing the IP PrintWay Startup Procedure” on page 181 for more information. This parameter is required.

AUTOSTOP=YES | NO
: Specifies whether or not the FSS address space is to be stopped automatically if all FSAs under control of the FSS are stopped. Specifying AUTOSTOP=YES can conserve system resources. This parameter is optional; if you omit it, the default is NO.

HASPFSSM=name | 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=HASPFSSM.

PRTnnnnn Statement

A PRT*nnnnn* statement defines the FSA to JES. This section describes the PRT*nnnnn* parameters that you might want to specify for an IP PrintWay FSA. Refer to *z/OS JES2 Initialization and Tuning Reference* for a complete list of parameters.

Each FSA must have a unique name; therefore, *nnnnn* must be a unique number from 1 to 32767. JES2 lets you define more than one FSA with one PRT*nnnnn* statement. For more information about how to define a range of FSAs with one PRT*nnnnn* statement, refer to *z/OS JES2 Initialization and Tuning Reference*.

If you create an FSA definition in the Printer Inventory for this FSA, you must specify this FSA name in the FSA definition. Although JES might let you use different formats to specify the FSA name in this statement, for example, PRT(*nnnnn*), always specify the FSA name as PRT*nnnnn* in the FSA definition.

PRT*nnnnn* statement parameters include:

CLASS | QUEUE=(*classes*)
Specifies the output classes processed by this FSA. List all classes to be selected by this FSA; do not separate each class 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.

FCB=*fcb/pagedefinition*
Specifies the one- to four- character name of a default FCB and page definition. IP PrintWay uses this name as the FCB and page definition for output that does not specify the FCB and PAGEDEF parameters on the OUTPUT JCL statement. Do *not* code the P1 prefix of the page definition name; IP PrintWay adds the P1 prefix.

If you omit this parameter, JES2 determines the default FCB and page definition name from the NIFCB parameter of the JES2 PRINTDEF statement.

Note: Omit this parameter and also omit the NIFCB parameter if you want the AFP to PCL, AFP to PostScript, and AFP to PDF transforms to use a page definition specified in a printer definition or in the transform configuration file for output submitted to IP PrintWay with OUTPUT JCL statements. See “Specifying a Default Font and Page Definition for Data Transforms” on page 170 for more information.

FORMS=(*form_names*)
Specifies the one- to eight-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=*fss_name*
Specifies the name of the FSS associated with this FSA. This parameter is required.

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.

PRESELCT=YES | NO
Specifies whether output data sets are preselected for this FSA. This parameter is optional. If you omit it, the default is YES.

ROUTECDE=(*destinations*)
Specifies the destinations processed by this FSA. List from one to four different destination names that can be selected by this FSA; separate each destination name 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. If you omit it, 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.

UCS={*font* **| 0}**
Specifies the one- to four- character member name of a coded font in a font library, not including the two-character prefix. IP PrintWay uses this font name for output that does not specify the CHARS or UCS parameter on the OUTPUT JCL statement. IP PrintWay adds a X1 to the name.

UCS=0 indicates that JES2 will not supply a default font to IP PrintWay.

If you omit this parameter, JES2 determines the default font from the NIUCS parameter of the JES2 PRINTDEF statement or, if NIUCS is not specified, uses GF10 as the default.

Note: Specify UCS=0 if you want the AFP to PCL, AFP to PostScript, and AFP to PDF transforms to use the font specified in the printer definition or transform configuration file for jobs submitted to IP PrintWay with OUTPUT JCL statements. See "Specifying a Default Font and Page Definition for Data Transforms" on page 170 for more information.

WS=(*work_selection_criteria***)**
Specifies the work-selection criteria for this FSA; separate each value with a comma. Refer to *z/OS JES2 Initialization and Tuning Reference* for the valid values and defaults. For an IP PrintWay FSA, consider the following values:

CL | Q Specifies that the FSA selects only those data sets with the same class that are specified in the CLASS parameter of this statement.

R Specifies that the FSA selects only those data sets with the same destination that are specified in the ROUTECDE parameter of this statement. For considerations related to this work-selection criterion, see "Determining Work-Selection Criteria" on page 169.

F Specifies that the FSA selects only those data sets with the same form name that is 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: You might want to specify the SEP=NO parameter to prevent transmission of the JESNEWS data set.

Defining the IP PrintWay Functional Subsystem in JES3

To define a functional subsystem for IP PrintWay in a JES3 environment, code these JES3 initialization statements:

- One FSSDEF statement to define each IP PrintWay functional subsystem (FSS)
- One DEVICE statement for each functional subsystem application (FSA) under control of the FSS

Also, specify TRUNC=NO on the JES3 SYSOUT statements for all output classes that IP PrintWay processes. This prevents JES from removing EBCDIC blanks at the end of each record on the JES spool, which JES does by default to save spool space. ASCII data streams, for example PCL, PostScript, and PDF data streams

might not print properly if JES removes EBCDIC blanks. The JES default is the value set in the TRUNC parameter of the JES3 BUFFER statement or, if none is specified, TRUNC=YES.

Sample JES3 Initialization Statements

Figure 29 shows sample JES3 statements for one FSS named PRINTWAY and one associated FSA named PRT1.

```
FSSDEF,FSSNAME=PRINTWAY,PNAME=ANFWPROC,TYPE=WTR
DEVICE,DTYPE=PRTAFP1,FSSNAME=PRINTWAY,JNAME=PRT1,JUNIT=(,SYS1,UR,ON),
       MODE=FSS,PDEFAULT=(CHARS,FCB),WC=(E),WS=(CL,F)
```

Figure 29. Sample JES3 Initialization Statements. These statements are commonly found in the initialization-deck member of SYS1.PARMLIB. Continuation characters are not shown.

The following section describes the JES3 initialization statements shown in Figure 29, including some additional parameters that you might want to specify for an IP PrintWay FSS and FSA.

Note: These sample statements do not show all possible JES parameters. For a complete list of parameters and for detailed descriptions of each of the parameters, refer to *z/OS JES3 Initialization and Tuning Reference*.

FSSDEF Statement

The FSSDEF initialization statement is optional but recommended. If you omit it, JES3 generates a default FSS when an FSA is started.

FSSDEF statement parameters are:

FSSNAME=*fss_name*
: Specifies the name of this FSS. Each FSS must have a unique one- to eight-character name. This FSS name must match the FSS name in the DEVICE statement for each associated FSA. This parameter is required.

MSGDEST (*dest_class* **| JES)**
: Specifies the console destination class for messages concerning this FSS. This parameter is optional; if you omit it, JES sets a default.

PNAME=*procedure_name*
: Specifies the name of the procedure for starting this FSS. Different FSSDEF statements can refer to the same startup procedure. See “Customizing the IP PrintWay Startup Procedure” on page 181 for more information. This parameter is required.

SYSTEM=*system_name*
: Specifies the JES3 processor 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
: YES 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; if you omit it, the default is NO.

TYPE=WTR
: Specifies that the FSS is an output writer. This parameter is required.

DEVICE Statement

A DEVICE statement defines each FSA to JES. This section describes DEVICE parameters that you might want to specify for an IP PrintWay FSA. Refer to *z/OS JES3 Initialization and Tuning Reference* for a complete list of parameters.

DGROUP=*destination_name*

Specifies a one- to eight-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 this parameter only if you specify destination as a work-selection criteria on the WS parameter.

DTYPE=PRTAFP1

Specify PRTAFP1 as the device type for an IP PrintWay FSA. This parameter is required.

DYNAMIC=YES | NO

Specifies whether JES3 is to start and stop this FSA dynamically.

YES	Specifies that JES3 is to start this FSA whenever work is available for it. JES3 stops this FSA, and deactivates the address space when no work is available.
NO	Specifies that the operator is to start and stop this FSA. Specify NO to keep the address space active between transmission of data sets.

This parameter is optional. If you omit it, the default is NO.

FSSNAME=*fss_name*

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. If you omit it, the default is the name of this FSA specified in the JNAME parameter.

JNAME=*fsa_name*

Specifies the unique one- to eight-character name of this FSA. This parameter is required.

JUNIT= (*devnum,main,msgdest***,ON | OFF)**

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=FSS

Specifies that this FSA is managed by an FSS. This parameter is required.

PDEFAULT={NONE | CHARS | FCB | (CHARS,FCB)}

Specifies the JES3 default values that JES3 does not supply to IP PrintWay.

CHARS	Indicates that JES3 will not supply a default font to IP PrintWay.
FCB	Indicates that JES3 will not supply a default FCB and page definition name to IP PrintWay.

NONE Indicates that JES3 will supply a default font, FCB name, and page definition name to IP PrintWay.

This parameter is optional. The default is NONE.

Note: Specify `PDEFAULT=(CHARS,FCB)` if you want the AFP to PCL, AFP to PostScript, and AFP to PDF transforms to use the default font and page definition specified in the printer definition or the transform configuration file for jobs submitted to IP PrintWay with a JCL statement. See "Specifying a Default Font and Page Definition for Data Transforms" on page 170 for more information.

WC=(*classes***)**
Specifies the output classes processed by this FSA. List all classes to be selected by this FSA; separate each class 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=(*work_selection_criteria***)**
Specifies the work-selection criteria for this FSA; separate each value with a comma. Refer to *z/OS JES3 Initialization and Tuning Reference* for the valid values and the default. For an IP PrintWay FSA, consider the following values.

C Causes JES3 to pass the FCB name specified on the OUTPUT JCL statement to the FSA.

CL Specifies that the FSA selects only those data sets with the same class that are specified in the WC parameter of this statement.

D Specifies that the FSA selects only those data sets with the same destination as in the DGROUP parameter of this statement. For considerations related to this work-selection criterion, see "Determining Work-Selection Criteria" on page 169.

F Requests that JES3 pass the forms name specified on a JCL statement to the FSA.

If you want IP PrintWay to use the form name specified in a JCL parameter to select any printer definitions in the Printer Inventory, specify WS=(F) as one of the work-selection criteria. If you do not specify forms as a work-selection criterion, JES3 does not pass the form name specified in JCL to the IP PrintWay FSA.

If you want PrintWay to use the FCB to format output, specify WS=(C) as one of the work-selection criteria. If you do not specify FCB as a work-selection criterion, JES3 does not pass the FCB name specified in JCL to the PrintWay FSA.

Note: You might want to specify the BURST=NO parameter to prevent transmission of the JESNEWS data set.

Updating the BPXPRMxx Member of SYS1.PARMLIB

When you start an IP PrintWay FSA, a z/OS UNIX process is started. If you start multiple IP PrintWay FSAs, you might need to increase the number of active z/OS UNIX processes that the z/OS system allows. You can specify the number of allowed processes in the following parameters in the BPXPRMxx member of SYS1.PARMLIB:

Parameter **Meaning**

MAXPROCUSER
The maximum number of processes that a single z/OS UNIX user ID can have concurrently active.

Recommendation: Specify a value of at least 70.

MAXPROCSYS
The maximum number of processes that the z/OS system allows.

Recommendation: Add 70 to the current value for each IP PrintWay FSS that you create. For example, if you create 10 IP PrintWay FSSs, add 700 (70 * 10) to the current value. If you use Print Interface, also add another 200 to the MAXPROCSYS value.

Example: If you have 10 IP Printway FSSs, and also use Print Interface, specify the following statements in the BPXPRMxx member:

```
MAXPROCUSER(70)   /*70 (PrintWay)                                  */
MAXPROCSYS(1100)  /*700 (PrintWay)+200 (Print Interface)+200 (initial)*/
```

Tip: You can use the SETOMVS or SET OMVS command to dynamically increase the MAXPROCUSER and MAXPROCSYS values; however, to make a permanent change, edit the BPXPRMxx member that is used for IPLs. For more information about the BPXPRMxx member, refer to *z/OS MVS Initialization and Tuning Reference*.

APF-Authorizing 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 LNKLST statement.
- If the LNKAUTH parameter that is placed in the IEASYSxx member of SYS1.PARMLIB does not specify LNKLST, also add the data sets to the APF statement in the PROGxx member.

As an alternative to updating the PROGxx member, you can update the LNKLSTxx 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*.

Allocating and Initializing 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 creates the entries in the transmission-queue data set. You cannot add entries to it; however, you can use the Infoprint Server ISPF panels to browse, modify, and delete 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*.

You can allocate one transmission-queue data set to be shared by all IP PrintWay functional subsystems (FSSs). Or, you can create a separate data set for use by each IP PrintWay FSS. If more than one FSS can transmit output data sets to the same print queue, IBM recommends sharing the transmission-queue data set so that data sets are printed in the correct order.

The transmission-queue data set can be shared with other instances of IP PrintWay running at the *same* level on other systems; however, because the format of this data set changed in Infoprint Server for OS/390 V2R8, it *cannot* be shared with another instance of IP PrintWay running on at a *previous* level.

Note: When multiple FSSs share a transmission-queue data set, you must allocate the data set with SHAREOPTIONS (4 3).

The IP PrintWay transmission-queue data set is a VSAM key-sequenced data set. Use the IDCAMS program with the specifications shown in Figure 30 to allocate it. Refer to *z/OS DFSMS Access Method Services for Catalogs* if you need more information about allocating VSAM data sets.

You can define the IP PrintWay transmission-queue data set with any name. You might want to use the IP PrintWay default data set name **ANF.QUEUE**. Unless you define another default, this default appears on the IP PrintWay ISPF panels. This default is also used in the sample JCL provided by IP PrintWay.

Use the sample JCL in SYS1.SAMPLIB(ANFDEAL) to allocate the IP PrintWay transmission-queue data set. You can copy and modify this JCL for your installation.

```
DEFINE CLUSTER -
          (NAME(ANF.QUEUE) -
             VOLUMES(volser) -
             INDEXED -
             SPEED -
             SHAREOPTIONS(4 3)) -
           DATA -
             (NAME(ANF.QUEUE.DATA) -
             CYL(2 1) -
             KEYS(20 0) -
             RECORDSIZE(1292 2048) -
             FREESPACE(10,10) -
             CISZ(24576)) -
           INDEX -
             (NAME(ANF.QUEUE.INDEX))
```

Figure 30. IDCAMS Parameters to Allocate a Transmission-Queue Data Set — SYS1.SAMPLIB(ANFDEAL)

Notes:

1. Specify the correct data set name for your installation in the NAME parameter. Replace *volser* with the volume ID.
2. Use the values for the RECORDSIZE, SHAREOPTIONS, and KEYS options shown in Figure 30.

3. Adjust the value of the FREESPACE option based on experience with the data set. The size of the IP PrintWay transmission-queue data set changes dynamically, according to the number of data sets retained after successful or unsuccessful transmission.

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.

Restricting Access to the Transmission-Queue Data Set

All ISPF users can use the Infoprint Server ISPF panels to update the IP PrintWay transmission-queue data set, even if you have established a RACF administration group for the Printer Inventory. You can, however, use Resource Access Control Facility (RACF) or a similar program to restrict access to the data set.

If you have RACF 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-only 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.

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. In this way, the size of the message data set remains stable, with old messages 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 run the ANFMFILE program.

The message-log data set requires the following specifications:
- 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

Secondary space is unnecessary because the ANFMFILE program formats all available space. To change the size of the IP PrintWay message-log data set, rerun the ANFMFILE program.

Note: The ANFMFILE program produces a D37 abend when it completes processing.

You can suppress selected messages that IP PrintWay writes to the message-log data set by writing a Message exit. See “Message Exit (ANFUXMSG)” on page 227 for more information.

You can view messages in the message-log data set by using the Infoprint Server ISPF panels. Refer to *z/OS Infoprint Server Operation and Administration* for information.

Customizing the IP PrintWay Startup Procedure

To start an IP PrintWay FSS, you must use a cataloged startup procedure. The startup procedure identifies:

- The program name, region size, and time-out value
- Path of the Printer Inventory
- Location of the IP PrintWay transmission-queue and message-log data sets
- Environment variables

The following table summarizes the customization tasks related to setting up the startup procedure. Required tasks are required by all installations; optional tasks are required only if the listed condition applies to your installation.

Task	Condition	See Page:
Creating a User ID to Assign to the IP PrintWay Startup Procedure	Required	181
Defining the IP PrintWay Startup Procedure to RACF	Required	182
Editing the ANFWPROC Startup Procedure	Optional: To change default settings	182

Creating a User ID to Assign to the IP PrintWay Startup Procedure

The user ID assigned to the IP PrintWay startup procedure must be defined to RACF with access to z/OS UNIX.

Steps for creating a RACF user ID with access to z/OS UNIX:

1. Create a user ID with an OMVS segment, a home directory, and a default group that has an OMVS segment with a group identifier (GID).

   You can select any name for the user ID. Use the RACF ADDUSER command to create a new user ID; use the RACF ALTUSER command to alter an existing user ID to give it access to z/OS UNIX. For the default group, you can select any group that has an OMVS segment and a GID. The user ID must be connected to the default group.

   For example, the following RACF commands define user AOPSTC with default group AOPOPER. For *user-identifier*, specify an integer that is different from other UIDs in your installation. The NOPASSWORD operand defines the user ID as a *protected* user ID; protected user IDs cannot log on to the z/OS system, and they cannot be revoked due to incorrect password attempts.

   ```
   ADDUSER AOPSTC OMVS(UID(user-identifier) HOME('/u/aopstc')
      PROGRAM('/bin/sh')) DFLTGRP(AOPOPER) NOPASSWORD
   ```

2. Create the home directory specified in the ADDUSER command. The following z/OS UNIX commands switch to an effective UID of 0 and create a home directory for user ID AOPSTC with the required permissions:

   ```
   su 0
   cd /u
   mkdir -m 0700 aopstc
   chown -R AOPSTC /u/aopstc
   ```

   To use the **su** command, you must be permitted to the BPX.SUPERUSER profile in the FACILITY class within RACF.

3. Give the user ID RACF update access to the IP PrintWay transmission-queue and message-log data sets. Also, give the user ID RACF read access to the

TCP/IP TCPIP.DATA data set. The following RACF commands give user ID AOPSTC access to data sets ANF.QUEUE, ANF.MSGFILE, and TCPIP.TCPIP.DATA:

```
PERMIT ANF.QUEUE ACCESS(UPDATE) ID(AOPSTC)
PERMIT ANF.MSGFILE ACCESS(UPDATE) ID(AOPSTC)
PERMIT TCPIP.TCPIP.DATA ACCESS(READ) ID(AOPSTC)
```

Defining the IP PrintWay Startup Procedure to RACF

You must define the IP PrintWay startup procedure to the RACF STARTED profile in the FACILITY class. If you want to use the started procedure table (ICHRIN03) instead, refer to *z/OS Security Server RACF Security Administrator's Guide* for more information.

The following RACF commands define the ANFWPROC procedure to RACF. This example assumes that the user ID you defined for use with this procedure is AOPSTC and that AOPSTC is connected to group AOPOPER.

```
RDEFINE STARTED ANFWPROC.* STDATA(USER(AOPSTC) GROUP(AOPOPER))
SETROPTS RACLIST(STARTED) REFRESH
```

Editing the ANFWPROC Startup Procedure

IP PrintWay provides a startup procedure in SYS1.PROCLIB(ANFWPROC). You can use the same startup procedure to start more than one IP PrintWay FSS but only if you want all IP PrintWay FSSs to share the same IP PrintWay transmission-queue and message-log data sets. You must specify the name of this startup procedure in the JES initialization statement for each FSS.

Steps for Editing the IP PrintWay Startup Procedure

1. Copy the SYS1.PROCLIB(ANFWPROC) procedure to a procedure library that is known to JES. For information about procedure libraries, refer to *z/OS JES2 Initialization and Tuning Guide* and *z/OS JES3 Initialization and Tuning Guide*.
2. Edit the ANFWPROC procedure if any of the following conditions apply to your installation:
   - If the high-level qualifier for the TCP/IP data sets is not TCPIP, change the TCPHLQ parameter on the PROC statement.
   - If the Printer Inventory name is not the default name, AOP1, specify the inventory name in the PARM parameter of the EXEC statement.
   - If you run many FSAs in the IP PrintWay FSS, you might need to increase the size in the REGION parameter of the EXEC statement.
   - If you do not use the default names for the IP PrintWay transmission-queue and message-log data sets, change the data set names in the procedure.
   - If you use an IP PrintWay exit and the library that contains the exit is not in the standard z/OS search order, add a STEPLIB statement to identify the library.
   - If you did not add the Language Environment run-time library, CEE.SCEERUN, to the system LNKLST, add this data set to the STEPLIB DD statement.
   - If you plan to use the IPP protocol and did not add the C^{++} run-time library, CBC.SCLBDLL, to the system LNKLST, add this data set to the STEPLIB DD statement.
   - If you plan to use the IPP or e-mail protocol and did not install Infoprint Server or z/OS UNIX sendmail files in default locations, specify the directories in environment variables in the STDENV DD statement.

3. Make sure that the name of this procedure is specified in the JES initialization statements for all IP PrintWay FSSs that you want to use this procedure.
4. If IP PrintWay is running, 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 *z/OS Infoprint Server Operation and Administration* for information about how to stop and start IP PrintWay.

ANFWPROC Procedure

Figure 31 shows the ANFWPROC procedure provided in SYS1.PROCLIB. JES initialization statements for an FSS that uses this startup procedure are shown in Figure 28 on page 172 (JES2) and in Figure 29 on page 175 (JES3).

```
//ANFWPROC PROC  HLQ=ANF,
//               TCPHLQ=TCPIP
//* ----------------------------------------------------
//* this is a sample procedure for starting IP PrintWay
//* ----------------------------------------------------
//IEFPROC  EXEC  PGM=ANFFIEP,REGION=4M,TIME=NOLIMIT,PARM='INV=AOP1'
//*----------------------------------------------------
//* AOP1 is the default name of the Printer Inventory.
//* This must be modified if this is not the
//* name being used for the inventory.
//*----------------------------------------------------
//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 31. IP PrintWay Startup Procedure — SYS1.PROCLIB(ANFWPROC)

The JCL statements you can specify in the startup procedure are:

***//proc_name* PROC HLQ=ANF, TCPHLQ=TCPIP**

Names a cataloged procedure. *proc_name* must match the procedure name in a JES initialization statement for this FSS:

- PROC parameter of the JES2 FSS(*fss_name*) statement
- PNAME parameter of the JES3 FSSDEF statement

The HLQ keyword identifies the high-level qualifier for the IP PrintWay data sets; TCPHLQ identifies the high-level qualifier for the TCPIP.DATA data set. Change these values for your installation if necessary.

***//label* EXEC PGM=ANFFIEP,REGION=4M,TIME=NOLIMIT,PARM='INV=*inventory*'**

Invokes the IP PrintWay program.

PGM=ANFFIEP
The IP PrintWay program entry point.

REGION=4M The amount of below-the-line storage required by the FSS.

Note: A region size of 4 megabytes is suitable for many installations; however, if many FSAs run in the IP PrintWay FSS, you might need to specify a region size larger than 4M.

TIME=NOLIMIT
TIME=NOLIMIT prevents TIMEOUT abends.

PARM='INV=*inventory***'**
The name 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.

//ANFQUEUE DD DSN=&HLQ..QUEUE,DISP=SHR
The IP PrintWay transmission-queue data set. This statement is required.

Note: To ensure that data sets are always transmitted in the same order as IP PrintWay acquires them from the JES spool, specify the same transmission-queue data set for each IP PrintWay FSS.

//SYSTCPD DD DSN=&TCPHLQ..TCPIP.DATA,DISP=SHR
The TCPIP.DATA data set. This statement is optional. For a description of how IP PrintWay searches for the TCPIP.DATA data set, see "Customizing z/OS Communications Server" on page 165.

//ANFMMSG DD DSN=&HLQ..MSGFILE,DISP=SHR
The IP PrintWay message-log data set. This statement is required.

//STDENV DD DSN=*datasetname***,DISP=SHR**

//STDENV DD PATH='*UNIXfilename***',PATHOPTS=ORDONLY**
Specifies a data set or UNIX file that contains z/OS UNIX environment variables. Specify this DD statement only if IP PrintWay uses the Internet Printing Protocol (IPP) or e-mail transmission protocol and the default values for the environment variables are not suitable. The default values are suitable for installations that installed Infoprint Server and sendmail files in the default directories.

The following environment variables apply when IP PrintWay uses either the IPP or e-mail protocol:

Variable	Meaning
LANG	Specifies the language in which you want to receive messages. This variable is required if the NLSPATH variable contains %L and you do not want to receive messages in English. **Default:** None
LIBPATH	Specifies 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. **Default:** LIBPATH=/usr/lpp/Printsrv/lib:/lib:/usr/lib
NLSPATH	Specifes where Infoprint Server and z/OS UNIX message catalogs can be found. Specify the same catalogs as you specify in the z/OS UNIX NLSPATH environment variable in the **/etc/profile** file. **Default:** NLSPATH=/usr/lpp/Printsrv/%L/%N:/usr/lib/nls/msg/%L/%N

PATH Specifies 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.

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

The following environment variable applies when IP PrintWay uses the e-mail protocol:

AOPMAILER Specifies the full path name of the z/OS UNIX sendmail command. If you installed the sendmail executable file in the default directory, **/usr/sbin**, you do not need to specify this variable.

Default: AOPMAILER=/usr/sbin/sendmail

Specify environment variables in either a UNIX file or an MVS data set. The MVS data set must have variable blocked (VB) record format. Specify the environment variables in the format *variable*=*value*, with one environment variable per line or record. For example:

```
LIBPATH=/usr/mylib
```

See "Setting Environment Variables for the Printer Inventory Manager" on page 31 for a full description of these environment variables. Refer to *z/OS C/C++ Programming Guide* if you need more information about how to specify environment variables in the STDENV data set.

Changing LE Run-time Options

Figure 32 shows the run-time options for the LE environment that PrintWay establishes. If you want to change the LE run-time options, create a CEEUOPT object file as described in *z/OS Language Environment Programming Guide*. Then create an SMPE 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 32. LE Run-time Options for IP PrintWay

Creating a VTAM APPL Resource Definition

When IP PrintWay prints to VTAM-controlled printers, it runs as a VTAM application program. Therefore, if you plan to print to VTAM-controlled printers, 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.

Create the APPL statement in a new or existing application major node definition. Figure Figure 33 shows two sample APPL statements under a major node named PRINTWAY.

```
PRINTWAY VBUILD TYPE=APPL Application major node
*
PWAPPL01 APPL   AUTH=ACQ  Lets IP PrintWay acquire printer
*
PWAPPL02 APPL   AUTH=ACQ  Lets IP PrintWay acquire printer
```

Figure 33. Sample APPL Statements for IP PrintWay FSSs

where:

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

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

Note: After you create the APPL statement, you must specify the name of the APPL statement in the FSS definition. See “Creating FSS and FSA Definitions in the Printer Inventory” for information.

Creating FSS and FSA Definitions in the Printer Inventory

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.
- FSA definition. IP PrintWay uses values specified in this definition when an IP PrintWay FSA starts.

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* for information about how to use the Infoprint Server ISPF panels or PIDU.

Creating 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. An IP PrintWay FSS definition is *required* in the following situations:

- If any of the FSAs in the FSS will transmit output to VTAM-controlled printers. In this case, you must specify the name of the VTAM APPL statement you defined for the FSS.
- If you need 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.

Table 10 on page 187 shows the fields that you can specify in an FSS definition. The table lists the field names on the ISPF panel, a description of each field, and the default value that IP PrintWay uses if you do not create an FSS definition or if you leave a field blank.

Table 10. IP PrintWay FSS Definition

ISPF Field Name	Description	Default
Applid	The application program ID that IP PrintWay uses to establish a VTAM session with printers. This ID must match the name of the APPL statement defined to VTAM. **Note:** Specify this field only after you have installed the Coax Printer Support feature of Infoprint Server Transforms. If this field is specified but the Coax Printer Support feature is not installed, IP PrintWay abends.	None
Description	A description of the FSS definition, for information purposes only.	None
Document code page	The default EBCDIC code page to use as the source code page when IP PrintWay translates data from EBCDIC to ASCII.	IBM-1047
FSS name	The name of the FSS as defined to JES. It must match the name of the FSS statement in JES2 and the FSSDEF statement in JES3.	None
Hiperspace blocks	The maximum number of 4 KB blocks that each FSA can use in hiperspace. The number you specify can affect system performance and also can limit the size of the data sets the FSA can process. The maximum amount of hiperspace available is 2 gigabytes. You might need to increase this value to process large data sets. IP PrintWay always writes data sets to hiperspace when it uses the LPR transmission protocol. It also writes data sets to hiperspace when it uses the IPP, direct-sockets, or e-mail transmission protocol and either the **Maximum document size** or **Delete form feed** option is selected in the printer definition. IP PrintWay does *not* write data sets to hiperspace when it transmits data to VTAM-controlled printers.	8,000 (32 MB)
National language	The language IP PrintWay uses for messages.	English
Old style translation	An indicator of whether IP PrintWay is to use the standard TCP/IP translate table to convert data from EBCDIC to ASCII when the printer definition does not specify document or printer code pages, a customized TCP/IP translate table, or a DBCS translate table. This field is ignored when IP PrintWay prints to VTAM-controlled printers or to e-mail destinations. [1]	IP PrintWay uses code pages to translate data from EBCDIC to ASCII. IP PrintWay uses the code page that is specified in the **Document code page** field as the source code page and IBM-850 as the target code page.
TCP/IP job name	The name of the TCP/IP startup procedure.	The default TCP/IP name

Table 10. IP PrintWay FSS Definition (continued)

ISPF Field Name	Description	Default
Trace mode	The type of IP PrintWay tracing for the FSS, and the default tracing mode for all FSAs. Tracing starts when the FSS and FSAs starts. The trace mode specified in an FSA definition overrides this trace mode.	No tracing
Trace prompt	An indicator of whether the operator is prompted each time the FSS starts. If this option is selected, message ANFM020A is issued to the operator.	No prompting
Trace table size	The number of 4 KB pages of storage (above the 16 MB line) to allocate for each internal FSA trace table.	32 (128 KB)

1. When printing to VTAM-controlled printers or to e-mail destinations, if the printer definition does not specify code pages or a customized TCP/IP translate table, IP PrintWay does not convert data between code pages.

The following ISPF panel shows a sample FSS definition.

```
                                IP PrintWay FSS

FSS name. . . PRINTWAY
Description . My IP PrintWay FSS                                    (extend)

  _ Old-style translation
  Hiperspace blocks . . 8000
  TCP/IP job name . . . MYTCPIP
  Document code page. . IBM-850
  Applid. . . . . . . . PWAPPL01
  National language . . 1   1. English 2. Japanese
  Trace mode. . . . . . 2   1. None  2. Internal  3. No printing  4. Full
     __ Trace prompt
     Trace table size . ___
```

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.

Creating FSA Definitions

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 a 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.

The following ISPF panel shows a sample IP PrintWay FSA definition.

```
                              IP PrintWay FSA

FSA Name. . . PRT1
Description . Printers in Boulder                                    (extend)

Trace mode . . 2  1. None  2. Internal  3. No printing  4. Full
```

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.

Writing IP PrintWay Exit programs

Note
This section contains programming-interface information.

The following optional exits let you customize IP PrintWay processing:

- Routing Exit

  The Routing exit lets you inspect routing parameters for each data set before the data set is transmitted. You can change the address of the target printer or the e-mail address, change some transmission options, hold the data set on the JES spool, or delete the data set from the JES spool. You *cannot* change the transmission protocol that IP PrintWay uses (LPR, IPP, direct sockets, VTAM, or e-mail). See “Routing Exit (ANFUXRTG)” on page 196 for more information.
- Begin Data Set Exit

  The Begin Data Set exit lets you inspect the transmission options for a data set before the data set is transmitted. You can change transmission options and also add one or more records to the beginning of the data set. See “Begin Data Set Exit” on page 204 for more information.
- Record Exit

  The Record exit lets you add, replace, or delete records in a data set before the data set is transmitted. See “Record Exit” on page 210 for more information.
- End Data Set Exit

  The End Data Set exit lets you add records to the end of a data set before the data set is transmitted. See “End Data Set Exit” on page 216 for more information.
- SMF Exit

  The SMF exit lets you replace or suppress the System Management Facilities (SMF) type-6 record that IP PrintWay writes for each data set that has been transmitted. See “SMF Exit (ANFUXSMF)” on page 222 for more information.
- Message Exit

  The Message exit lets you suppress or change the text of any message that IP PrintWay writes to the IP PrintWay message-log data set. See “Message Exit (ANFUXMSG)” on page 227 for more information.
- Response Notification Exit

  The Response Notification exit lets you perform an action based on the success or failure of the transmission as indicated by the response notification code. See “Response Notification Exit (ANFUXRSP)” on page 228 for more information.

Each exit, except for the Message exit, can also create a message for IP PrintWay to issue to the operator's console or to the IP PrintWay message-log data set so that the exit can inform the operator or system programmer about actions taken by the exit.

IP PrintWay calls the exits in the following order:

1. **Initial data set processing exits**
   - **Routing Exit**

     Once with each data set acquired from the JES spool
   - **Response Notification Exit**

     Once with each data set acquisition if an error occurs that keeps IP PrintWay from transmitting the data set
2. **Data set transmission processing exits**

   Data set transmission processing exits run in sequence. IP PrintWay does not invoke any other exits until after the data set has been transmitted to the remote system.
   - **Begin Data Set Exit**

     Once with each transmission attempt
   - **Record Exit**

     Once with each record, after the Begin Data Set Exit completes
   - **End Data Set Exit**

     Once with each transmission attempt, after the last Record Exit completes
   - **Response Notification Exit**

     Once with each transmission attempt, after the End Data Set Exit completes
3. **SMF Exit**

   When a data set is to be purged from the JES queue, following the IP PrintWay retention rules

IP PrintWay calls the Message exit before writing any message to the IP PrintWay message-log data set.

Note: When the **Resubmit for filtering** option is selected in the printer definition, IP PrintWay calls the Routing, Begin Data Set, Record, and End Data set exits after Print Interface has transformed the data. See "Resubmit for Filtering Considerations for IP PrintWay Exits" for more information.

Resubmit for Filtering Considerations for IP PrintWay Exits

Refer to *z/OS Infoprint Server Operation and Administration* for a detailed description of IP PrintWay and Print Interface processing when the **Resubmit for filtering** option is selected in a printer definition. When the **Resubmit for filtering** option is selected, IP PrintWay processes data two times, once when it selects the original data set from the JES spool and a second time when it selects the data set that Print Interface has allocated the JES spool.

When the **Resubmit for filtering** option is selected, IP PrintWay calls the Response Notification and SMF exits once for each data set. However, IP PrintWay calls the Routing, Begin Data Set, Record, and End Data Set exits only for the second data set. When the Routing, Begin Data Set, Record, and End Data Set exits are called, Print Interface has transformed data into the new data format.

When IP PrintWay calls the Routing, Begin Data Set, Record, and End Data Set exits, some fields in the ANFUEXTP control block no longer contain the values

originally specified by the job submitter in JCL parameters; instead, these fields contain values specified in the printer definition or, if no value is specified, a default value assigned by JES. Table 11 summarizes the values in some of the ANFUEXTP fields when the **Resubmit for filtering** option is selected.

Table 11. Contents of ANFUEXTP Fields When Resubmit for Filtering Option is Selected

ANFUEXTP Field Names	Contents of Field
XTPKEY	Name of the printer definition.
XTPIPADR, XTPPRTQU, XTPIPORT	IP address, print queue name, and port number specified in the printer definition.
XTPRETNS, XTPRETNF	Retain times specified in the printer definition.
XTPRTRYL, XTPRTRYT	Retry values specified in the printer definition.
XTPUSRID	User ID of the person who submitted the original batch job.

To obtain the original JCL values specified by the job submitter, use the Scheduler Work Block Text Unit (SWBTU). (The address of the SWBTU is in field XTPSWBTP in the ANFUEXTP control block.) Following are the JCL parameters in the SWBTU that contain the original values specified by the job submitter when the Routing, Begin Data Set, Record, and End Data Set exits are called:

- ADDRESS, BUILDING, DEPT, NAME, ROOM, TITLE
- DATACK
- DUPLEX
- CHARS (If the CHARS parameter is not specified, the value in the UCS parameter is used.)
- COPIES
- FORMDEF
- FORMS
- INTRAY, OUTBIN
- OFFSETXB, OFFSETXF, OFFSETYB, OFFSETYF
- OVERLAYB, OVERLAYF
- PAGEDEF (If the PAGEDEF parameter is not specified, the value in the FCB parameter is used.)
- PRMODE
- TRC
- USERLIB

JCL parameters not in this list, such as the CLASS and DEST parameters, do not contain the original JCL value specified by the job submitter when the Routing, Begin Data Set, Record, and End Data Set exits are called; instead, they contain values from the Allocation section of the printer definition.

The Begin Data Set, End Data Set, SMF, and Response Notification exits can check field XTP_FILTERING in the ANFUEXTP control block to determine whether or not the **Resubmit for filtering** option is selected in the printer definition.

Programming Considerations

Consider the following points when coding an exit:

- You should be an experienced programmer who is aware of overall system implications.
- Do not issue an explicit or implied WAIT in the exit program, because it puts the IP PrintWay FSA into a WAIT state.
- The exits run in 31-bit addressing mode.
- Code references to fields in control blocks using the names associated with each field in the DSECT control block. Do not use absolute displacements, because these displacements can change from one release to another.
- Code the exits to be reentrant.
- All exit programs run in supervisor state with a protection key of 1.
- Programming exceptions cause IP PrintWay to abend; an ESTAI is in effect while the exit programs are running.
- Each IP PrintWay FSA calls all exits except for the Message exit serially, passing the same Exit Parameter Area (ANFUEXTP) to each exit. IP PrintWay passes the Message exit a separate copy of the ANFUEXTP control block.
- All exits, except for the Message exit, can obtain two work areas and pass pointers to these work areas in the Exit Parameter Area. Using the work areas, one exit can pass data to another exit or to the next invocation of the same exit.
- The data set transmission processing exits are invoked in sequence for the transmission of a single spool data set. This allows information specific to a spool data set to be passed from one of these exits to another, using the work area. Because the Routing exit, the SMF exit, and the Message exit are not invoked in sequence for a particular spool data set, you should not attempt to pass specific spool data set information using these exits.
- A return code of 8 to the Response Notification exit indicates that an error was encountered in IP PrintWay's initial processing of a data set. In this case, no data set transmission processing exits are invoked; therefore, no communication with these exits is possible.

Note: Future IP PrintWay program maintenance may require that you reassemble the exits.

Common Control Blocks

All exits, except for the Message exit, have access to data set and job information contained in the following control blocks:

- JES job separator page data area (IAZJSPA)
- Scheduler work block text unit (SWBTU)

All exits have access to data set and job information contained in the following control block:

- IP PrintWay Exit Parameter Area (ANFUEXTP)

JES Job Separator Page Data Area (IAZJSPA)

The JES job separator page data area (IAZJSPA) is a system control block that contains job and data set information. Refer to *z/OS MVS Using the Functional Subsystem Interface* for more information on the contents of the IAZJSPA.

Scheduler Work Block Text Unit (SWBTU)

The SWBTU is a system control block that contains JCL parameters for the data set. These JCL values are merged from the DD statement, the OUTPUT statement, and JES defaults, according to a hierarchy determined by JES. If NetSpool or Print Interface allocated the data set on the JES spool, the JCL values come from the printer definition or Infoprint Server job attributes.

You can use the SWBTUREQ macro to obtain the JCL parameters specified for the data set from the SWBTU. For example, you could obtain distribution parameters specified on the OUTPUT statement and print them on a separator page created in your Begin Data Set exit. Refer to *z/OS Infoprint Server User's Guide* for information on how to specify distribution parameters on an OUTPUT statement.

For more information about the SWBTUREQ macro, refer to *z/OS MVS Programming: Authorized Assembler Services Reference SET-WTO*. For more information about SWBs and how to use the SWB token to retrieve keywords, refer to *z/OS MVS Using the Subsystem Interface*.

If your installation selects the **Resubmit for filtering** option in the printer definitions, see "Resubmit for Filtering Considerations for IP PrintWay Exits" on page 190 for more information about the JCL parameters in this control block.

IP PrintWay Exit Parameter Area (ANFUEXTP)

The IP PrintWay Exit Parameter Area (ANFUEXTP) provides information to the exits about the data set being processed. Also, the exits can set flags in ANFUEXTP to specify the action IP PrintWay is to take at return and can modify values in ANFUEXTP to change the processing of IP PrintWay.

The same ANFUEXTP control block is shared by all exits, except for the Message exit, that are called by the same IP PrintWay FSA; however, an exit program does not need to enqueue on this control block, because IP PrintWay calls the exits serially.

See the description of each exit for the fields that are valid at input and the fields that each exit can modify. Figure 34 on page 194 shows the format of ANFUEXTP. Macro ANFUEXTP provides a DSECT in assembler language for this control block.

```
Offsets  Hex Type      Length Name          Description
    0    (0) Structure  1280  ANFUEXTP      Exit parameter area
    0    (0) Character     4  XTPID         Eye catcher - 'XTP '
    4    (4) Unsigned      4  XTPSAVE(18)   Save area for exit
   76   (4C) Address       4  XTPWORK1      Work area for exit
   80   (50) Address       4  XTPWORK2      Work area for exit
   84   (54) Character    53  XTPDSNAM      Data set name
  137   (89) Character     3  *             Reserved
  140   (8C) Character     8  XTPUSRID      User id of submitter
  148   (94) Bit string    1  XTPSRCFL      Flags showing where values
                                            came from. If the bit is on
                                            the value came from JCL on
                                            OUTPUT; otherwise it came from
                                            the printer definition or defaults:
             1...  ....       XTPSOPTS       PRTOPTNS from JCL
             .1..  ....       XTPSIPAD       IPADDR from JCL
             ..1.  ....       XTPSPRTQ       PRTQUEUE from JCL
             ...1  ....       XTPSRTNS       RETAINS from JCL
             ....  1...       XTPSRTNF       RETAINF from JCL
             ....  .1..       XTPSRTYL       RETRYL from JCL
             ....  ..1.       XTPSRTYT       RETRYT from JCL
             ....  ...1       XTPSPORT       PORTNO from JCL
149   (95) Bit string      1  XTPDSFLG      Data set information flags:
             1...  ....       XTPDSCC        data set has carriage controls
             .1..  ....       XTPDSHLD       release data set to JES; data
                                              set is held on JES spool
             ..1.  ....       XTPDSTRM       release data set to JES; data
                                              set is deleted from JES spool
             ...1  ....       XTPDSRTG       reselect printer definition
             ....  1...       XTPDSJOB       data set is first in job
             ....  .1..       XTPDSJNW       data set is JESNEWS data set
             ....  ..1.       XTPDSEJB       data set is last in job
             ....  ...1       XTPDSCON       data set is in a concatenation
  150   (96) Bit string    1  XTPRCFLG      Record flags:
             1...  ....       XTPRCTRN       translate record from exit
             .1..  ....       XTPRCCC        record from exit has
                                              carriage controls
             ..1.  ....       XTPRCORG       transmit original record
             ...1  ....       XTPRDFST       1st call to exit for record
                                              or data set
             ....  1...       XTPRLAST       last record from exit
             ....  .1..       XTPRCEXT       transmit record from exit
             ....  ..1.       XTPRCDBL       record from exit is double-byte
             ....  ...1       *              reserved
  151   (97) Bit string    1  XTPMSGFL      Message flags:
             1...  ....       XTPM2CON       issue message to console
             .1..  ....       XTPM2MDS       issue message to message
                                              data set
             ..11  1111       *              reserved
  152   (98) Address       4  XTPERPTR      Address of record from exit
  156   (9C) Unsigned      4  XTPERLEN      Length of record from exit
  160   (A0) Address       4  XTPPRPTR      Address of original record
  164   (A4) Unsigned      4  XTPPRLEN      Length of original record
  168   (A8) Character    16  XTPOPTNM      Name of components
  184   (B8) Unsigned      2  XTPLOLEN      Length of LPR options
  186   (BA) Unsigned      2  XTPPQLEN      Length of value in XTPPRTQU
  188   (BC) Unsigned      2  XTPIALEN      Length of value in XTPIPADR
  190   (BE) Unsigned      2  XTPMSGLN      Length of message
  192   (C0) Character   256  XTPPRTQU      Name of print queue or VTAM logmode
448  (1C0) Character   256  XTPIPADR      IP address/host name, URL, LU name,
                                            or e-mail addresses
```

Figure 34. Format of the IP PrintWay Exit Parameter Area (ANFUEXTP) (Part 1 of 2)

```
  704  (2C0) Unsigned      4  XTPRETNS       Success retain time (see note)
  708  (2C4) Unsigned      4  XTPRETNF       Failure retain time (see note)
  712  (2C8) Unsigned      4  XTPRTRYL       Retry limit
  716  (2CC) Unsigned      4  XTPRTRYT       Retry time (see note)
  720  (2D0) Character    17  XTPKEY         Name of printer definition
  720  (2D0) Character     8  XTPDEST        DEST
  728  (2D8) Character     1  XTPCLASS       CLASS
  729  (2D9) Character     8  XTPFORMS       FORMS
  737  (2E1) Bit string    1  XTPRECFM       Record format
  738  (2E2) Unsigned      2  XTPMRECL       Maximum record length
  740  (2E4) Address       4  XTPSWBTP       Address of SWBTU area
  744  (2E8] Unsigned      4  XTPSWBTL       Length of SWBTU area
  748  (2EC) Address       4  XTPJSPAP       Address of JSPA
  752  (2F0) Unsigned      4  XTPCOPYS       Number of copies
  756  (2F4) Address       4  XTPMSGP        Address of message
  760  (2F8) Unsigned      4  XTPIPORT       PORTNO value
  764  (2FC) Bit String    1  XTP_MISC_FLAGS Miscellaneous flags:
             1... ....       XTP_FIRST_REC   first record of a data set
             .111 1111       *               reserved
  765  (2FD) Character     3  *              Reserved
=============================================================================
 The following fields are used by the Block Letter program.
=============================================================================
  768  (300) Bit string    1  XTPBIFLG       Input flags for block program:
             1...  ....      XTPBSLNT        slant the letters
             .1..  ....      XTPBPRFM        performance (narrow) letters
             ..1.  ....      XTPBLJST        left justify letters
             ...1  ....      XTPBFRST        first call to block program
                                              for this string
             ....  1111      *               reserved
  769  (301) Bit string    1  XTPBOFLG       Output flags for block program:
             1...  ....      XTPBLAST        this is the last line
             .111  1111      *               reserved
  770  (302) Unsigned      1  XTPBLENG       Length of output record
  771  (303) Character     1  *              Reserved
  772  (304) Character     8  XTPBSTRG       String to convert, blank padded
  780  (30C) Character   150  XTPBOUTP       Output string
  930  (3A2) Character    16  XTPBWRKA       Work area for block program
  946  (3B2) Character     2  *              Reserved
  948  (3B4) Signed        4  XTPRSPCD       Response code
=============================================================================
 The following fields are used by all exits.
=============================================================================
  952  (3B8) Unsigned      4  XTP_ADDR_TYPE  Protocol that IP PrintWay uses:
                                             0 = LPR
                                             1 = IPP
                                             2 = direct sockets
                                             3 = VTAM
                                             4 = e-mail
  956  (3BC) Unsigned      4  XTP_ROUTING_KEY_TYPE
                                            Value in XTPKEY field:
                                             0 = DEST, CLASS, and FORMS
                                             1 = Printer definition name
960  (3C0) Unsigned      4  XTP_FILTERING  Resubmit for filtering values:
                                             0 = No resubmit for filtering
                                             1 = This is first data set processed
                                             2 = This is second data set processed
  964  (3C4) Character     8  XTP_JCL_DEST   DEST from JCL
  972  (3CC) Character     1  XTP_JCL_CLASS  CLASS from JCL
  973  (3CD) Character     8  XTP_JCL_FORMS  FORMS from JCL
  981  (3D5) Character    19  *              Reserved
  1000 (3E8) Character   868  XTPOPTNS       Transmission options
                                            (See macro for field names)
```

Figure 34. Format of the IP PrintWay Exit Parameter Area (ANFUEXTP) (Part 2 of 2)

Note: The time in the `XTPRETNS`, `XTPRETNF`, and `XTPRTRYT` fields is a hexadecimal number, calculated as follows: (*hours* x 10 000) + (*minutes* x 100) + *seconds*. For example, if the time period is 24 hours, 5 minutes, and 1 second, this field contains 03AB75, the hexadecimal representation of the decimal number 240501.

Routing Exit (ANFUXRTG)

The Routing exit lets you inspect routing parameters for each data set before the data set is transmitted. When the Routing exit receives control, the routing parameters from the DD and OUTPUT JCL statement for the data set and from the printer definition in the Printer Inventory have been merged. The parameters on the DD and OUTPUT JCL statement always override the values specified in the printer definition. The XTPSRCFL field in the ANFUEXTP control block indicates whether or not each routing parameter was specified on a JCL statement.

If you determine in the exit that one or more routing parameters need to be changed, you can either (1) modify the routing parameters directly in the exit or (2) request that IP PrintWay obtain new routing parameters from a specific printer definition. You identify the new printer definition by either the printer definition name or the class, destination, and form name assigned to it. IP PrintWay does *not* call the Routing exit again for this data set after obtaining parameters from the new printer definition.

Instead of changing routing parameters, you can request that IP PrintWay have JES hold the data set on the JES spool for operator intervention, or you can request that IP PrintWay delete the data set from the JES spool.

IP PrintWay does *not* issue any messages to the IP PrintWay message log or to the job submitter when an exit requests that a data set be held or deleted. However, you can request that IP PrintWay issue a message created by the exit either to the operator's console or to the IP PrintWay message-log data set. IP PrintWay assigns the same message ID to each message created by an exit: ANFM155I.

IP PrintWay calls the same Routing exit before processing each data set. You can install different Routing exits for different IP PrintWay FSSs. See "Installing Exits" on page 236 for information.

You might want to write a Routing exit to perform one of the following functions:

- Enforce limits in JCL parameters. For example, your installation might want to limit the length of the retention time specified by job submitters on the OUTPUT JCL statement. To do this, your Routing exit could inspect the values specified in the RETAINS and RETAINF JCL parameters and change them if they exceed the maximum time defined for your installation.
- Restrict the use of certain printers to specific users. To do this, your Routing exit could inspect (1) the user ID of the job submitter and (2) the host name or IP address and the print queue name or port of the target printer. Your exit could hold or delete a data set if the user ID of the job submitter is not on your list of authorized users IDs.

Note: IP PrintWay does *not* call the Routing exit under these conditions:

- IP PrintWay cannot find a printer definition for a data set and the DEST=IP JCL parameter is not specified. In this case, IP PrintWay requests that JES hold the data set for operator intervention.

- The **Resubmit for filtering** option is selected in the printer definition and IP PrintWay is transmitting the data set to Print Interface for filtering. See "Resubmit for Filtering Considerations for IP PrintWay Exits" on page 190 for more information.

Input to the Routing Exit

At entry to the Routing exit, the registers contain the following:

Register 1 Pointer to the address of the ANFUEXTP control block. The ANFUEXTP control block contains addresses of the IAZJSPA and the SWBTU control blocks.

Register 13 Address of an 18-word save area for saving the caller's registers.

Register 14 Return address.

Register 15 Entry-point address.

The following fields in the ANFUEXTP control block are valid when the Routing exit is called. The format of ANFUEXTP is shown in Figure 34 on page 194.

XTPSAVE A register save area for use by the exit.

XTPWORK1 Address of a work area previously obtained by an exit. This field contains zeroes if a work area has not been obtained.

XTPWORK2 Address of a second work area previously obtained by an exit. This field contains zeroes if a work area has not been obtained.

XTPDSNAM Fully-qualified name of the data set being processed, padded to the right with blanks.

XTPUSRID User ID of the job submitter, padded to the right with blanks, with the following exceptions:

- If NetSpool allocated the data set on the JES spool, this field contains the user ID assigned to the NetSpool started task.
- If Print Interface allocated the data set on the JES spool and the print request was not submitted from an OS/390 or z/OS system, this field contains the user ID from the remote system of the person who submitted the print request

XTPSRCFL Flags indicating whether or not a parameter was specified as a JCL parameter.

If the flag is set to 1, the indicated parameter was specified as a JCL parameter.

If the flag is set to 0, the JCL parameter was not specified. Instead, IP PrintWay obtained the value from a printer definition. If the DEST=IP parameter was specified in JCL without the name of a printer definition, IP PrintWay obtained the value from the IP PrintWay default printer definition; if no PrintWay default printer definition exists, IP PrintWay sets a default value.

Flag	Meaning When Set to B'1'
XTPSOPTS	The PRTOPTNS parameter specified the name of components in the Printer Inventory.
XTPSIPAD	The DEST=IP parameter specified the host name or IP address of the target system.
XTPSPRTQ	The PRTQUEUE parameter specified the name of the target print queue.

XTPSRTNS The RETAINS parameter specified the retention time after a successful transmission.

XTPSRTNF The RETAINF parameter specified the retention time after a failed transmission.

XTPSRTYL The RETRYL parameter specified the retry limit.

XTPSRTYT The RETRYT parameter specified the retry time.

XTPSPORT The PORTNO parameter specified the port number.

XTPDSFLG Flags containing data set information:

Flag	Meaning When Set to B'1'
XTPDSJOB	This is the first data set in the JES output subgroup.
XTPDSJNW	This data set is the JESNEWS data set.
XTPDSEJB	This is the last data set in the JES output subgroup.
XTPDSCON	This data set is part of an IP PrintWay concatenation. This flag is set when the **Concatenate job** option is selected in the **Dataset grouping** field of the printer definition and IP PrintWay is transmitting this data set with other data sets in the same transmission.

XTPOPTNM The component name that was specified in the PRTOPTNS JCL parameter. The name is case sensitive and is padded to the right with blanks. This field contains blanks if no component name was specified in the PRTOPTNS parameter.

XTPPQLEN Length of the value in field XTPPRTQU.

XTPIALEN Length of the value in field XTPIPADR.

XTPPRTQU Name of the target print queue specified in the PRTQUEUE JCL parameter or in a printer definition; or the VTAM logmode specified in a printer definition. Field XTP_ADDR_TYPE identifies which value is in this field.

XTPIPADR Field XTP_ADDR_TYPE identifies which of the following values this field contains:

- Host name or IP address of the target system specified in the DEST=IP JCL parameter or in a printer definition
- URL of the target system specified in the printer definition
- LU name of the target printer specified in the printer definition
- E-mail addresses specified in the printer definition

XTPRETNS Period of time IP PrintWay is to retain the data set on the JES spool after a successful transmission. This time was specified in the RETAINS JCL parameter or in the printer definition. IP PrintWay might have set a default value of zeroes. See the note on page 196 for the format of this field.

XTPRETNF Period of time IP PrintWay is to retain the data set on the JES spool after all transmission retries have failed. This time was specified in the RETAINF JCL parameter or in the printer definition. IP PrintWay might have set a default value of zeroes. See the note on page 196 for the format of this field.

XTPRTRYL Number of times IP PrintWay is to retry transmission of the data set. This number was specified in the RETRYL JCL parameter or in the printer definition. IP PrintWay might have set a default value of zero.

XTPRTRYT Period of time IP PrintWay is to wait between retries of the transmission. This time was specified in the RETRYT JCL parameter or in the printer definition. IP PrintWay might have set a default value of zero. See the note on page 196 for the format of this field.

XTPKEY If XTP_ROUTING_KEY_TYPE is set to '1', this field contains the name of the printer definition that was specified in JCL or in the printer definition; the name is padded to the right with blanks. The printer definition name is case sensitive.

Fields XTPDEST, XTPCLASS, and XTPFORMS are mapped to the same location as XTPKEY.

XTPDEST If XTP_ROUTING_KEY_TYPE is set to '0', this field contains the value in the DEST field of the printer definition that IP PrintWay uses to print the data set. The name is padded to the right with blanks. If the DEST field in the printer definition is not specified, this field contains blanks.

Note: Use field XTP_JCL_DEST if you want to obtain the value specified in the JCL parameter.

XTPCLASS If XTP_ROUTING_KEY_TYPE is set to '0', this field contains the value in the CLASS field of the printer definition that IP PrintWay uses to print the data set. If the CLASS field in the printer definition is not specified, this field contains blanks.

Note: Use field XTP_JCL_CLASS if you want to obtain the value specified in the JCL parameter.

XTPFORMS If XTP_ROUTING_KEY_TYPE is set to '0', this field contains the value in the FORMS field of the printer definition that IP PrintWay uses to print the data set. The name is padded to the right with blanks. If the FORMS field in the printer definition is not specified, this field contains blanks.

Note: Use field XTP_JCL_FORMS if you want to obtain the value specified in the JCL parameter.

XTPRECFM Record format as indicated in the job file control block (JFCB). Refer to *z/OS MVS Data Areas, Vol 3 (IVT-RCWK)* for a description of the JFCRECFM field in the JFCB.

XTPMRECL Maximum record length

XTPSWBTP Address of the SWBTU for the data set. See "Scheduler Work Block Text Unit (SWBTU)" on page 192 for more information.

XTPSWBTL Length of the SWBTU.

XTPJSPAP Address of the JES job separator page data area for the data set being processed (IAZJSPA). See "JES Job Separator Page Data Area (IAZJSPA)" on page 192 for more information.

XTPCOPYS Number of copies of the data set. This number was specified in JCL, in the printer definition, or JES provided a default value.

XTPIPORT Port number on the target system. This number was specified in the PORTNO JCL parameter or in the printer definition.

XTP_ADDR_TYPE
This field identifies the transmission protocol and the contents of fields XTPIPADR, XTPPRTQU and XTPIPORT:

Value	Protocol	XTPIPADR	XTPPRTQU	XTPIPORT
0	LPR	IP address	Print queue name	
1	IPP	URL		
2	Direct sockets	IP address		Port number
3	VTAM	LU name	Logmode	
4	e-mail	e-mail addresses		

XTP_ROUTING_KEY_TYPE
An indicator of how IP PrintWay selected the printer definition it will use to obtain printing options:

Value	Meaning
0	IP PrintWay selected the printer definition by matching the DEST, CLASS, and FORMS fields specified in JCL parameters with the values in the printer definition.
1	IP PrintWay selected the printer definition using the printer definition name specified in JCL parameters or specified by NetSpool or Print Interface.

XTP_JCL_DEST
The DEST value with which the output data set was allocated on the JES spool. If NetSpool or Print Interface allocated the data set, then this is the DEST value in the printer definition. Otherwise, it is the DEST value specified in a JCL parameter unless the **Resubmit for filtering** option is selected in the printer definition, in which case this field contains the DEST value specified in the printer definition. The value is padded to the right with blanks.

XTP_JCL_CLASS
The CLASS value with which the output data set was allocated on the JES spool. If NetSpool or Print Interface allocated the data set, then this is the CLASS value in the printer definition. Otherwise, it is the CLASS value specified in a JCL parameter unless the **Resubmit for filtering** option is selected in the printer definition, in which case this field contains the CLASS value specified in the printer definition. The value is padded to the right with blanks.

XTP_JCL_FORMS
The FORMS value with which the output data set was allocated on the JES spool. If NetSpool or Print Interface allocated the data set, then this is the FORMS value in the printer definition. Otherwise, it is the FORMS value specified in a JCL parameter. The value is padded to the right with blanks.

Output from the Routing Exit

At return, restore the contents of all registers. Do not set a return code in register 15.

The exit can modify the following fields in the ANFUEXTP control block:

XTPWORK1 Address of a work area obtained by this exit. This address is passed to every exit except the Message exit.

XTPWORK2 Address of a second work area obtained by this exit. This address is passed to every exit except the Message exit.

XTPDSFLG Flags indicating the action you want IP PrintWay to take upon return. IP PrintWay sets these flags to B'0' before calling the exit. To transmit the data set, do *not* set any of these flags.

Note: Set only one of the following flags.

Flag	Meaning When Set to B'1'
XTPDSHLD	Hold the data set on the JES spool for operator intervention.
XTPDSTRM	Delete the data set from the JES spool.
XTPDSRTG	Reselect the printer definition using the value in the XTPKEY field.

XTPMSGFL Flags indicating the destination of a message created by this exit. IP PrintWay sets these flags to B'0' before calling the exit. Set both flags to B'1' to issue the message to both destinations. Place the address of the message in field XTPMSGP and the length of the message in field XTPMSGLN.

Flag	Meaning When Set to B'1'
XTPM2CON	Issue the message to the operator's console.
XTPM2MDS	Issue the message to the IP PrintWay message-log data set.

XTPMSGLN Length of the message whose address is in field XTPMSGP.

XTPMSGP Address of the message created by this exit. IP PrintWay adds a message ID, ANFM155I, to the beginning of this message.

If you set the XTPDSRTG flag, you can also change the following fields in the ANFUEXTP control block. (IP PrintWay ignores changes you make to fields containing routing parameters, such as fields XTPOPTNM, XTPPRTQU, XTPIPADR, XTPRETNS, XTPRETNF, XTPRTRYL, and XTPRTRYT.)

XTPKEY Name of the printer definition you want IP PrintWay to use. This field is case sensitive. Also set XTP_ROUTING_KEY_TYPE to '1'.

Fields XTPDEST, XTPCLASS, and XTPFORMS are mapped to the same location as XTPKEY.

XTPCLASS Class specified in the CLASS field of the printer definition you want IP PrintWay to use. Do not use lowercase characters. Also set XTP_ROUTING_KEY_TYPE to '0'.

XTPDEST Destination name specified in the DEST field of the printer definition you want IP PrintWay to use,

padded to the right with blanks. Do not use lowercase characters. Also set XTP_ROUTING_KEY_TYPE to 0.

Note: Do not specify DFLTNTRY as the destination name.

XTPFORMS Form name specified in the FORMS field of the printer definition you want IP PrintWay to use, padded to the right with blanks. Do not use lowercase characters. Also set XTP_ROUTING_KEY_TYPE to '0'.

XTPCOPYS Number of copies of the data set. Specify a number from 1 - 255.

XTP_ROUTING_KEY_TYPE
Indicates how you want IP PrintWay to select the new printer definition:

Value	Meaning
0	IP PrintWay is to select the printer definition by matching the values in fields XTPCLASS, XTPDEST, and XTPFORMS with the values in the printer definition.
1	IP PrintWay is to select the printer definition using the printer definition name in field XTPKEY.

If you do *not* set any of the flags in field XTPDSFLG, you can change the values in the following fields in the ANFUEXTP control block. IP PrintWay uses the changed values when it transmits the data set. (IP PrintWay ignores changes you make to fields XTPKEY, XTPCLASS, XTPFORMS, and XTPDEST.)

XTPOPTNM The component name that was specified in the PRTOPTNS JCL parameter. The name is case sensitive and is padded to the right with blanks. This field contains blanks if no component name was specified in the PRTOPTNS parameter.

XTPPQLEN Length of the value in field XTPPRTQU.

XTPPRTQU Name of the target print queue or the VTAM logmode.

XTPIALEN Length of the value in field XTPIPADR.

XTPIPADR Specify one of the following values:

- Host name or IP address of the target system
- URL of the target system
- LU name of the target printer
- E-mail address list. Separate each e-mail address or z/OS UNIX sendmail alias name with a comma.

XTPRETNS Period of time IP PrintWay is to retain the data set on the JES spool after a successful transmission. See the note on page 196 for the format of this field.

XTPRETNF Period of time IP PrintWay is to retain the data set on the JES spool after all transmission retries have failed. See the note on page 196 for the format of this field.

XTPRTRYL Number of times IP PrintWay is to retry transmission of the data set. Specify a number from 0 - 32767.

XTPRTRYT Period of time IP PrintWay is to wait between retries of the transmission. See the note on page 196 for the format of this field.

XTPCOPYS Number of copies of the data set. Specify a number from 1 - 255.

XTPIPORT Port number on the target system.

Examples

1. To hold the data set on the JES spool for operator intervention, set flag XTPDSHLD=1 in the ANFUEXTP control block.
2. To delete the data set from the JES spool, set flag XTPDSTRM=1 in the ANFUEXTP control block.
3. To request that IP PrintWay obtain routing parameters from a specific printer definition, do the following:
   - Set XTPDSRTG=1 in the ANFUEXTP control block.
   - Specify the printer definition name in field XTPKEY; or specify the class, destination name, and forms name in fields XTPCLASS, XTPDEST, and XTPFORMS.
   - Set XTP_ROUTING_KEY_TYPE, depending on whether you specified the printer definition name or the class, destination, and forms names.
4. To transmit the data set without changing any routing information, simply return control to IP PrintWay.
5. To transmit the data set to:
   - A different port number, overriding the previous port number:
     – Place the port number in XTPIPORT
   - A different print queue, overriding the previous print queue:
     – Place the name of the target print queue in field XTPPRTQU.
     – Place the length of the target print queue name in field XTPPQLEN.
   - A different VTAM LU name, overriding the previous LU name:
     – Place the name of the LU name in field XTPIPADR.
     – Place the length of the LU name in field XTPIALEN.
   - A different e-mail address, overriding the previous e-mail addresses:
     – Place the e-mail addresses and z/OS UNIX sendmail alias names in field XTPIPADR. Separate each address and alias name with a comma.
     – Place the length of the e-mail addresses in field XTPIALEN.

   Note: You *cannot* change the protocol that IP PrintWay uses in the XTP_ADDR_TYPE field.
6. To request that IP PrintWay issue a message to the IP PrintWay message-log data set, do the following:
   - Set XTPM2MDS=1 in the ANFUEXTP control block to write the message to the IP PrintWay message-log data set.
   - Place the address of the message in field XTPMSGP. IP PrintWay adds a message ID, ANFM155I, to the beginning of this message.
   - Place the length of the message in field XTPMSGLN.

Refer also to the sample exits distributed with IP PrintWay and described in “Sample Exits” on page 235.

Begin Data Set Exit

The Begin Data Set exit lets you inspect the transmission options for a data set before the data set is transmitted. In this exit, you can change most of the transmission options, and you can add a record to the beginning of the data set. You can also inspect the routing parameters of the data set, but you *cannot* change them.

If you add a record, you can specify whether or not (1) IP PrintWay should translate the record to ASCII or EBCDIC, (2) the record contains a carriage control character, and (3) the record contains double-byte characters. You can add more than one record to the beginning of the data set by requesting that IP PrintWay call this exit multiple times for the data set.

The Begin Data Set exit can use the Block Letter program (ANFUBLK), provided with IP PrintWay, to build block letters suitable for printing on a separator page. See "Using the Block Letter Program (ANFUBLK)" on page 234 for more information.

You can also create a message for IP PrintWay to issue either to the operator's console or to the IP PrintWay message-log data set. IP PrintWay assigns the same message ID to each message created by an exit: ANFM155I.

You name the Begin Data Set exit in the printer definition; therefore, you can write different Begin Data Set exits for different data sets, provided that the data sets use different printer definitions.

You might want to write a Begin Data Set exit to perform one of the following functions:

- Add printer instructions (a printer setup string) to the start of the data set. For example, your installation might want to add printer instructions for landscape printing. [1]
- Add a custom separator page. On the separator page, you can print job and data set information contained in the JES job separator page data area (IAZJSPA) or the scheduler work block text unit (SWBTU). Refer to "JES Job Separator Page Data Area (IAZJSPA)" on page 192 and "Scheduler Work Block Text Unit (SWBTU)" on page 192 for more information about these system control blocks.

Notes:

1. IP PrintWay does not call the Begin Data Set exit when the **Resubmit for filtering** option is selected in the printer definition and IP PrintWay is transmitting the data set to Print Interface for filtering. See "Resubmit for Filtering Considerations for IP PrintWay Exits" on page 190 for more information.
2. IP PrintWay adds printer instructions specified in the **Document header** field of the printer definition to the data set *after* it calls the Begin Data Set exit. Therefore, printer instructions in the **Document header** field do *not* apply to any records added by the Begin Data Set exit. For example, if the **Document header** field contains printer instructions for landscape printing, and you add a separator page in the Begin Data Set exit, then the separator page does *not* print in the landscape direction. If you want the separator page to print in the landscape direction, you must add printer instructions for landscape printing in the Begin Data Set exit.

1. If you add a setup string to data, in some cases you may need to redefine the print driver on your workstation. For example, when printing data on an HP Laserjet IIISi printer on a Windows NT system, define the HP print driver as a "generic/text only" driver. This prevents the print driver from changing your setup strings to be invalid.

Input to the Begin Data Set Exit

At entry to the Begin Data Set exit, the registers contain the following:

Register 1 Pointer to the address of the ANFUEXTP control block. The ANFUEXTP control block contains addresses of the IAZJSPA and the SWBTU control blocks.

Register 13 Address of an 18-word save area for saving the caller's registers.

Register 14 Return address.

Register 15 Entry-point address.

The following fields in ANFUEXTP are valid when the Begin Data Set exit is called. The format of the ANFUEXTP control block is shown in Figure 34 on page 194.

XTPSAVE A register save area for use by the exit.

XTPWORK1 Address of a work area previously obtained by an exit. This field contains zeroes if a work area has not been obtained.

XTPWORK2 Address of a second work area previously obtained by an exit. This field contains zeroes if a work area has not been obtained.

XTPDSNAM Fully qualified name of the data set being processed, padded to the right with blanks.

XTPUSRID User ID of the job submitter, padded to the right with blanks, with the following exceptions:

- If NetSpool allocated the data set on the JES spool, this field contains the user ID assigned to the NetSpool started task.
- If Print Interface allocated the data set on the JES spool and the print request was not submitted from an OS/390 or z/OS system, this field contains the user ID from the remote system of the person who submitted the print request

XTPDSFLG Flags containing data set information:

Flag	Meaning When Set to B'1'
XTPDSCC	This data set contains carriage control characters.
XTPDSJOB	This is the first data set in the job.
XTPDSJNW	This data set is the JESNEWS data set.
XTPDSEJB	This is the last data set in the job.
XTPDSCON	This data set is part of a concatenation.

XTPRCFLG Flags containing data set information:

Flag	Meaning When Set to B'1'
XTPRDFST	This is the first call to the exit for this data set.

XTPPQLEN Length of the value in field XTPPRTQU.

XTPIALEN Length of the value in field XTPIPADR.

XTPPRTQU Name of the target print queue specified in the PRTQUEUE JCL parameter, in a printer definition, or by the Routing exit; or the VTAM logmode specified in a printer definition or by the Routing exit. Field XTP_ADDR_TYPE identifies which value is in this field.

XTPIPADR Field XTP_ADDR_TYPE identifies which of the following values this field contains:

- Host name or IP address of the target system specified in the DEST=IP JCL parameter, in a printer definition, or by the Routing exit
- URL of the target system specified in the printer definition or by the Routing exit
- LU name of the target printer specified in the printer definition or by the Routing exit
- E-mail addresses specified in the printer definition or by the Routing exit

XTPRETNS Period of time IP PrintWay is to retain the data set on the JES spool after a successful transmission. This time was specified in the RETAINS JCL parameter, in the printer definition, or by the Routing exit. IP PrintWay might have set a default value of zeroes. See the note on page 196 for the format of this field.

XTPRETNF Period of time IP PrintWay is to retain the data set on the JES spool after all transmission retries have failed. This time was specified in the RETAINF JCL parameter, in the printer definition, or by the Routing exit. IP PrintWay might have set a default value of zeroes. See the note on page 196 for the format of this field.

XTPRTRYL Number of times IP PrintWay is to retry transmission of the data set. This number was specified in the RETRYL JCL parameter, in the printer definition, or by the Routing exit. IP PrintWay might have set a default value of zero.

XTPRTRYT Period of time IP PrintWay is to wait between retries of the transmission. This time was specified in the RETRYT JCL parameter, in the printer definition, or by the Routing exit. IP PrintWay might have set a default value of zero. See the note on page 196 for the format of this field.

XTPKEY If XTP_ROUTING_KEY_TYPE is set to '1', contains the name of the printer definition. This name was specified in JCL, in the printer definition, or by the Routing exit; the name is padded to the right with blanks. This field is case sensitive.

Fields XTPDEST, XTPCLASS, and XTPFORMS are mapped to the same location as XTPKEY.

XTPDEST If XTP_ROUTING_KEY_TYPE is set to '0', this field contains the value in the DEST field of the printer definition that IP PrintWay uses to print the data set. The name is padded to the right with blanks. If the DEST field in the printer definition is not specified, this field contains blanks.

Note: Use field XTP_JCL_DEST if you want to obtain the value specified in a JCL parameter.

XTPCLASS If XTP_ROUTING_KEY_TYPE is set to '0', this field contains the value in the CLASS field of the printer definition that IP PrintWay uses to print the data set. If the CLASS field in the printer definition is not specified, this field contains blanks.

Note: Use field XTP_JCL_CLASS if you want to obtain the value specified in a JCL parameter.

XTPFORMS If XTP_ROUTING_KEY_TYPE is set to '0', this field contains the value in the FORMS field of the printer definition that IP PrintWay uses to print the data set. The name is padded to the right with blanks. If the FORMS field in the printer definition is not specified, this field contains blanks.

Note: Use field XTP_JCL_FORMS if you want to obtain the value specified in a JCL parameter.

XTPRECFM Record format as indicated in the job file control block (JFCB). Refer to *z/OS MVS Data Areas, Vol 3 (IVT-RCWK)* for a description of the JFCRECFM field in the JFCB.

XTPMRECL Maximum record length

XTPSWBTP Address of the SWBTU for the data set. See "Scheduler Work Block Text Unit (SWBTU)" on page 192 for more information.

XTPSWBTL Length of the SWBTU.

XTPJSPAP Address of the JES job separator page data area for the data set being processed (IAZJSPA). See "JES Job Separator Page Data Area (IAZJSPA)" on page 192 for more information.

XTPCOPYS Number of copies of the data set. This number was specified in JCL, by the Routing exit, or JES provided a default value.

XTPIPORT Port number on the target system. This number was specified in the PORTNO JCL parameter or in the printer definition.

XTP_ADDR_TYPE
This field identifies the transmission protocol and the contents of fields XTPIPADR, XTPPRTQU and XTPIPORT:

Value	Protocol	XTPIPADR	XTPPRTQU	XTPIPORT
0	LPR	IP address	Print queue name	
1	IPP	URL		
2	Direct sockets	IP address		Port number
3	VTAM	LU name	Logmode	
4	e-mail	e-mail addresses		

XTP_ROUTING_KEY_TYPE
An indicator of how IP PrintWay selected the printer definition it will use to obtain printing options:

Value	Meaning
0	IP PrintWay selected the printer definition by matching the DEST, CLASS, and FORMS fields specified in JCL statements with the values in the printer definition.
1	IP PrintWay selected the printer definition using the

printer definition name specified on the OUTPUT JCL statement or specified by NetSpool or Print Interface.

If field XTP_FILTERING = 2, then this field contains 1.

XTP_FILTERING
This field identifies when IP PrintWay processes data two times (as two separate data sets) because the **Resubmit for filtering** option is selected in the printer definition. See "Resubmit for Filtering Considerations for IP PrintWay Exits" on page 190 for more information.

Value	Meaning
0	The **Resubmit for filtering** option is not selected in the printer definition.
2	The **Resubmit for filtering** option is selected and either this is the second data set IP PrintWay processes or the job was originally submitted directly to Print Interface and IP PrintWay processes only one data set. Print Interface has already transformed the input data into the format required by the target printer. For example, if the **afp2pcl** filter is specified in the printer definition, the input data is now in PCL format.

XTP_JCL_DEST
The DEST value with which the output data set was allocated on the JES spool. If NetSpool or Print Interface allocated the data set, then this is the DEST value in the printer definition. Otherwise, it is the DEST value in the OUTPUT JCL statement unless the **Resubmit for filtering** option is selected in the printer definition, in which case this field contains the DEST value specified in the printer definition. The value is padded to the right with blanks.

XTP_JCL_CLASS
The CLASS value with which the output data set was allocated on the JES spool. If NetSpool or Print Interface allocated the data set, then this is the CLASS value in the printer definition. Otherwise, it is the CLASS value specified in JCL statements unless the **Resubmit for filtering** option is selected in the printer definition, in which case this field contains the CLASS value specified in the printer definition. The value is padded to the right with blanks.

XTP_JCL_FORMS
The FORMS value with which the output data set was allocated on the JES spool. If NetSpool or Print Interface allocated the data set, then this is the FORMS value in the printer definition. Otherwise, it is the FORMS value in JCL parameters. The value is padded to the right with blanks.

XTPOPTNS
Transmission options. These options were specified in the printer definition or in the components named in the PRTOPTNS JCL parameter. Refer to the ANFUEXTP macro for the field names. For each transmission option, there are two fields:

- The first field contains the length of the value. A number greater than zero indicates that the second field contains a valid value.
- The second field contains the value of the option.

Output from the Begin Data Set Exit

At return, restore the contents of all registers. Do not set a return code in register 15.

The exit can set the following fields in the ANFUEXTP control block:

XTPWORK1 Address of a work area obtained by this exit. This address is passed to every exit except the Message exit.

XTPWORK2 Address of a second work area obtained by this exit. This address is passed to every exit except the Message exit.

XTPRCFLG Flags to request that IP PrintWay add either the original record or a new record and to specify characteristics of the new record. IP PrintWay sets these flags to B'0' before calling the exit. To transmit the data set without adding a record to the beginning of the data set, do *not* set any of these flags to B'1'.

Flag	Meaning When Set to B'1'
XTPRCTRN	Translate the new record to either ASCII or EBCDIC as specified by the code pages and other translation options in the printer definition.
XTPRCCC	The new record contains a carriage control character in column 1.
XTPRLAST	Do not call this exit again to add another record to the beginning of this data set.
XTPRCEXT	Add the record whose address is in field XTPERPTR.
XTPRCDBL	The new record contains double-byte characters.

XTPMSGFL Flags indicating the destination of a message created by this exit. IP PrintWay sets these flags to B'0' before calling the exit. Set both flags to B'1' to issue the message to both destinations. Place the address of the message in field XTPMSGP and the length of the message in field XTPMSGLN.

Flag	Meaning When Set to B'1'
XTPM2CON	Issue the message to the operator's console.
XTPM2MDS	Issue the message to the IP PrintWay message-log data set.

XTPMSGLN Length of the message whose address is in field XTPMSGP.

XTPMSGP Address of the message created by this exit. IP PrintWay adds a message ID, ANFM155I, to the beginning of this message.

XTPERPTR Address of the record to be added.

XTPERLEN Length of the record to be added.

XTPOPTNS The transmission options. Refer to the ANFUEXTP macro for the field names. For each option, there are two fields:

- The first field contains the length of the value. A number greater than zero indicates that the next field contains a valid value.
- The second field contains the value for the option.

If you add or change a value for one of the options, also set the length in the length field.

Examples

1. To add one record to the beginning of the data set, set the following fields in ANFUEXTP:
   - Set flag XTPRCEXT=1 to request that IP PrintWay add a record.
   - Place the address of the new record in field XTPERPTR.
   - Place the length of the new record in field XTPERLEN.
   - Set flags to indicate the characteristics of the record you are adding:
     - Set flag XTPRCTRN=1 if you want IP PrintWay to translate the record to ASCII or EBCDIC.
     - Set flag XTPRCCC=1 if the record contains a carriage-control character.
     - Set flag XTPRCDBL=1 if the record contains double-byte characters.
   - Set flag XTPRLAST=1 to indicate this is the last record you are adding.
2. To add or change a value for a transmission option, specify these values in the XTPOPTNS field of the ANFUEXTP control block:
   - Specify the value in the field for that transmission option. See the ANFUEXTP macro for the field names and valid values.
   - Specify the length of the value in the corresponding length field. A zero length indicates that the value is not specified.

   For example, to request that the target LPD print a banner (separator) page, set the following fields:
   - Set XTPO_LPR_PRINT_BANNER to 1.
   - Set XTPO_LPR_PRINT_BANNER to the length of XTPO_LPR_PRINT_BANNER.

   If you specify a character string, for example in the XTPO_LPR_BANNER_CLASS field, set the length to the actual length of the character string.
3. To transmit the data set with no changes, simply return control to IP PrintWay.
4. To request that IP PrintWay issue a message to the IP PrintWay message-log data set, do the following:
   - Set XTPM2MDS=1 in the ANFUEXTP control block to write the message to the IP PrintWay message-log data set.
   - Place the address of the message in field XTPMSGP. IP PrintWay adds a message ID, ANFM155I, to the beginning of this message.
   - Place the length of the message in field XTPMSGLN.

Refer also to the sample exits distributed with IP PrintWay and described in “Sample Exits” on page 235.

Record Exit

The Record exit allows you to inspect each record in a data set before the data set is transmitted. You can add one or more records, replace a record, or delete a record. You can also inspect the routing parameters and the transmission options for the data set, but you *cannot* change them.

If you request to add or replace a record, you can specify whether or not (1) IP PrintWay should translate the new record to ASCII, (2) the new record contains a carriage control character, and (3) the new record contains double-byte characters. You can add more than one record at the same location in the data set by requesting that IP PrintWay call the exit multiple times.

You can also create a message for IP PrintWay to issue either to the operator's console or to the IP PrintWay message-log data set. IP PrintWay assigns the same message ID to each message created by an exit: ANFM155I.

You name the Record exit in the printer definition. Therefore, you can write different Record exits for different data sets, provided that the data sets use different printer definitions.

Note: IP PrintWay does not call the Record exit in these situations:

- When a record was added by the Begin Data Set exit or the End Data Set exit.
- When the **Resubmit for filtering** option is selected in the printer definition and IP PrintWay is transmitting the data set to Print Interface for filtering. See "Resubmit for Filtering Considerations for IP PrintWay Exits" on page 190 for more information.

Input to the Record Exit

At entry to the Record exit, the registers contain the following:

Register 1 Pointer to the address of the ANFUEXTP control block. The ANFUEXTP control block contains addresses of the IAZJSPA and the SWBTU control blocks.

Register 13 Address of an 18-word save area for saving the caller's registers.

Register 14 Return address.

Register 15 Entry-point address.

The following fields in ANFUEXTP are valid when the Record exit is called. The format of the ANFUEXTP control block is shown in Figure 34 on page 194.

XTPSAVE A register save area for use by the exit.

XTPWORK1 Address of a work area previously obtained by an exit. This field contains zeroes if a work area has not been obtained.

XTPWORK2 Address of a second work area previously obtained by an exit. This field contains zeroes if a work area has not been obtained.

XTPDSNAM Fully-qualified name of the data set being processed, padded to the right with blanks.

XTPUSRID User ID of the job submitter, padded to the right with blanks, with the following exceptions:

- If NetSpool allocated the data set on the JES spool, this field contains the user ID assigned to the NetSpool started task.
- If Print Interface allocated the data set on the JES spool and the print request was not submitted from an OS/390 or z/OS system, this field contains the user ID from the remote system of the person who submitted the print request

XTPDSFLG Flags containing data set information:

Flag	Meaning When Set to B'1'
XTPDSCC	This data set contains carriage control characters.
XTPDSJNW	This data set is the JESNEWS data set.

XTPRCFLG Flags containing record information:

Flag	Meaning When Set to B'1'

XTPRDFST This is the first call to the exit for this record.

XTPPRPTR Address of the record currently in the data set (original record).

XTPPRLEN Length of record whose address is in field XTPPRPTR.

XTPPQLEN Length of the value in field XTPPRTQU.

XTPIALEN Length of the value in field XTPIPADR.

XTPPRTQU Name of the target print queue specified in the PRTQUEUE JCL parameter, in a printer definition, or by the Routing exit; or the VTAM logmode specified in a printer definition or by the Routing exit. Field XTP_ADDR_TYPE identifies which value is in this field.

XTPIPADR Field XTP_ADDR_TYPE identifies which of the following values this field contains:

- Host name or IP address of the target system specified in the DEST=IP JCL parameter, in a printer definition, or by the Routing exit
- URL of the target system specified in the printer definition or by the Routing exit
- LU name of the target printer specified in the printer definition or by the Routing exit
- E-mail addresses specified in the printer definition or by the Routing exit

XTPRETNS Period of time IP PrintWay is to retain the data set on the JES spool after a successful transmission. This time was specified in the RETAINS JCL parameter, in the printer definition, or by the Routing exit. IP PrintWay might have set a default value of zeroes. See the note on page 196 for the format of this field.

XTPRETNF Period of time IP PrintWay is to retain the data set on the JES spool after all transmission retries have failed. This time was specified in the RETAINF JCL parameter, in the printer definition, or by the Routing exit. IP PrintWay might have set a default value of zeroes. See the note on page 196 for the format of this field.

XTPRTRYL Number of times IP PrintWay is to retry transmission of the data set. This number was specified in the RETRYL JCL parameter, in the printer definition, or by the Routing exit. IP PrintWay might have set a default value of zero.

XTPRTRYT Period of time IP PrintWay is to wait between retries of the transmission. This time was specified in the RETRYT JCL parameter, in the printer definition, or by the Routing exit. IP PrintWay might have set a default value of zero. See the note on page 196 for the format of this field.

XTPKEY If XTP_ROUTING_KEY_TYPE is set to '1, contains the name of the printer definition. This name was specified in JCL, in the printer definition, or by the Routing exit; the name is padded to the right with blanks.

Fields XTPDEST, XTPCLASS, and XTPFORMS are mapped to the same location as XTPKEY.

XTPDEST If XTP_ROUTING_KEY_TYPE is set to '0', this field contains the value in the DEST field of the printer definition that IP PrintWay uses to print the data

set. The name is padded to the right with blanks. If the DEST field in the printer definition is not specified, this field contains blanks.

Note: Use field XTP_JCL_DEST if you want to obtain the value specified in JCL parameters.

XTPCLASS If XTP_ROUTING_KEY_TYPE is set to '0', this field contains the value in the CLASS field of the printer definition that IP PrintWay uses to print the data set. If the CLASS field in the printer definition is not specified, this field contains blanks.

Note: Use field XTP_JCL_CLASS if you want to obtain the value specified in JCL parameters..

XTPFORMS If XTP_ROUTING_KEY_TYPE is set to '0', this field contains the value in the FORMS field of the printer definition that IP PrintWay uses to print the data set. The name is padded to the right with blanks. If the FORMS field in the printer definition is not specified, this field contains blanks.

Note: Use field XTP_JCL_FORMS if you want to obtain the value specified in JCL parameters.

XTPSWBTP Address of the SWBTU for the data set. See "Scheduler Work Block Text Unit (SWBTU)" on page 192 for more information.

XTPSWBTL Length of the SWBTU.

XTPJSPAP Address of the JES job separator page data area for the data set being processed (IAZJSPA). See "JES Job Separator Page Data Area (IAZJSPA)" on page 192 for more information.

XTPIPORT Port number on the target system. This number was specified in the PORTNO JCL parameter or in the printer definition.

XTP_MISC_FLAGS
Flags containing miscellaneous information:

Flag	Meaning When Set to B'1'
XTP_FIRST_REC	This is the first record in the exit.

XTP_ADDR_TYPE
This field identifies the transmission protocol and the contents of fields XTPIPADR, XTPPRTQU and XTPIPORT:

Value	Protocol	XTPIPADR	XTPPRTQU	XTPIPORT
0	LPR	IP address	Print queue name	
1	IPP	URL		
2	Direct sockets	IP address		Port number
3	VTAM	LU name	Logmode	
4	e-mail	e-mail addresses		

XTP_ROUTING_KEY_TYPE
An indicator of how IP PrintWay selected the printer definition it will use to obtain printing options:

Value	Meaning
0	IP PrintWay selected the printer definition by matching the DEST, CLASS, and FORMS fields specified in JCL statements with the values in the printer definition.
1	IP PrintWay selected the printer definition using the printer definition name specified on the OUTPUT JCL statement or specified by NetSpool or Print Interface.

XTP_JCL_DEST
The DEST value with which the output data set was allocated on the JES spool. If NetSpool or Print Interface allocated the data set, then this is the DEST value in the printer definition. Otherwise, it is the DEST value in the OUTPUT JCL statement unless the **resubmit for filtering** option is selected in the printer definition, in which case this field contains the DEST value specified in the printer definition. The value is padded to the right with blanks.

XTP_JCL_CLASS
The CLASS value with which the output data set was allocated on the JES spool. If NetSpool or Print Interface allocated the data set, then this is the CLASS value in the printer definition. Otherwise, it is the CLASS value in JCL statements unless the **resubmit for filtering** option is selected in the printer definition, in which case this field contains the CLASS value specified in the printer definition. The value is padded to the right with blanks.

XTP_JCL_FORMS
The FORMS value with which the output data set was allocated on the JES spool. If NetSpool or Print Interface allocated the data set, then this is the FORMS value in the printer definition. Otherwise, it is the FORMS value in JCL statements. The value is padded to the right with blanks.

XTPOPTNS Transmission options. These options were specified in the printer definition or in the components named in the PRTOPTNS JCL parameter. Refer to the ANFUEXTP macro for the field names. For each transmission option, there are two fields:

- The first field contains the length of the value. A number greater than zero indicates that the second field contains a valid value.
- The second field contains the value of the option.

Output from the Record Exit

At return, restore the contents of all registers. Do not set a return code in register 15.

The exit can set the following fields in the ANFUEXTP control block:

XTPWORK1 Address of a work area obtained by this exit. This address is passed to every exit except the Message exit.

XTPWORK2 Address of a second work area obtained by this exit. This address is passed to every exit except the Message exit.

XTPRCFLG Flags to request that IP PrintWay add a record and to specify characteristics of a new record. IP PrintWay sets these flags to B'0' before calling the exit.

Flag	Meaning When Set to B'1'
XTPRCTRN	Translate the new record to either ASCII or EBCDIC as specified by the code pages and other translation options in the printer definition. This flag does *not* apply to the original record.
XTPRCCC	The new record contains a carriage control character in column 1. This flag does *not* apply to the original record.
XTPRCORG	Add the original record to the data set.
XTPRLAST	Do *not* call this exit again for this record. If neither field XTPRCORG nor XTPRCEXT are set, IP PrintWay ignores this flag and does *not* call the exit again for this record.
XTPRCEXT	Add the new record whose address is in field XTPERPTR. Do *not* set this flag and flag XTPRCORG at the same time. If flag XTPRCORG is set, IP PrintWay ignores this flag.
XTPRCDBL	The new record contains double-byte characters. This flag does *not* apply to the original record.

XTPMSGFL Flags indicating the destination of a message created by this exit. Set both flags to B'1' to issue the message to both destinations. Place the address of the message in field XTPMSGP and the length of the message in field XTPMSGLN.

Flag	Meaning When Set to B'1'
XTPM2CON	Issue the message to the operator's console.
XTPM2MDS	Issue the message to the IP PrintWay message-log data set.

XTPERPTR Address of the record to be added.

XTPERLEN Length of the record to be added.

XTPPRPTR Address of the original record in the data set. You can change the contents of the original record.

XTPPRLEN Length of the original record whose address is in field XTPPRPTR.

XTPMSGLN Length of the message whose address is in field XTPMSGP.

XTPMSGP Address of the message created by this exit. IP PrintWay adds a message ID, ANFM155I, to the beginning of this message.

Examples

1. To add a new record after the original record in the data set, set the following fields in ANFUEXTP:
   - Set flag XTPRCORG=1 to request that IP PrintWay transmit the original record.
   - Set flag XTPRLAST=0 to indicate that IP PrintWay is to call the exit again to add another record.

On the next call to the Record Exit, set the following fields in ANFUEXTP:

- Set flag XTPRCORG=0 to request that IP PrintWay not add the original record.
- Set flag XTPRCEXT=1 to request that IP PrintWay add a new record.
- Place the address of the new record in field XTPERPTR.
- Place the length of the new record in field XTPERLEN.
- Set flags to indicate the characteristics of the new record:
  - Set flag XTPRCTRN=1 if you want IP PrintWay to translate the record to ASCII or EBCDIC.
  - Set flag XTPRCCC=1 if the record contains a carriage-control character.
  - Set flag XTPRCDBL=1 if the record contains double-byte characters.
- Set flag XTPRLAST=1 to indicate this is the last record you are adding.

2. To replace the original record, set the following fields in ANFUEXTP:
   - Set flag XTPRCORG=1 to request that IP PrintWay add the original record.
   - Modify the record whose address is in field XTPPRPTR; however, do not increase the length of the record in the original location. To make the record longer, create the record in another location and place the address of the record in field XTPPRPTR.
   - If the length of the record has changed, place the new record length in XTPPRLEN.
   - Set flag XTPRLAST=1 to indicate this is the last record to be added.
3. To make *no* changes to the original record:
   - Set flag XTPRCORG=1 to request that IP PrintWay add the original record.
   - Set flag XTPRLAST=1 to indicate this is the last record to be added.
4. To delete the original record, simply return control to IP PrintWay.
5. To request that IP PrintWay issue a message to the IP PrintWay message-log data set, do the following:
   - Set XTPM2MDS=1 in the ANFUEXTP control block to write the message to the IP PrintWay message-log data set.
   - Place the address of the message in field XTPMSGP. IP PrintWay adds a message ID, ANFM155I, to the beginning of this message.
   - Place the length of the message in field XTPMSGLN.

Refer also to the sample exits distributed with IP PrintWay and described in “Sample Exits” on page 235.

End Data Set Exit

The End Data Set exit lets you to add one or more records to the end of a data set. You can also inspect the routing parameters and the transmission options for the data set, but you *cannot* change them.

If you add a record, you can specify whether or not (1) IP PrintWay should translate the new record to ASCII, (2) the new record contains a carriage control character, and (3) the new record contains double-byte characters. To add more than one record, you must request that IP PrintWay call the exit multiple times.

You can also create a message for IP PrintWay to issue either to the operator’s console or to the IP PrintWay message-log data set. IP PrintWay assigns the same message ID to each message created by an exit: ANFM155I.

The End Data Set exit can use the Block Letter program (ANFUBLK), provided with IP PrintWay, to build block letters suitable for printing on a separator page. See "Using the Block Letter Program (ANFUBLK)" on page 234 for more information.

You name the End Data Set exit in the printer definition. Therefore, you can write different End Data Set exits for different data sets, provided that the data sets use different options entries.

You might want to write an End Data Set exit to perform one of the following functions:

- Append fonts to be transmitted to the printer.
- Add a custom separator page. On the separator page you can print job and data set information contained in the JES job separator page data area (IAZJSPA) or the scheduler work block text unit (SWBTU). See "JES Job Separator Page Data Area (IAZJSPA)" on page 192 and "Scheduler Work Block Text Unit (SWBTU)" on page 192 for more information about these system control blocks.

Notes:

1. IP PrintWay does *not* call the End Data Set exit when the **Resubmit for filtering** option is selected in the printer definition and IP PrintWay is transmitting the data set to Print Interface for filtering. See "Resubmit for Filtering Considerations for IP PrintWay Exits" on page 190 for more information.
2. IP PrintWay adds printer instructions specified in the **Document trailer** field of the printer definition to the data set *before* it calls the End Data Set exit. Therefore, if the printer instructions in the **Document trailer** field reset the printer to its original status, printer instructions specified in the **Document header** field do not apply to records added by the End Data Set exit. For example, if the **Document header** field contains printer instructions for landscape printing and the **Document trailer** field contains printer instructions to reset the printer (to portrait orientation), a separator page created by the End Data Set exit does *not* print in the landscape direction. If you want your separator page to print in the landscape direction, you must add printer instructions for landscape printing in the End Data Set exit.

Input to the End Data Set Exit

At entry to the End Data Set exit, the registers contain the following:

Register 1 Pointer to the address of the ANFUEXTP control block. The ANFUEXTP control block contains addresses of the IAZJSPA and the SWBTU control blocks.

Register 13 Address of an 18-word save area for saving the caller's registers.

Register 14 Return address.

Register 15 Entry-point address.

The following fields in ANFUEXTP are valid when the End Data Set exit is called. The format of the ANFUEXTP control block is shown in Figure 34 on page 194.

XTPSAVE A register save area for use by the exit.

XTPWORK1 Address of a work area previously obtained by an exit. This field contains zeroes if a work area has not been obtained.

XTPWORK2 Address of a second work area previously obtained by an exit. This field contains zeroes if a work area has not been obtained.

XTPDSNAM Fully qualified name of the data set being processed, padded to the right with blanks.

XTPUSRID User ID of the job submitter, padded to the right with blanks, with the following exceptions:

- If NetSpool allocated the data set on the JES spool, this field contains the user ID assigned to the NetSpool started task.
- If Print Interface allocated the data set on the JES spool and the print request was not submitted from an OS/390 or z/OS system, this field contains the user ID from the remote system of the person who submitted the print request

XTPDSFLG Flags containing data set information:

Flag	Meaning When Set to B'1'
XTPDSCC	This data set contains carriage control characters.
XTPDSJOB	This is the first data set in the job.
XTPDSJNW	This data set is the JESNEWS data set.
XTPDSEJB	This is the last data set in the job.
XTPDSCON	This data set is part of a concatenation.

XTPRCFLG Flags containing data set information:

Flag	Meaning When Set to B'1'
XTPRDFST	This is the first call to the exit for this data set.

XTPPQLEN Length of the value in field XTPPRTQU.

XTPIALEN Length of the value in field XTPIPADR.

XTPPRTQU Name of the target print queue specified in the PRTQUEUE JCL parameter, in a printer definition, or by the Routing exit; or the VTAM logmode specified in a printer definition or by the Routing exit. Field XTP_ADDR_TYPE identifies which value is in this field.

XTPIPADR Field XTP_ADDR_TYPE identifies which of the following values this field contains:

- Host name or IP address of the target system specified in the DEST=IP JCL parameter, in a printer definition, or by the Routing exit
- URL of the target system specified in the printer definition or by the Routing exit
- LU name of the target printer specified in the printer definition or by the Routing exit
- E-mail addresses specified in the printer definition or by the Routing exit

XTPRETNS Period of time IP PrintWay is to retain the data set on the JES spool after a successful transmission. This time was specified in the RETAINS JCL parameter, in the printer definition, or by the Routing exit. IP PrintWay might have set a default value of zeroes. See the note on page 196 for the format of this field.

XTPRETNF Period of time IP PrintWay is to retain the data set on the JES spool after all transmission retries have failed. This time was specified in the RETAINF JCL parameter, in the printer definition, or by the Routing exit. IP PrintWay may have set a default value of zeroes. See the note on page 196 for the format of this field.

XTPRTRYL Number of times IP PrintWay is to retry transmission of the data

set. This time was specified in the RETRYL JCL parameter, in the printer definition, or by the Routing exit. IP PrintWay might have set a default value of zero.

XTPRTRYT Period of time IP PrintWay is to wait between retries of the transmission. This time was specified in the RETRYT JCL parameter, in the printer definition, or by the Routing exit. IP PrintWay may have set a default value of zero. See the note on page 196 for the format of this field.

XTPKEY If XTP_ROUTING_KEY_TYPE is set to '1, contains the name of the printer definition. This name was specified in JCL, in the printer definition, or by the Routing exit; the name is padded to the right with blanks.

Fields XTPDEST, XTPCLASS, and XTPFORMS are mapped to the same location as XTPKEY.

XTPDEST If XTP_ROUTING_KEY_TYPE is set to '0', this field contains the value in the DEST field of the printer definition that IP PrintWay uses to print the data set. The name is padded to the right with blanks. If the DEST field in the printer definition is not specified, this field contains blanks.

Note: Use field XTP_JCL_DEST if you want to obtain the value specified on the OUTPUT JCL statement.

XTPCLASS If XTP_ROUTING_KEY_TYPE is set to '0', this field contains the value in the CLASS field of the printer definition that IP PrintWay uses to print the data set. If the CLASS field in the printer definition is not specified, this field contains blanks.

Note: Use field XTP_JCL_CLASS if you want to obtain the value specified in JCL statements.

XTPFORMS If XTP_ROUTING_KEY_TYPE is set to '0', this field contains the value in the FORMS field of the printer definition that IP PrintWay uses to print the data set. The name is padded to the right with blanks. If the FORMS field in the printer definition is not specified, this field contains blanks.

Note: Use field XTP_JCL_FORMS if you want to obtain the value specified in JCL statements.

XTPRECFM Record format as indicated in the job file control block (JFCB). Refer to *z/OS MVS Data Areas, Vol 3 (IVT-RCWK)* for a description of the JFCRECFM field in the JFCB.

XTPMRECL Maximum record length

XTPSWBTP Address of the SWBTU for the data set. See "Scheduler Work Block Text Unit (SWBTU)" on page 192 for more information.

XTPSWBTL Length of the SWBTU.

XTPJSPAP Address of the JES job separator page data area for the data set

being processed (IAZJSPA). See “JES Job Separator Page Data Area (IAZJSPA)” on page 192 for more information.

XTPCOPYS Number of copies of the data set. This number was specified in JCL, by the Routing exit, or JES provided a default value.

XTPIPORT Port number on the target system. This number was specified in the PORTNO JCL parameter or in the printer definition.

XTP_FILTERING
This field identifies when IP PrintWay processes data two times (as two separate data sets) because the **Resubmit for filtering** option is selected in the printer definition. See “Resubmit for Filtering Considerations for IP PrintWay Exits” on page 190 for more information.

Value	Meaning
0	The **Resubmit for filtering** option is not selected in the printer definition.
2	The **Resubmit for filtering** option is selected and either this is the second data set IP PrintWay processes or the job was originally submitted directly to Print Interface and IP PrintWay processes only one data set. Print Interface has already transformed the input data into the format required by the target printer. For example, if the **afp2pcl** filter is specified in the printer definition, the input data is now in PCL format.

XTP_JCL_DEST
The DEST value with which the output data set was allocated on the JES spool. If NetSpool or Print Interface allocated the data set, then this is the DEST value in the printer definition. Otherwise, it is the DEST value in the OUTPUT JCL statement unless the **Resubmit for filtering** option is selected in the printer definition, in which case this field contains the DEST value specified in the printer definition. The value is padded to the right with blanks.

XTP_JCL_CLASS
The CLASS value with which the output data set was allocated on the JES spool. If NetSpool or Print Interface allocated the data set, then this is the CLASS value in the printer definition. Otherwise, it is the CLASS value in JCL statements unless the **Resubmit for filtering** option is selected in the printer definition, in which case this field contains the CLASS value specified in the printer definition. The value is padded to the right with blanks.

XTP_JCL_FORMS
The FORMS value with which the output data set was allocated on the JES spool. If NetSpool or Print Interface allocated the data set, then this is the FORMS value in the printer definition. Otherwise, it is the FORMS value in a JCL statement. The value is padded to the right with blanks.

XTPOPTNS Transmission options. These options were specified in the printer definition or in the components named in the PRTOPTNS JCL parameter. Refer to the ANFUEXTP macro for the field names. For each transmission option, there are two fields:

- The first field contains the length of the value. A number greater than zero indicates that the second field contains a valid value.

- The second field contains the value of the option.

If you change a value for one of the options, also set the length of the value in the length field that precedes that field.

Output from the End Data Set Exit

At return, restore the contents of all registers. Do not set a return code in register 15.

The exit can set the following fields in the ANFUEXTP control block:

XTPWORK1 Address of a work area obtained by this exit. This address is passed to every exit except the Message exit.

XTPWORK2 Address of a second work area obtained by this exit. This address is passed to every exit except the Message exit.

XTPRCFLG Flags to request that IP PrintWay add a record and to specify characteristics of the new record. IP PrintWay sets these flags to B'0' before calling the exit. To transmit the data set without adding a record to the end of the data set, do *not* set any of these flags.

Flag	Meaning When Set to B'1'
XTPRCTRN	Translate the new record to either ASCII or EBCDIC as specified by the code pages and other translation options in the printer definition.
XTPRCCC	The new record contains a carriage control character in column 1.
XTPRLAST	Do not call this exit again to add another record to the end of this data set.
XTPRCEXT	Add the record whose address is in field XTPERPTR.
XTPRCDBL	The new record contains double-byte characters.

XTPMSGFL Flags indicating the destination of a message created by this exit. IP PrintWay sets these flags to B'0' before calling the exit. Set both flags to B'1' to issue the message to both destinations. Place the address of the message in field XTPMSGP and the length of the message in field XTPMSGLN.

Flag	Meaning When Set to B'1'
XTPM2CON	Issue the message to the operator's console.
XTPM2MDS	Issue the message to the IP PrintWay message-log data set.

XTPERPTR Address of the record to be added.

XTPERLEN Length of the record to be added.

XTPMSGLN Length of the message whose address is in field XTPMSGP.

XTPMSGP Address of the message created by this exit. IP PrintWay adds a message ID, ANFM155I, to the beginning of this message.

Examples

1. To add one record to the end of the data set, set the following fields in ANFUEXTP:
   - Set flag XTPRCEXT=1 to request that IP PrintWay add a record.

- Place the address of the new record in field XTPERPTR.
- Place the length of the new record in field XTPERLEN.
- Set flags to indicate the characteristics of the record you are adding:
  - Set flag XTPRCTRN=1 if you want IP PrintWay to translate the record to ASCII or EBCDIC.
  - Set flag XTPRCCC=1 if the record contains a carriage-control character.
  - Set flag XTPRCDBL=1 if the record contains double-byte characters.
- Set flag XTPRLAST=1 to indicate this is the last record you are adding.

2. To transmit the data set with no changes, simply return control to IP PrintWay.
3. To request that IP PrintWay issue a message to the IP PrintWay message-log data set, do the following:
   - Set XTPM2MDS=1 in the ANFUEXTP control block to write the message to the IP PrintWay message-log data set.
   - Place the address of the message in field XTPMSGP. IP PrintWay adds a message ID, ANFM155I, to the beginning of this message.
   - Place the length of the message in field XTPMSGLN.

Refer also to the sample exits distributed with IP PrintWay and described in "Sample Exits" on page 235.

SMF Exit (ANFUXSMF)

The SMF exit allows you to replace or suppress the system management facilities (SMF) type-6 records written by IP PrintWay. SMF is an optional control program that collects information to assist you in evaluating system use. If you do not write an SMF exit, IP PrintWay writes a standard SMF type-6 record for each data set transmitted to the target print queue.

For more information about the SMF type-6 record that IP PrintWay writes, refer to *z/OS Infoprint Server Operation and Administration*. For more information about SMF and about macro IFASMFR, which maps the SMF type 6-record, refer to *z/OS MVS System Management Facilities (SMF)*.

You can also request that IP PrintWay issue a message produced in the exit to the operator's console or to the IP PrintWay message-log data set. IP PrintWay assigns the same message ID to all messages produced by any exit: ANFM155I.

IP PrintWay calls the same SMF exit for all data sets. You can install different SMF exits for different IP PrintWay FSSs. See "Installing Exits" on page 236 for information.

You might want to write an SMF exit for these reasons:

- To include additional information in the SMF type-6 record.
- To suppress the SMF record for JESNEWS data sets. Field XTPDSJNW indicates whether or not this is a JESNEWS data set.
- To suppress the first SMF record that IP PrintWay writes for a data set when the **Resubmit for filtering** field is selected in the printer definition. IP PrintWay writes one SMF record when it transmits the data set to Print Interface for filtering (that is, to transform data); this record contains the IP address of the z/OS system. IP PrintWay writes a second SMF record when it transmits the data set to the target printer; this record contains the IP address of the target printer.

For more information, see the description of field XTP_FILTERING in “Input to the SMF Exit” and see “Resubmit for Filtering Considerations for IP PrintWay Exits” on page 190. Also, see “Examples” on page 226 for an example of how to suppress the first SMF record.

Note: Fields in macro IFASMFR that are provided by SVC 83 are not available to this exit.

Input to the SMF Exit

At entry to the SMF exit, the registers contain the following:

Register 1 Pointer to the address of the ANFUEXTP control block. The ANFUEXTP control block contains addresses of the IAZJSPA and the SWBTU control blocks.

Register 13 Address of an 18-word save area for saving the caller’s registers.

Register 14 Return address.

Register 15 Entry-point address.

The following fields in ANFUEXTP are valid when the SMF exit is called. The format of the ANFUEXTP control block is shown in Figure 34 on page 194.

XTPSAVE A register save area for use by the exit.

XTPWORK1 Address of a work area previously obtained by an exit. This field contains zeroes if a work area has not been obtained.

XTPWORK2 Address of a second work area previously obtained by an exit. This field contains zeroes if a work area has not been obtained.

XTPDSNAM Fully qualified name of the data set being processed, padded to the right with blanks.

XTPUSRID User ID of the job submitter, padded to the right with blanks, with the following exceptions:

- If NetSpool allocated the data set on the JES spool, this field contains the user ID assigned to the NetSpool started task.
- If Print Interface allocated the data set on the JES spool and the print request was not submitted from an OS/390 or z/OS system, this field contains the user ID from the remote system of the person who submitted the print request

XTPDSFLG Flags containing data set information:

Flag	Meaning When Set to B’1’
XTPDSJOB	This is the first data set in the job.
XTPDSJNW	This data set is the JESNEWS data set.
XTPDSEJB	This is the last data set in the job.
XTPDSCON	This data set is part of a concatenation.

XTPPRPTR Address of the SMF record for the data set.

XTPPRLEN Length of the SMF record for the data set.

XTPKEY If XTP_ROUTING_KEY_TYPE is set to ’1, contains the name of the printer definition. This name was specified in JCL, in the printer definition, or by the Routing exit; the name is padded to the right with blanks.

Fields XTPDEST, XTPCLASS, and XTPFORMS are mapped to the same location as XTPKEY.

XTPDEST If XTP_ROUTING_KEY_TYPE is set to '0', this field contains the value in the DEST field of the printer definition that IP PrintWay uses to print the data set. The name is padded to the right with blanks. If the DEST field in the printer definition is not specified, this field contains blanks.

Note: Use field XTP_JCL_DEST if you want to obtain the value specified on the OUTPUT JCL statement.

XTPCLASS If XTP_ROUTING_KEY_TYPE is set to '0', this field contains the value in the CLASS field of the printer definition that IP PrintWay uses to print the data set. If the CLASS field in the printer definition is not specified, this field contains blanks.

Note: Use field XTP_JCL_CLASS if you want to obtain the value specified in a JCL statements.

XTPFORMS If XTP_ROUTING_KEY_TYPE is set to '0', this field contains the value in the FORMS field of the printer definition that IP PrintWay uses to print the data set. The name is padded to the right with blanks. If the FORMS field in the printer definition is not specified, this field contains blanks.

Note: Use field XTP_JCL_FORMS if you want to obtain the value specified in a JCL statement.

XTPSWBTP Address of the SWBTU for the data set. See "Scheduler Work Block Text Unit (SWBTU)" on page 192 for more information.

XTPSWBTL Length of the SWBTU.

XTPJSPAP Address of the JES job separator page data area for the data set being processed (IAZJSPA). See "JES Job Separator Page Data Area (IAZJSPA)" on page 192 for more information.

XTPCOPYS Number of copies of the data set. This number was specified in JCL, by the Routing exit, or JES provided a default value.

XTP_ROUTING_KEY_TYPE
An indicator of how IP PrintWay selected the printer definition it will use to obtain printing options:

Value	Meaning
0	IP PrintWay selected the printer definition by matching the DEST, CLASS, and FORMS fields specified on JCL statements with the values in the printer definition.
1	IP PrintWay selected the printer definition using the printer definition name specified on the OUTPUT JCL statement or specified by NetSpool or Print Interface.

If field XTP_FILTERING = 2, then this field contains 1.

XTP_FILTERING

This field identifies when IP PrintWay processes data two times (as two separate data sets) because the **Resubmit for filtering** option is selected in the printer definition. See "Resubmit for Filtering Considerations for IP PrintWay Exits" on page 190 for more information.

Value	Meaning
0	The **Resubmit for filtering** option is not selected in the printer definition.
1	The **Resubmit for filtering** option is selected and this is the first data set. IP PrintWay is transmitting this data set to Print Interface for filtering. The SMF record contains the IP address of the z/OS system.
2	The **Resubmit for filtering** option is selected and either this is the second data set IP PrintWay processed or the job was originally submitted directly to Print Interface and IP PrintWay processed the data set only one time. IP PrintWay is transmitting this data set to the target printer. The SMF record contains the IP address of the target printer.

XTP_JCL_DEST

The DEST value with which the output data set was allocated on the JES spool. If NetSpool or Print Interface allocated the data set, then this is the DEST value in the printer definition. Otherwise, it is the DEST value in the OUTPUT JCL statement unless XTP_FILTERING=2, in which case this field contains the DEST value specified in the printer definition. The value is padded to the right with blanks.

XTP_JCL_CLASS

The CLASS value with which the output data set was allocated on the JES spool. If NetSpool or Print Interface allocated the data set, then this is the CLASS value in the printer definition. Otherwise, it is the CLASS value specified in a JCL statement unless XTP_FILTERING=2, in which case this field contains the CLASS value specified in the printer definition. The value is padded to the right with blanks.

XTP_JCL_FORMS

The FORMS value with which the output data set was allocated on the JES spool. If NetSpool or Print Interface allocated the data set, then this is the FORMS value in the printer definition. Otherwise, it is the FORMS value in a JCL statement. The value is padded to the right with blanks.

Output from the SMF Exit

At return, restore the contents of all registers. Do not set a return code in register 15.

The exit can set the following fields in the ANFUEXTP control block:

XTPWORK1 Address of a work area obtained by this exit. This address is passed to every exit except the Message exit.

XTPWORK2 Address of a second work area obtained by this exit. This address is passed to every exit except the Message exit.

XTPRCFLG Flag to request that IP PrintWay write or suppress the SMF record. IP PrintWay sets this flag to B'0' before calling the exit.

Flag	Meaning When Set to B'1'
XTPRCORG	Write the SMF record; otherwise, IP PrintWay suppresses the record.

XTPMSGFL Flags indicating the destination of a message created by this exit. IP PrintWay sets these flags to B'0' before calling the exit. Set both flags to B'1' to issue the message to both destinations. Place the address of the message in field XTPMSGP and the length of the message in field XTPMSGLN.

Flag	Meaning When Set to B'1'
XTPM2CON	Issue the message to the operator's console.
XTPM2MDS	Issue the message to the IP PrintWay message-log data set.

XTPPRPTR Address of the SMF type-6 record to be written for the data set. You can change the contents and length of the original record.

XTPPRLEN Length of the SMF type-6 record whose address is in field XTPPRPTR.

XTPMSGLN Length of the message whose address is in field XTPMSGP.

XTPMSGP Address of the message created by this exit. IP PrintWay adds a message ID, ANFM155I, to the beginning of this message.

Examples

1. To replace the SMF record, set the following fields in ANFUEXTP:
   - Set flag XTPRCORG=1 to request that IP PrintWay transmit the original record.
   - Modify the SMF record whose address is in field XTPPRPTR.
   - If the length of the SMF record has changed, place the new length in field XTPPRLEN.
2. To write the original SMF record, with no changes, set the following field in ANFUEXTP:
   - Set flag XTPRCORG=1 to request that IP PrintWay write the original SMF record.
3. To suppress the SMF record, simply return control to IP PrintWay.
4. To suppress the first of two SMF records IP PrintWay writes when the **Resubmit for filtering** field is selected in the printer definition:
   - If XTP_FILTERING=1, then return control to IP PrintWay. This action causes IP PrintWay to suppress the first SMF record.
   - If XTP_FILTERING=2, then set XTPRCORG=1. This action causes IP PrintWay to write the second SMF record.
5. To request that IP PrintWay issue a message to the IP PrintWay message-log data set, do the following:
   - Set XTPM2MDS=1 in the ANFUEXTP control block to write the message to the IP PrintWay message-log data set.

- Place the address of the message in field XTPMSGP. IP PrintWay adds a message ID, ANFM155I, to the beginning of this message.
- Place the length of the message in field XTPMSGLN.

Refer also to the sample exits distributed with IP PrintWay and described in "Sample Exits" on page 235.

Message Exit (ANFUXMSG)

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 IP PrintWay message-log data set, including messages created by another IP PrintWay exit.

IP PrintWay calls the same Message exit for all data sets. You can install different Message exits for different IP PrintWay FSSs. See "Installing Exits" on page 236 for information.

Input to the Message Exit

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.

The following fields in ANFUEXTP are valid when the Message exit is called. The format of the ANFUEXTP control block is shown in Figure 34 on page 194.

XTPSAVE A register save area for use by the exit.

XTPMSGLN Length of the message whose address is in field XTPMSGP.

XTPMSGP Address of the message to be issued. This message has a message ID, followed by one blank and the message text.

XTP_JCL_DEST
The DEST value with which the output data set was allocated on the JES spool. If NetSpool or Print Interface allocated the data set, then this is the DEST value in the printer definition. Otherwise, it is the DEST value in the OUTPUT JCL statement unless the **Resubmit for filtering** option is selected in the printer definition, in which case this field contains the DEST value specified in the printer definition. The value is padded to the right with blanks.

XTP_JCL_CLASS
The CLASS value with which the output data set was allocated on the JES spool. If NetSpool or Print Interface allocated the data set, then this is the CLASS value in the printer definition. Otherwise, it is the CLASS value specified in a JCL statement unless the **Resubmit for filtering** option is selected in the printer definition, in which case this field contains the CLASS value specified in the printer definition. The value is padded to the right with blanks.

XTP_JCL_FORMS
The FORMS value with which the output data set was allocated on the JES spool. If NetSpool or Print Interface allocated the data set,

then this is the FORMS value in the printer definition. Otherwise, it is the FORMS value in a JCL statement. The value is padded to the right with blanks.

Output from the Message Exit

At return, restore the contents of all registers. Do not set a return code in register 15.

The exit can set the following fields in the ANFUEXTP control block:

XTPMSGFL Flag indicating whether or not IP PrintWay is to issue the message. IP PrintWay sets this flag to B'0' before calling the exit.

Flag	Meaning When Set to B'1'
XTPM2MDS	Issue the message to the IP PrintWay message-log data set.

XTPMSGLN Length of the message whose address is in field XTPMSGP.

XTPMSGP Address of the message to be issued. This message should have a message ID, followed by the message text.

Examples

1. To write the message unchanged to the IP PrintWay message-log data set, set the following fields in ANFUEXTP:
   - Set flag XTPM2MDS=1 to request that IP PrintWay write the message to the IP PrintWay message-log data set.
2. To suppress the message, simply return control to IP PrintWay.
3. To replace the message, set the following fields in ANFUEXTP:
   - Set XTPM2MDS=1 in the ANFUEXTP control block to write the message to the IP PrintWay message-log data set.
   - Modify the message whose address is in field XTPMSGP. Alternatively, place the address of another message in field XTPMSGP.
   - If the length of the message has changed, place the new length in field XTPMSGLN.

Refer also to the sample exits distributed with IP PrintWay and described in "Sample Exits" on page 235.

Response Notification Exit (ANFUXRSP)

The Response Notification Exit allows you to take action, based on the success or failure of the transmission, as indicated by the response notification code. Such actions might include the generation of user-defined messages for either the operator's console or the IP PrintWay message-log data set, or both. Actions outside of IP PrintWay also can be initiated, based on the success of the transmission, by calling modules that are external to IP PrintWay.

IP PrintWay calls the same response exit for all data set transmissions. You can install different response exits for different IP PrintWay FSSs. See "Installing Exits" on page 236 for information.

Input to the Response Notification Exit

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.

The following input fields in ANFUEXTP are valid when the Response Notification exit is called. The format of the ANFUEXTP control block is shown in Figure 34 on page 194.

XTPSAVE A register save area for use by this exit.

XTPWORK1 Address of a work area previously obtained by an exit. This field contains zeros if this work area has not been obtained.

XTPWORK2 Address of a second work area previously obtained by an exit. This field contains zeros if this work area has not been obtained.

XTPDSNAM Fully qualified name of the data set being processed, padded to the right with blanks.

XTPUSRID User ID of the job submitter, padded to the right with blanks, with the following exceptions:

- If NetSpool allocated the data set on the JES spool, this field contains the user ID assigned to the NetSpool started task.
- If Print Interface allocated the data set on the JES spool and the print request was not submitted from an OS/390 or z/OS system, this field contains the user ID from the remote system of the person who submitted the print request

XTPSRCFL Flags indicating whether a parameter was specified in a JCL parameter.

If the flag is set to 1, the indicated parameter was specified in a JCL parameter.

If the flag is set to 0, the JCL parameter was not specified. Instead, IP PrintWay obtained the value from a printer definition. If the **DEST=***IP parameter* was specified in JCL, IP PrintWay obtained the value from the PrintWay default printer definition. If no PrintWay default printer definition exists, IP PrintWay set a default value.

Flag Meaning when set to 1.

XTPOPTS The PRTOPTNS parameter specified the name of components in the Printer Inventory.

XTPSIPAD The **DEST=***IP parameter* specified the host name or IP address of the target system.

XTPSPRTQ The PRTQUEUE parameter specified the name of the target print queue.

XTPSRTNS The RETAINS parameter specified the retention time after a successful transmission.

XTPSRTNF The RETAINF parameter specified the retention time after a failed transmission.

XTPSRTYL The RETYLF parameter specified the retry limit.

XTPSRTYT The RETYLT parameter specified the retry time.

XTPDSFLG Flags containing data set information.

Flag **Meaning When Set to B'1'**

XTPMDSCC This data set contains carriage control characters.

XTPDSJOB This is the first data set in the job.

XTPDSJNW This data set is the JESNEWS data set.

XTPDSEJB This is the last data set in the job.

XTPDSCON This data set is part of a concatenation.

XTPPQLEN Length of the value in XTPPRTQU.

XTPIALEN Length of the value in XTPIPADR.

XTPPRTQU Field XTP_ADDR_TYPE identifies which value is in this field:

- Name of the print queue specified in the PRTQUEUE JCL parameter, in the printer definition, or by the Routing exit.
- Name of the VTAM logmode specified in a printer definition or by the Routing exit.

Or, the XTPPRTQU might not be available to the Response Notification exit if the Response code is X'08'.

XTPIPADR Field XTP_ADDR_TYPE identifies which of the following values this field contains:

- Host name or IP address of the target system specified in the DEST=IP JCL parameter, in the printer definition, or by the Routing exit
- URL of the target system specified in the printer definition or by the Routing exit
- LU name of the target printer specified in the printer definition or by the Routing exit
- E-mail addresses specified in the printer definition or by the Routing exit

XTPRETNS Period of time IP PrintWay is to retain the data set on the JES spool after a successful transmission. This time was specified in the RETAINS JCL parameter or in the printer definition. IP PrintWay might have set a default value of zeros. See the note on page 196 for the format of this field. XTPRETNS may not be available to the Response Notification exit if the Response code is X'08'.

XTPRETNF Period of time IP PrintWay is to retain the data set on the JES spool after an **all transmission retries** has failed. This time was specified in the RETAINF JCL parameter or in the printer definition. IP PrintWay may have set a default value of zeros. See the note on page 196 for the format of this field. XTPRETNF may not be available to the Response Notification exit if the Response code is X'08'.

XTPRTRYL The number of times IP PrintWay is to retry transmission of the data set. This time was specified in the RETRYL JCL parameter or in the printer definition. IP PrintWay might have set a default value of zeros. XTPRTRYL may not be available to the Response Notification exit if the Response code is X'08'.

XTPRTRYT The period of time IP PrintWay is to wait between retries of the transmission. This time was specified in the RETRYT JCL parameter or in the printer definition. IP PrintWay may have set a default value of zeros. See the note on page 196 for the format of

this field. XTPRTRYT may not be available to the Response Notification exit if the Response code is X'08'.

XTPKEY If XTP_ROUTING_KEY_TYPE is set to ’1, contains the name of the printer definition. This name was specified in JCL, in the printer definition, or by the Routing exit; the name is padded to the right with blanks.

Fields XTPDEST, XTPCLASS, and XTPFORMS are mapped to the same location as XTPKEY.

XTPDEST If XTP_ROUTING_KEY_TYPE is set to ’0’, this field contains the value in the DEST field of the printer definition that IP PrintWay uses to print the data set. The name is padded to the right with blanks. If the DEST field in the printer definition is not specified, this field contains blanks.

Note: Use field XTP_JCL_DEST if you want to obtain the value specified on the OUTPUT JCL statement.

XTPCLASS If XTP_ROUTING_KEY_TYPE is set to ’0’, this field contains the value in the CLASS field of the printer definition that IP PrintWay uses to print the data set. If the CLASS field in the printer definition is not specified, this field contains blanks.

Note: Use field XTP_JCL_CLASS if you want to obtain the value specified in a JCL statement.

XTPFORMS If XTP_ROUTING_KEY_TYPE is set to ’0’, this field contains the value in the FORMS field of the printer definition that IP PrintWay uses to print the data set. The name is padded to the right with blanks. If the FORMS field in the printer definition is not specified, this field contains blanks.

Note: Use field XTP_JCL_FORMS if you want to obtain the value specified in a JCL statement.

XTPRECFM The record format, as indicated in the job file control block (JFCB). Refer to *z/OS MVS Data Areas, Vol 3 (IVT-RCWK)* for a description of the JFCRECFM field in the JFCB.

XTPMRECL Maximum record length.

XTPSWBTP Address of the SWBTU for the data set. See “Scheduler Work Block Text Unit (SWBTU)” on page 192 for more information.

XTPSWBTL Length of the SWBTU area.

XTPJSPAP Address of the JES job separator page data area for the data set being processed (IAZJSPA). See “JES Job Separator Page Data Area (IAZJSPA)” on page 192 for more information.

XTPCOPYS Number of copies of the data set. This number was specified in JCL or JES provided a default value.

XTPRSPCD The Response Notification code.

The response notification code is stored in XTPRSPCD. It can contain any of the following values:

Value	Meaning
00	PrintWay completed a successful transmission. After the successful retain time expires, the output data set is removed from the JES spool.
04	The transmission was unsuccessful.
08	The initial processing of the data set by PrintWay was unsuccessful.
12	The transmission was unsuccessful. The retry limit is now 0. After the failure retain time expires, the output data set is removed from the JES spool.
16	Because a spool I/O error occurred, the data set is unprintable.

XTPIPORT
Port number on the target system. This number was specified in the PORTNO JCL parameter or in the printer definition.

XTP_FILTERING
This field identifies when IP PrintWay processes data two times (as two separate data sets) because the **Resubmit for filtering** option is selected in the printer definition. See “Resubmit for Filtering Considerations for IP PrintWay Exits” on page 190 for more information.

Value	Meaning
0	The **Resubmit for filtering** option is not selected in the printer definition.
1	The **Resubmit for filtering** option is selected in the printer definition and this is the first data set IP PrintWay processed. IP PrintWay transmitted this data set to Print Interface for filtering.
2	The **Resubmit for filtering** option is selected in the printer definition and either this is the second data set IP PrintWay processed or the data set was originally submitted directly to Print Interface and IP PrintWay processed only one data set. IP PrintWay transmitted this data set to the target printer.

XTP_JCL_DEST
The DEST value with which the output data set was allocated on the JES spool. If NetSpool or Print Interface allocated the data set, then this is the DEST value in the printer definition. Otherwise, it is the DEST value in the OUTPUT JCL statement unless XTP_FILTERING=2, in which case this field contains the DEST value specified in the printer definition. The value is padded to the right with blanks.

XTP_JCL_CLASS
The CLASS value with which the output data set was allocated on the JES spool. If NetSpool or Print Interface allocated the data set, then this is the CLASS value in the printer definition. Otherwise, it is the CLASS value specified in a JCL statement unless

XTP_FILTERING=2, in which case this field contains the CLASS value specified in the printer definition. The value is padded to the right with blanks.

XTP_JCL_FORMS
The FORMS value with which the output data set was allocated on the JES spool. If NetSpool or Print Interface allocated the data set, then this is the FORMS value in the printer definition. Otherwise, it is the FORMS value specified in a JCL statement. The value is padded to the right with blanks.

XTPOPTNS Transmission options. These options were specified in the printer definition or in the components named in the PRTOPTNS JCL parameter. Refer to the ANFUEXTP macro for the field names. For each transmission option, there are two fields:

- The first field contains the length of the value. A number greater than zero indicates that the second field contains a valid value.
- The second field contains the value of the option.

IP PrintWay calls the same response exit for all data set transmissions. You can install different response exits for different PrintWay FSSs. See “Installing Exits” on page 236 for information.

Output from the Response Notification Exit

At return, restore the contents of all registers. Do not set a return code in register 15.

XTPMSGFL Flags indicating the destination of a message created by this exit. IP PrintWay sets these flags to B’0’ before calling the exit. Set both flags to B’1’ to issue the message to both destinations. Place the address of the message in field XTPMSGP and the length of the message in field XTPMSGLN.

Flag	Meaning When Set to B’1’
XTPM2CON	Issue the message to the console.
XTPM2MDS	Issue the message to the IP PrintWay message-log data set.

XTPMSGLN Length of message, whose address is in XTPMSGP.

XTPMSGP Address of message created by this exit. IP PrintWay adds a message ID ANFM155I to the beginning of this message.

Examples

1. To issue a message to the IP PrintWay message log when IP PrintWay has exhausted the transmission retry limit, code the Response user exit to:
   - Check that XTPRSPCD = 12.
   - Assemble the transmission failure message.
   - Place the address of the message into XTPMSGP in the ANFUEXTP control block to point to the message.
   - Place the length of the message in field XTPMSGLN.
   - Set XTPM2MDS to B’1’ to indicate that the message should be printed to the log.
2. To issue a message to the operator’s console when IP PrintWay has found a data set that failed pre-transmission processing:
   - Check that XTPRSPCD = 08.

- Assemble the processing failure message.
- Place the address of the message into XTPMSGP in the ANFUEXTP control block to point to the message.
- Place the length of the message in field XTPMSGLN.
- Set XTPM2CON to 1 to indicate that the message should be printed to the log.

Using the Block Letter Program (ANFUBLK)

The Begin Data Set exit and the End Data Set exit can use the Block Letter program (ANFUBLK) provided with IP PrintWay to build block letters suitable for printing on separator pages. The block letters can be (1) straight or slanted, and (2) centered or left-justified. They are 12 characters high, using 12 printed lines, and can be either 8 or 12 characters wide.

ANFUBLK creates one line at a time for the exit to add to the data set. The exit must call ANFUBLK multiple times to receive all lines, until the ANFUBLK program indicates that this is the last line.

ANFUBLK produces block characters using tables in ANFUBTBL. The tables produce block characters from the following input. All other characters are translated to blanks.
- Uppercase letters: A - Z
- Symbols: $, #, and @
- Numerals: 0 - 9
- Lowercase letters, a - z, which are translated to uppercase
- Superscript EBCDIC numerals, which are translated to numerals, 0 - 9

IP PrintWay provides the assembler language source code for ANFUBTBL in SYS1.SAMPLIB(ANFUBTBS). You can modify the source to create your own block letters. The comments in ANFUBTBS describe how to change the tables. After changing the tables, assemble ANFUBTBS, rename it to ANFUBTBL, and use the sample link job in SYS1.SAMPLIB(ANFULINK) to create your exit module.

Input to ANFUBLK

When calling the Block Letter program, set the registers as follows:

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.

Also, set the following fields in ANFUEXTP. The format of the ANFUEXTP control block is shown in Figure 34 on page 194.

XTPBIFLG Flags indicating the type of block letters you want:

Flag	Meaning When Set to B'1'
XTPBSLNT	Slant letters to the right.
XTPBPRFM	Create narrow letters for faster printing. Narrow letters are 8 characters wide, centered on an 80-character line; otherwise, letters are 12 characters wide, centered on a 132-character line.

XTPBLJST Left justify the letters; otherwise, letters are centered.

Note: During centering calculations, the Block Letter program ignores blanks to the right of the last character in field `XTPBSTRG`.

XTPBFRST This is the first call to the Block Letter program to build block letters for the string whose address is in XTPBSTRG.

XTPBSTRG String to convert to block letters, from one to eight characters, padded to the right with blanks.

Output from ANFUBLK

ANFUBLK sets register 15 to 0, and also sets the following fields in ANFUEXTP:

XTPBLAST Flag set to 1 if this is the last line. If this flag is set to 0, call the Block Letter program again.

XTPBLENG Length of the line in field `XTPBOUTP`.

XTPBOUTP The output line containing part of the block letter. The first character or each line contains a carriage control character.

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.
- ANFUXRSP — takes action according to the response notification code.

You can either view the samples online or print them. You can modify a sample exit, assemble, and install it to provide your own version of the exit. Object code for the sample exits is not provided.

Installing Exits

To install an exit, compile and link-edit the 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 options 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 either:

- 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 LNKLST. If you place the exit in a LNKLST 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.

Customizing PSF for OS/390 for Sharing Network Printers

PSF for OS/390 is a separately orderable product that runs on z/OS and controls printing on AFP 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.

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 & z/OS: Customization* for information about how to configure PSF for OS/390 for printer sharing.

Chapter 9. Installing and Customizing the Infoprint Server Windows Client

This chapter describes how to customize the Infoprint Server Windows client, which consists of the Infoprint Port Monitor. The Infoprint Port Monitor lets users print from any Windows application that supports printing. You do not need to install the Infoprint Port Monitor to print from Windows systems because users can also use several other print-submission methods. See “Print Interface” on page 5 for an overview of the Windows client and how this component fits into your system.

Table 12 compares the print-submission methods you can use to print from Windows systems. In addition to using the Infoprint Port Monitor for Windows, users can use standard Windows Server Message Block (SMB) protocol, Internet Printing Protocol (IPP), or the LPR to LPD protocol.

Table 12. Comparison of Windows Printing Methods

Function	Infoprint Port Monitor	SMB	IPP	LPR/LPD
Specify Infoprint Server job attributes[1]	Yes[2]	No	Yes[3]	No
Obtain status of print job	No	Yes[4]	No	No
Windows systems supported	Windows 95/98, Me, NT (4.0 or higher), 2000, XP	Windows 95/98, NT, 2000	Windows 2000, XP	Windows NT, 2000, XP
Client installation required	Yes[5]	No	No	Yes
Infoprint Server daemon that must be started)	**aoplpd** (LPD)	None [6]	**aopippd** (IPP server)	**aoplpd** (LPD)

1. Infoprint Server job attributes are described in *z/OS Infoprint Server User’s Guide*.
2. The Infoprint Port Monitor lets users specify distribution information for the printer’s separator page (such as name and department), as well as other job attributes, on a panel that is optionally displayed before printing.
3. Users specify IPP job attributes instead of Infoprint Server job attributes. Infoprint Server converts IPP job attributes to corresponding Infoprint Server job attributes when possible. If the target printer is *not* IPP-enabled, however, Infoprint Server can support only a subset of IPP job attributes. See “Customizing the IPP Workstation Client” on page 70 for more information.
4. When the Windows job status window is open, z/OS system performance might be degraded.
5. If the z/OS printer is a Windows network printer, the Infoprint Port Monitor does *not* need to be installed on each user’s Windows system; however, users can modify Infoprint Server job attributes before printing a document only from the system on which the port monitor is installed.
6. To print using SMB protocol, you do not need to start additional Infoprint Server daemons; however, you must customize and start the z/OS SMB print server. See “Customizing Print Interface Support for the z/OS SMB Server” on page 66 for information.

To use the Infoprint Port Monitor for Windows, follow these steps:

1. Apply Infoprint Server PTFs UW86129 and UW87698.

© Copyright IBM Corp. 1995, 2002

2. Customize the Printer Inventory Manager and Print Interface, including the Print Interface LPD.
3. Download and install the Infoprint Port Monitor for Windows from the Printing Systems Division (PSD) Web site: http://www.ibm.com/printers/download.html.

   You can also download the port monitor from one of the following z/OS directories:

   - English version: **/usr/lpp/Printsrv/win/En_US**
   - Japanese version: **/usr/lpp/Printsrv/win/Ja_JP**

     The Japanese version is *not* the latest version of the Port Monitor and does not support the **Unattended mode** option.

   Optionally, download the port monitor to a shared Windows folder, so that Windows users can install the port monitor from the shared folder.

   Visit the PSD Web site or refer to *z/OS Infoprint Server User's Guide* for complete information about how to download and install the port monitor.
4. Define a z/OS printer to the Windows workstation as a local Windows printer. Use the Windows Add Printer wizard to perform this task. The Printer Inventory Manager daemon must already be started.

   To add a printer and configure the Infoprint port, the following information is required:

   - IP address or host name of the z/OS system.
   - The number of the port at which the Print Interface LPD is listening. Specify the same number as specified in the **lpd-port-number** attribute in the **aopd.conf** file.
   - The name of the printer definition for the z/OS printer in the Printer Inventory.
   - The name of the printer driver to associate with the z/OS printer on the Windows workstation.

     If the z/OS printer is an IBM AFP printer, users can associate the IBM AFP printer driver with the printer. The IBM AFP printer driver creates AFP output on the workstation. Users can download the IBM AFP printer driver from the PSD Web site.

     As an alternative to using the AFP printer driver, if your installation has installed Infoprint Server Transforms, users can associate a PCL or PostScript printer driver with the printer. The PCL or PostScript driver creates PCL or PostScript output on the workstation. The PCL to AFP transform or the PostScript to AFP transform converts the PCL or PostScript output to AFP format on the z/OS system. See Chapter 5, "Customizing Infoprint Server Transforms and the Transform Manager" on page 83 for information about customizing Infoprint Server Transforms.

   You can define the z/OS printer as a shared Windows printer in your Windows network. If you do, other Windows users can print to the printer without downloading the Infoprint Port Monitor to their Windows workstation. However, when the printer is defined as a shared Windows printer, other Windows users *cannot* modify the settings of the Infoprint port before printing a document to the shared printer. For example, a Windows user cannot specify separator page information or other job attributes before printing a document.

Tip: After the Infoprint Port Monitor is installed, use the online help provided with the port monitor to obtain step-by-step instructions to guide you through the Add Printer wizard. The online help also contains instructions for reconfiguring an

Infoprint port and adding additional Infoprint ports. After you install the port monitor, a shortcut to the online help is automatically placed on your Windows desktop.

Guideline: If you define a z/OS printer as a Windows shared printer, select the following options when you configure the Infoprint port for the printer:

- Select the **Unattended port** option in the Infoprint Port Configuration dialog box so that the Infoprint Port Monitor does not display error messages on the system where the Infoprint Port Monitor is installed.
- Do *not* select the **Prompt for attributes when printing** option in the Infoprint Port Monitor Options dialog box so that the Infoprint Port Monitor does not prompt for attributes on the system where the Infoprint Port Monitor is installed.

Appendix A. Checklist of Customization Tasks

This checklist summarizes the major tasks involved in customizing Infoprint Server and refers you to the section in this publication that describes each task. Perform steps 1 – 3 in order; perform subsequent steps in any order unless otherwise noted.

You do not need to perform all of these tasks. See the task table at the beginning of each chapter to determine whether or not the task is required.

__ 1. Install the Infoprint Server element of z/OS; refer to *z/OS Program Directory*.

__ 2. Determine which of the following optional Infoprint Server components you need to customize for your installation:

- Print Interface
- Infoprint Server Transforms and Transform Manager
- SNMP subagent
- NetSpool
- IP PrintWay
- Windows client

See Chapter 2, “Customization Roadmap” on page 23.

Rule: All installations must customize the Printer Inventory Manager component.

__ 3. **Printer Inventory Manager**: See Chapter 3, “Customizing the Printer Inventory Manager” on page 25.

__ a. “Identifying Software Requirements” on page 25

__ b. “Enabling the Infoprint Server Feature” on page 26

__ c. “Making LE Run-time Libraries Available” on page 27

__ d. “Understanding the Infoprint Server HFS Directories” on page 26

__ e. “Creating the Infoprint Server Configuration File (**aopd.conf**)” on page 28

__ f. “Setting Environment Variables for the Printer Inventory Manager” on page 31

__ g. “Establishing Security for the Printer Inventory and Operator Commands” on page 33

__ h. “Customizing Startup and Shutdown Commands and Procedures” on page 35

__ i. “Enabling Infoprint Server ISPF Panels” on page 42

__ j. “Backing Up and Restoring the Printer Inventory” on page 44

__ 4. **Print Interface**: See Chapter 4, “Customizing Print Interface” on page 49.

__ a. “Identifying Software Requirements” on page 50

__ b. “Common Customization for Print Interface” on page 51

__ 1) “Customizing z/OS Communications Server” on page 51

__ 2) “Customizing JES to Specify No Blank Truncation” on page 53

__ 3) “Editing the aopd.conf Configuration File” on page 54

__ 4) “Writing a Filter Program” on page 55

__ c. “Customizing the z/OS UNIX Printing Commands” on page 60

__ d. “Customizing the AOPPRINT JCL Procedure” on page 60

__ e. “Customizing the Print Interface LPD” on page 61

© Copyright IBM Corp. 1995, 2002

__ 1) “Customizing z/OS Communications Server (TCP/IP)” on page 62
__ 2) “Editing the aopd.conf Configuration File” on page 64
__ 3) “Customizing LPR Clients” on page 65

__ f. “Customizing Print Interface Support for the z/OS SMB Server” on page 66
__ 1) “Customizing the z/OS SMB Server” on page 67

__ g. “Customizing the Print Interface IPP Server” on page 67
__ 1) “Editing the aopd.conf Configuration File” on page 68
__ 2) “Setting Environment Variables for the IPP Server” on page 69
__ 3) “APF-Authorizing Java Run-Time Libraries” on page 69
__ 4) “Customizing the IPP Workstation Client” on page 70

__ h. “Customizing SAP R/3” on page 71

If the SAP R/3 application server runs on the z/OS system:
__ 1) “Defining an SAP User for the Callback Daemon” on page 73
__ 2) “Creating the SAP Callback Daemon Configuration File (**aopsapd.conf**)” on page 73
__ 3) “Setting Environment Variables for the SAP OMS” on page 75
__ 4) “Defining a ROMS and LOMS to the SAP R/3 Application Server” on page 75
__ 5) “Modifying the LOMS to Specify Job Attributes” on page 76
__ 6) “Customizing the SAP to AFP Transform” on page 102

If the SAP R/3 application server does *not* run on the z/OS system:
__ 1) “Customizing the Print Interface LPD” on page 61
__ 2) “Customizing the SAP to AFP Transform” on page 102

__ i. “Customizing the Print Interface Subsystem” on page 79

__ 5. **Infoprint Server Transforms** and the **Transform Manager**: See Chapter 5, “Customizing Infoprint Server Transforms and the Transform Manager” on page 83.

__ a. Install Infoprint Server Transforms; refer to the program directory for the features you ordered.

__ b. “Identifying Software Requirements” on page 85

__ c. “Installing and Enabling Infoprint Server Transforms” on page 86

__ d. “Customizing the Transform Manager” on page 86
__ 1) “Editing the Infoprint Server Configuration File (**aopd.conf**)” on page 87
__ 2) “Creating the Transform Configuration File (**aopxfd.conf**)” on page 87
__ 3) “Setting Environment Variables for the Transform Manager” on page 91

__ e. “Customizing the PCL to AFP Transform” on page 92
__ 1) “Editing the Transform Configuration File (**aopxfd.conf**)” on page 92

__ f. “Customizing the PostScript to AFP and PDF to AFP Transform” on page 95
__ 1) “Establishing Security for the PostScript and PDF to AFP Transform” on page 97

__ 2) “Editing the Transform Configuration File (**aopxfd.conf**)” on page 98

__ 3) “Customizing PostScript and PDF to AFP Transform Resource Files” on page 101

__ g. “Customizing the SAP to AFP Transform” on page 102

__ 1) “Setting Environment Variables for the SAP to AFP Transform” on page 103

__ 2) “Installing Multibyte Conversion Tables for the SAP to AFP Transform” on page 103

__ 3) “Customizing SAP to AFP Configuration Files” on page 104

__ h. “Customizing the AFP to PCL, AFP to PDF, and AFP to PostScript Transforms” on page 115

__ 1) “Editing the Transform Configuration File (**aopxfd.conf**)” on page 116

__ 2) “Customizing the Font-Mapping Table” on page 133

__ 3) “Scaling 240-Pel to 300-Pel Fonts using the AOXCF30 Program” on page 135

__ 6. **SNMP subagent**: See Chapter 6, “Customizing the SNMP Subagent” on page 139.

__ a. “Identifying Software Requirements” on page 139

__ b. “Customizing and Starting the z/OS Communications Server SNMP Agent” on page 140

__ c. “Editing the **aopd.conf** Configuration File” on page 141

__ d. “Enabling SNMP Reporting in the Printer Inventory” on page 141

__ e. Start each PSF printer for which you enabled reporting.

__ f. “Customizing the SNMP Manager” on page 142

Note: Perform steps c, d, and e, in order, after you start the SNMP subagent.

__ 7. **NetSpool**: See Chapter 7, “Customizing NetSpool” on page 143.

__ a. “Customizing JES to Specify No Blank Truncation” on page 143

__ b. “Allocating and Initializing the Message Log” on page 144

__ c. “Customizing the NetSpool Startup Procedure” on page 144

__ d. “Changing LE Run-time Options” on page 150

__ e. “Writing NetSpool Exit Programs” on page 150

__ 8. **IP PrintWay**: See Chapter 8, “Customizing IP PrintWay” on page 163.

__ a. “Identifying Software Requirements” on page 165

__ b. “Customizing z/OS Communications Server” on page 165

__ c. “Defining the IP PrintWay Functional Subsystem” on page 169

__ d. “Updating the BPXPRMxx Member of SYS1.PARMLIB” on page 178

__ e. “APF-Authorizing SYS1.IMAGELIB” on page 178

__ f. “Allocating and Initializing the Transmission-Queue Data Set” on page 178

__ g. “Restricting Access to the Transmission-Queue Data Set” on page 180

__ h. “Allocating and Initializing the IP PrintWay Message-Log Data Set” on page 180

__ i. “Customizing the IP PrintWay Startup Procedure” on page 181

__ j. “Changing LE Run-time Options” on page 185

__ k. “Creating a VTAM APPL Resource Definition” on page 185

__ l. “Creating FSS and FSA Definitions in the Printer Inventory” on page 186

__ m. “Writing IP PrintWay Exit programs” on page 189

__ n. “Customizing PSF for OS/390 for Sharing Network Printers” on page 236

__ 9. **Windows client**: See Chapter 9, “Installing and Customizing the Infoprint Server Windows Client” on page 237.

__ a. Download and install the Infoprint Port Monitor for Windows.

__ b. Customize Print Interface and the Print Interface LPD.

__ c. Add a z/OS printer to the Windows system and customize the Infoprint Port Monitor for Windows.

Note: Perform step c after you start the Printer Inventory Manager.

Appendix B. Infoprint Server Files

This appendix describes Infoprint Server files that let you customize Infoprint Server.

aopd.conf -- Infoprint Server Configuration File

Purpose

This file contains attributes that customize Infoprint Server.

Format

[**#** *comment*]
[**ascii-codepage** = *codepage*]
[**ebcdic-codepage** = *codepage*]
[**base-directory** = *path*]
[**inventory** = *inventory*]
[**ipp-port-number** = *portnumber*]
[**job-prefix** = *prefix*]
[**lpd-port-number** = *portnumber*]
[**snmp-community** = *name* | **public**]
[**start-daemons** = **{** [**ippd**] [**lpd**] [**snmpd**] [**subd**] [**xfd**] **}**]

ascii-codepage = *codepage*

The name of an ASCII code page. For code page names, refer to *z/OS C/C++ Programming Guide*. Infoprint Server uses this code page in the following ways:

- By default, this code page is displayed in the **Printer code page** field in ISPF panels for new IP PrintWay printer definitions that use the LPR, direct sockets, or IPP protocol to transmit data sets to the remote system.
- Print Interface uses this code page as the default source (document) code page when it converts data streams from ASCII to EBCDIC. A document code page specified in the printer definition or by the job submitter overrides this value.
- Print Interface uses this code page as the target (printer) code page when it converts PostScript documents created with a non-ASCII code page.
- NetSpool uses this code page as the default target (printer) code page when it converts data streams from EBCDIC to ASCII. A printer code page specified in the printer definition overrides this value.

Default: `ascii-codepage = ISO8859-1`

base-directory = *pathname*

The directory path in which the Printer Inventory Manager creates Printer Inventory files. Also, Print Interface creates files in this directory.

Note: Do *not* change this attribute while any Infoprint Server daemons are running.

Default: `base-directory = /var/Printsrv`

© Copyright IBM Corp. 1995, 2002

ebcdic-codepage = *codepage*

The name of an EBCDIC code page. For code page names, refer to *z/OS C/C++ Programming Guide*. Infoprint Server uses this code page in the following ways:

- The Printer Inventory Manager displays this code page in the **Printer code page** field in ISPF panels for new PSF for OS/390 printer definitions, General printer definitions, and IP PrintWay printer definitions that use the VTAM or e-mail protocol to transmit data sets to the remote system.
- Print Interface uses this code page as the source (document) code page when Print Interface cannot determine the user locale for the **lp** command.
- NetSpool uses this code page as the default source (document) code page when it converts data streams from EBCDIC to ASCII. A document code page in the printer definition overrides this value.

Default: `ebcdic-codepage = IBM-1047`

inventory = *inventory*

The name of the Printer Inventory and the Print Interface subsystem. Specify exactly four uppercase or lowercase letters or numbers. This name is case sensitive.

If you use the Print Interface subsystem, the first character must be an uppercase letter; it *cannot* be a lowercase letter or numeric. The job submitter specifies this name in the SUBSYS parameter on the DD JCL statement.

If you start NetSpool or IP PrintWay, specify this name on the EXEC statement of the startup procedure. Also, specify this name in the PSF for OS/390 startup procedure if you want PSF for OS/390 to use information that is specified in PSF FSS and FSA definitions in the Infoprint Server Printer Inventory.

If you change the name of the Printer Inventory and Print Interface subsystem after the Printer Inventory Manager has been started, you must stop and restart all Infoprint Server daemons. You must also stop and restart all other components and products that specify this Inventory name in their startup procedures. You must stop and restart NetSpool and IP PrintWay. If the PSF startup procedure specifies the name of the Printer Inventory, you must also stop and restart PSF for OS/390.

Default: `inventory = AOP1`

ipp-port-number = *portnumber*

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.

Default: `ipp-port-number = 631`

job-prefix = *prefix*

A prefix that Print Interface uses for all Infoprint Server job IDs. Specify two letters, numbers, or national (@ $ #) characters. Enclose a prefix that contains national characters in single or double quotation marks. The first character *cannot* be numeric.

Print Interface assigns a unique Infoprint Server job ID to each sysout data set that it creates on the JES spool. The Infoprint Server job ID begins with

the prefix you specify. The z/OS UNIX **lpstat** and **cancel** commands use the Infoprint Server job ID to display and delete data sets.

z/OS also assigns a unique job ID to each output data set. The z/OS job ID is different from the Infoprint Server job ID. JES operator commands always return the z/OS job ID.

In most cases, the job ID field of the sysout data set on the JES spool contains the Infoprint Server job ID, instead of the z/OS job ID. However, in most cases where Print Interface processes a data set that already has a z/OS job ID associated with it, Print Interface uses the existing z/OS job ID in the job ID field, instead of overriding it with the Infoprint Server job ID.

Default: `job-prefix = PS`

Example: `job-prefix = 'P#'`

lpd-port-number = *portnumber*

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 (see Note 1.) Windows users must specify this port number when they configure the Infoprint Port Monitor for Windows. This attribute is optional.

Rule: This port number must not be reserved in the *hlq*.PROFILE.TCPIP data set. See Editing the *hlq*.PROFILE.TCPIP Data Set on page 51 for more information about the PORT statement.

Recommendation: Assign port 515 to the Print Interface LPD so that you can use all of the functions that Infoprint Server provides. The following limitations apply if you assign a port other than port 515:

- Remote users cannot use commands such as LPR and **enq** to submit print requests. This is because you *cannot* configure most LPR clients to send print requests to a port other than to the well-known port 515. The Infoprint Port Monitor for Windows, however, is an exception; you can configure the Port Monitor to send print requests to any port.
- IP PrintWay *cannot* use transforms provided by Infoprint Server Transforms to transform data with the IP PrintWay resubmit for filtering option. This is because IP PrintWay always resubmits data to Print Interface at port 515.

Default: `lpd-port-number = 515`

snmp-community = *name* **|** **public**

The name assigned to the SNMP community for making SNMP requests to the z/OS system. This name must match (1) the community name provided to the z/OS SNMP agent and (2) the community name defined to the SNMP manager (Network Printer Manager). If you do not provide a community name to the z/OS SNMP agent, the z/OS 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. The community name is case sensitive. Note that the z/OS agent and the SNMP manager may have other restrictions on the length of the community name or the allowed characters.

Default: `snmp-community = public`

start-daemons = { [ippd] [lpd] [snmpd] [subd] [xfd] }

The Infoprint Server daemons that are started when you issue the **aopstart**

command. You can specify one or more of the following daemon names, enclosed in braces. The Printer Inventory Manager 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.

ippd The **aopippd** daemon. This IPP server daemon processes print jobs submitted by an Internet Printing Protocol (IPP) client.

lpd The **aoplpd** daemon. This line printer daemon processes print jobs submitted with commands, such as **lpr**, and by the Infoprint Port Monitor for Windows.

snmpd The **aopsnmpd** daemon. This SNMP subagent daemon provides status information to the z/OS SNMP agent about printers controlled by PSF for OS/390.

subd The **aopsubd** daemon. This Print Interface subsystem daemon processes output data when the subsystem is requested in the SUBSYS parameter on a DD statement.

xfd The **aopxfd** daemon. This Transform Manager daemon manages some of the data stream transforms provided by Infoprint Server Transforms.

Example: `start-daemons = {lpd subd xfd}`

Default: `start-daemons = {lpd}`

Usage Notes

1. The Infoprint Server configuration file is an optional file. If present, it resides in the directory specified in the **AOPCONF** environment variable. If **AOPCONF** is not set, the default location is **/etc/Printsrv/aopd.conf**.
2. To create the configuration file, copy sample file **/usr/lpp/Printsrv/samples/aopd.conf** to **/etc/Printsrv/aopd.conf** or to the file location specified in **AOPCONF**.
3. All attributes in this configuration file are optional. If you do not specify a attribute, the default value is used.
4. If you edit the configuration file while Infoprint Server is running, you must stop and restart the Printer Inventory Manager daemon to pick up the changes.
5. When you edit this file, follow these syntax rules:
   - Use lowercase characters for the attributes and use uppercase or lowercase characters for the values, with blank characters before or after the equal sign, if desired.
   - 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.

aopxfd.conf -- Infoprint Server Transform Configuration File

Purpose

This file is required by the Infoprint Server Transform Manager. The Transform Manager manages transforms provided by Infoprint Server Transforms, a separate licensed program product. It can also manage other transforms provided by your installation. Create an entry for each transform in this file; if you define transform classes, create a separate entry for each transform class.

Format

*comment*
transform *transformname*[*_transformclass*]
  start-command = *"daemon* [*option*]...*"*
  [**environment** = {*name* –> *value* [*name* –> *value*]...}]
  [**maximum-idle-time** = *seconds*]
  [**minimum-active** = *number*]
  [**maximum-active** = *number*]
;

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`

To specify the **letter_300** transform class on the **pcl2afp** command, type:

```
pcl2afp -c 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.

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 of environment variables with the same names that were set when the **aopstart** command was issued.

Example: `environment = {AOP_RESOURCE_PATH –> /usr/lpp/Printsrv/ps2afp}`

Note: 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. You can specify a job name of 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*.

maximum-idle-time = *seconds*
The number of seconds before the Transform Manager shuts down an idle transform daemon and system resources are freed. The number you specify must be greater than 0.

This attribute is optional; however, to avoid having many transform daemons active, consider specifying either this attribute or the **maximum-active** attribute, or both.

Default: An idle transform daemon is not shut down.

minimum-active = *number*
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 = *number*
The maximum number of transform daemons that the Transform Manager keeps active. The number must be greater than 0 and greater than or equal to the number specified in **minimum-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.

Recommendations:

- If you use the Print Interface subsystem in your installation, do not specify too low a value because jobs that use the subsystem and write multiple data sets might not be able to complete.
- To avoid having many transform daemons active, consider specifying either **maximum-active** or **maximum-idle-time**, or both.

Default: No maximum number; transform daemons are started when needed.

PCL to AFP Transform Entry

transform pcl2afp[_*transformclass*]
start-command = "pcl2afpd [-m *nnn* {**K**|**M**}]"

[**environment** = {*name* –> *value* [*name* –> *value*]... }]
[**maximum-idle-time** = *seconds*]
[**minimum-active** = *number*]
[**maximum-active** = *number*]
;

transform pcl2afp[_*transformclass*_**]**
The name of a transform and an optional transform class. Specify from 1 to 63 characters, including letters, numbers, or special characters. **transform** indicates the beginning of an entry.

pcl2afp
The name of the PCL to AFP transform provided by Infoprint Server Transforms.

transformclass
The name of an optional transform class. To use this transform class, the class must be specified when the transform is invoked. Specify a transform class if you want to transform PCL data for printers with different resolutions or paper sizes.

Default: No transform class.

Example: `transform pcl2afp_letter_300`

To specify the **letter_300** transform class on the **pcl2afp** command, type:

```
pcl2afp -c letter_300
```

start-command = "pcl2afpd [-m *nnn* **{K|M}]"**
This attribute names the transform daemon and option. Enclose the value in single or double quotation marks if you specify the option.

pcl2afpd
The name of the PCL to AFP 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.

-m *nnn***{K|M}**
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 on.

Use the following algorithm or refer to the following table to determine the amount to initially specify; 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)

The following table shows the amount of storage for different paper sizes and resolutions (240, 300, and 600 pixels per inch), using the previous algorithm:

Paper Size	240	300	600
Letter	2.0M	3.2M	12.6M
Legal	2.6M	4.0M	16.1M
A4	2.1M	3.3M	13.1M

Example: `start-command = "pcl2afpd -m 4M"`

Default: 6M

environment = {*name* **->** *value* **[** *name* **–>** *value***]... }**

Environment variables that define the environment for the PCL to AFP transform. Enclose the environment variables in braces. The values in these environment variables override values of environment variables with the same names that were set when the **aopstart** command was issued.

The following environment variables are optional:

Name	Value
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`

Example: `environment = {AOP_RESOLUTION –> 300}`

PostScript and PDF to AFP Transform Entry

transform ps2afp[_*transformclass*]
  start-command = "ps2afpd -i *initializationfile* [**-M** *nnnnn***{K|M}**]"
  [**environment = {***name* **–>** *value* [*name* **–>** *value*]... **}**]
  [**maximum-idle-time =** *seconds*]
  [**minimum-active =** *number*]
  [**maximum-active =** *number*]
;

transform ps2afp[_*transformclass***]**

This attribute names the transform and an optional transform class. It must be the first attribute in the entry.

ps2afp

The name of the PostScript and PDF to AFP transform provided by Infoprint Server Transforms.

transformclass

The name of an optional transform class. Specify from 1 to 63 characters, including letters, numbers, or special characters. To use this transform class, the class must be specified when the transform

is invoked. Specify a transform class if you want to transform PostScript or PDF data using different transform resource files.

Default: No transform class

Example: `transform ps2afp_no_font_substitution`

To use this transform class on the **ps2afp** command, type:

```
ps2afp -c no_font_substitution
```

start-command = ″**ps2afpd -i** *initializationfile* **[-M** *nnnnn***{K|M}]**″

This attribute names the transform daemon, the initialization file, and the amount of memory to use. Enclose the value in single or double quotation marks. This attribute is required.

ps2afpd

The name of the PostScript and PDF to AFP transform daemon. 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.

-i *initializationfile*

The name of the initialization file. To use the default installation file, specify **ps2afpe.ps.bin**. The **-i** option is required.

Default: None

Example: `start-command = "ps2afpd -i ps2afpe.ps.bin"`

-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.

Default: 32M

Example: `start-command = "ps2afpd -i ps2afpe.ps.bin -M 32M"`

environment = {*name* **–>** *value* **[** *name* **–>** *value***]... }**

Environment variables that define the transform environment for the PostScript and PDF to AFP transform. Enclose the environment variables in braces. The values in these environment variables override environment variables with the same name that were set when the **aopstart** command was issued.

The following environment variable is optional:

Name	Value

AOP_RESOURCE_PATH 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 resource files, **/usr/lpp/Printsrv/ps2afp**, because the default directory is always searched last.

Default: AOP_RESOURCE_PATH –> /usr/lpp/Printsrv/ps2afp

Example: `environment = {AOP_RESOURCE_PATH –> /etc/Printsrv/ps2afp}`

AFP to PCL, AFP to PostScript, or AFP to PDF Transform Entry

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

If you want to define transform classes, create a separate transform entry for each transform class.

transform afp2*ccc*[_*transformclass*]
  start-command = afp2*ccc***d**
  [**environment = {***name* **–>** *value* [*name* **–>** *value*]... **}**]
  [**maximum-idle-time =** *seconds*]
  [**minimum-active =** *number*]
  [**maximum-active =** *number*]
;

transform afp2*ccc***[**_*transformclass***]**
Name of the transform and, optionally, the name of the transform class. **transform** indicates the beginning of a transform entry.

afp2*ccc*
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

transformclass
The name of an optional transform class. Specify from 1 to 63 characters, including letters, numbers, or special characters.

Default: No transform class is defined.

Example: `transform afp2pcl_us`

To use this transform class on the **afp2pcl** command, type:

```
afp2pcl -c us
```

start-command = afp2*ccc***d**
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. This attribute is required. Valid values are:

afp2pcld The AFP to PCL transform daemon

afp2pdfd The AFP to PDF transform daemon

afp2psd The AFP to PostScript transform daemon

environment = {*name* **–>** *value* **[***name* **–>** *value***]... }**
Environment variables that define the transform environment for the transform. Enclose the environment variables in braces. The values in these environment variables override environment variables with the same name that were set when the **aopstart** command was issued. Enclose the value in single or double quotation marks if the value contains special characters or spaces.

The following environment variables are optional:

Name **Value**

AOP_ANNOTATIONS
Indicates whether the AFP to PDF transform is to produce annotations in the PDF file. The AFP to PDF transform can create an annotation on each page. The annotation contains printing instructions such as ″Normal duplex Front side″. For more information about annotations, see the online help provided with Adobe Acrobat reader. This variable applies only to the AFP to PDF transform; other transforms ignore this variable. Valid values are:

yes The AFP to PDF transform produces annotations.

no The AFP to PDF transform does not produce annotations.

Default: AOP_ANNOTATIONS –> yes

Example: `environment = {AOP_ANNOTATIONS –> no}`

AOP_CHARS The default coded font. 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 one to four 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 six-character names, not counting the X0 or XZ prefix. For these fonts, use the four-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.

Recommendations:

- You might want to specify the same font as the resident font in the AFP printer to which output is usually printed.
- For the AFP to PDF and the AFP to PostScript transforms, specify an outline font because outline fonts provide higher quality output for printing and viewing.
- For the AFP to PCL transform, specify a raster font because this transform requires raster fonts.

Default: AOP_CHARS –> 60d9

Note: The transforms prefix X0 to this font.

Example: `environment = {AOP_CHARS –> 60d8}`

AOP_COLOR Indicates whether the transform is to produce color output. Specify **yes** if the printer supports color. This variable applies only to the AFP to PCL and AFP to PostScript transforms. The AFP to PDF transform ignores this variable and always produces output in color. Valid values are:

yes The transform produces color output.

no The transform performs color simulation.

Default: AOP_COLOR –> no

Example: `environment = {AOP_COLOR –> yes}`

AOP_CUTSHEET

Indicates whether the transform is to prepare the output for printing on a cutsheet printer. Valid values are:

yes The output is to be printed on a cutsheet printer. Therefore, the transform uses the cutsheet specification in the form definition to determine whether or not to send medium orientation information to the printer. Refer to the description of the CUTSHEET command in *IBM Page Printer Formatting Aid: User's Guide* for more information about the cutsheet specification in the form definition.

Tip: Select this option if your output is incorrectly printing in the down direction on a cutsheet printer.

no The output is not to be printed on a cutsheet printer. Therefore, the transform always sends medium orientation information to the printer.

Default: AOP_CUTSHEET –> no

Example: `environment = {AOP_CUTSHEET –> no}`

AOP_FLATE Indicates whether the AFP to PDF transform is to use the Adobe Flate compression algorithm to compress the output. 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:

yes The transform produces compressed output.

no The transform does not produce compressed output; however throughput is higher.

Default: AOP_FLATE –> yes

Example: `environment = {AOP_FLATE –> 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.
  - Specify AFP outline font libraries if data to be transformed refers to outline fonts.
- **AFP to PDF transform:**
  - Specify AFP 300-pel raster font libraries if *either* (1) you specify a raster font in the AOP_CHARS environment variable *or* (2) 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* (1) data to be transformed references outline fonts *or* (2) you specify AOP_FONTMAP –> yes to map raster to outline fonts.
- **AFP to PostScript transform:**
  - Specify AFP 300-pel raster font libraries if *either* (1) you specify a raster font in the AOP_CHARS environment variable *or* (2) 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* (1) data to be transformed references outline fonts *or* (2) you specify AOP_FONTMAP –> yes to map raster to outline fonts.

Default: AOP_FONTLIB –> ″sys1.font300 sys1.fontoln″

Examples:

```
environment={AOP_FONTLIB –>
    "sys1.font300 sys1.fontoln inst.font300"}
```

```
environment={AOP_FONTLIB –> "sys1.font300"}
```

AOP_FONTMAP

Indicates 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.

Recommendation: Enable font mapping for all transforms. See “Mapping AFP Raster and AFP Outline Fonts” on page 122 for more information.

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_PAGEDEFLIB 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_PAGEDEF

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_PAGEDEF –> p1p08682

Example: `environment = {AOP_PAGEDEF –> p1q09182}`

AOP_PAGEDEFLIB

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_PAGEDEFLIB –> ″sys1.pdeflib″

Example: `environment = {AOP_PAGEDEFLIB –> "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. Valid paper names and their sizes are:

Paper Name	Size
a3	292.25 x 413.25 mm
a4	210 x 297 mm
a4ee	210 x 297 mm
a5	148.3 x 210 mm
c5	162 x 229 mm (6.48 x 9.16 in.)
com10	4.125 x 9.5 in. (104.8 x 241.3 mm)
dl	110 x 220 mm (4.4 x 8.8 in)

executive	7.25 x 10.5 in (185 x 267 mm)
legal	8.5 x 14.0 in (216 x 356 mm)
letter	8.5 x 11.0 in (216 x 279 mm)
letteree	8.5 x 11.0 in (216 x 279 mm)
monarch	3.875 x 7.5 in. (98.4 x 190.5 mm)

Note: Use the **a4ee** and **letteree** paper names when the printer is configured to print edge-to-edge.

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 letter"}`

AOP_PJL Indicates whether the printer accepts all PCL 5 commands, including PJL commands. All printers that support PCL 5 support PJL commands. Some printers that support only PCL 4 do not support PJL commands. Sometimes, a printer that does not support PJL commands prints a smiley face where a PJL command occurs in the data stream or prints the PJL command on the first page.

This environment variable applies only to the AFP to PCL transform; the AFP to PostScript and AFP to PDF transforms ignore this variable. Valid values are:

yes	The printer accepts PJL commands.
no	The printer does not accept PJL commands; therefore, the AFP to PCL transform does not generate PJL commands.

Default: AOP_PJL –> yes

Example: `environment = {AOP_PJL –> no}`

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 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 0 2″

Example: `environment = {AOP_TRAYID –> "1 4 1 1 1 1 1 1 1 2"}`

Usage Notes

1. The transform configuration file is required when you start the Transform Manager daemon. Some attributes in the file are optional; if you do not specify an optional attribute, the default value is used.
2. The transform configuration file resides in the directory specified in the AOPXFD_CONF environment variable. If this environment variable is not set, the default location is **/etc/Printsrv/aopxfd.conf**.
3. To create the Transform configuration file, copy sample file **/usr/lpp/Printsrv/samples/aopxfd.conf** to **/etc/Printsrv/aopxfd.conf** or to the directory specified in the AOPXFD_CONF environment variable.
4. If you edit the configuration file after starting the transform daemon, use the **aopstop -d xfd** command to stop the transform daemon, and use the **aopstart** command to restart it.
5. When you edit this 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 and 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.
   - If you enclose a value in single or double quotation marks, type the entire value on the same line.
   - Start comments with: #
   - Include blank lines if desired.

Examples

```
transform pcl2afp
      start-command = "pcl2afpd -m 4M"
      minimum-active = 1
      maximum-active = 2
      maximum-idle-time = 300   # 5 minutes
      environment = {
          AOP_PAGE_HEIGHT -> 11i
          AOP_PAGE_WIDTH -> 8.5i
          AOP_RESOLUTION -> 240
```

```
          AOP_HORIZONTAL_MARGINS -> 0.167i
          AOP_VERTICAL_MARGINS -> 0.167i
       }
;
transform pcl2afp_letter_300
      start-command = pcl2afpd
      minimum-active = 0
      maximum-active = 2
      maximum-idle-time = 300  # 5 minutes
      environment = {
          AOP_PAGE_HEIGHT -> 11i
          AOP_PAGE_WIDTH -> 8.5i
          AOP_RESOLUTION -> 300
          AOP_HORIZONTAL_MARGINS -> 0.167i
          AOP_VERTICAL_MARGINS -> 0.167i
       }
;
transform ps2afp
      start-command = "ps2afpd -i ps2afpe.ps.bin"
      minimum-active = 1
      maximum-active = 2
      maximum-idle-time = 300   # 5 minutes
;
transform afp2pcl_us
   start-command = afp2pcld
   min-active = 1
   max-active = 2
   maximum-idle-time = 300 # 5 minutes
   environment = {
      AOP_FONTLIB -> "sys1.font300"
      AOP_PAGEDEFLIB -> "sys1.pdeflib"
      AOP_FORMDEFLIB -> "sys1.fdeflib"
      AOP_PAGESEGLIB -> "sys1.pseglib"
      AOP_OVERLAYLIB -> "sys1.overlib"
      AOP_TRAYID -> "1 4 0 0 0 0 0 0 0 2"
      AOP_PAPER ->
        "letter legal letter 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
   }
;
```

aopsapd.conf -- Infoprint Server SAP Callback Daemon Configuration File

Purpose

This file is required when you use the Infoprint Server SAP Output Management System (OMS).

Format

```
# comment
system system-name
     initial-target = instance-name
     client = clientname
     userid = userid
     password = password
     language = language
;
```

system *system-name*
: The name of the SAP R/3 system that contains an LOMS for which you want callback notification. The SAP R/3 system can run on any z/OS or non-z/OS system. This attribute must be the first attribute in the entry.

initial-target = *instance-name*
: The name of an SAP R/3 server instance on the SAP R/3 system defined in the **system** attribute. The Print Interface Callback daemon initially contacts this server instance to obtain configuration information about the LOMSs defined on that SAP R/3 system.

client = *clientname*
: The name of the client defined to SAP for use by the Callback daemon.

userid = *userid*
: The user ID defined to SAP for use by the Callback daemon.

password = *password*
: The password defined to SAP for use by the user ID.

language = *language*
: The language used for messages from the SAP R/3 application server.

Usage Notes

1. All attributes are required.
2. The configuration file resides in the directory specified in the AOPSAPD_CONF environment variable. If this environment variable is not set, the default location is **/etc/Printsrv/aopsapd.conf**.
3. To create the configuration file, copy sample file **/usr/lpp/Printsrv/samples/aopsapd.conf** to **/etc/Printsrv/aopsapd.conf** or to the directory specified in the AOPSAPD_CONF environment variable.
4. If you edit the configuration file after starting Infoprint Server, use the **aopstop** command or AOPSTOP procedure to stop all Infoprint Server daemons, and then use the **aopstart** command or AOPSTART procedure to restart them.
5. When you edit this file, follow these syntax rules:
   - End the last attribute in each 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.
- 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.

Examples

This example shows two sample entries in the **aopsapd.conf** configuration file.

```
system EZ0
   initial-target = boefcs1_EZ0_00
   client = 000
   userid = mike
   password = xxxxxxx
   language = EN
;

system EX0
   initial-target = boefcs5_EX0_00
   client = 000
   userid = mike
   password = xxxxxxx
   language = EN
;
```

Appendix C. Environment Variables

This appendix summarizes the environment variables that affect processing of Infoprint Server. Table 13 summarizes the variables and indicates where you need to set the variable:

- In the **/etc/profile** file. See "Setting Environment Variables for the Printer Inventory Manager" on page 31 for information about how to specify variables in this file.
- In the **aopstart** EXEC. See "Editing the **aopstart** EXEC" on page 36 for information about how to specify variables in this EXEC.
- In the STDENV data set of the IP PrintWay startup procedure. See "Editing the ANFWPROC Startup Procedure" on page 182 for information about how to specify variables in the startup procedure.

Some environment variables should be set in *multiple* locations. **Required** variables must be set; **optional** variables must be set only if the default value is not suitable for your installation.

Table 13. Where to Define Environment Variables

Set this environment variable	In /etc/profile file	In aopstart EXEC	In IP PrintWay startup procedure
_BPXK_SETIBMOPT_TRANSPORT	No	Optional	No
AOP_SAP2AFP_RESOURCES	Optional	Optional	No
AOPCONF	Optional	Optional	No
AOPMAILER	No	No	Optional
AOPOPTIONS	Optional	No	No
AOPPATH	Optional	No	No
AOPSAPD_CONF	Optional	Optional	No
AOPTRACEON[1]	Optional	Optional[2]	No
AOPTRACEDIR[1]	Optional	Optional	No
AOPXFD_CONF	No	Optional	No
CLASSPATH	No	Optional	No
JAVA_HOME	No	Optional	No
LIBPATH	Required	Optional	Optional
MANPATH	Required	No	No
NLSPATH	Required	Optional	Optional
PATH	Required	Optional	Optional
PRINTER or LPDEST	Optional	No	No
STEPLIB	Optional	Optional	No

1. The AOPTRACEON and AOPTRACEDIR variables control tracing. Refer to *z/OS Infoprint Server Messages and Diagnosis* for more information about these variables.
2. You can also specify the AOPTRACEON variable in the AOPSTART startup procedure. See "Editing the AOPSTART Startup Procedure" on page 39 for information.

© Copyright IBM Corp. 1995, 2002

Variable	Meaning
_BPXK_SETIBMOPT_TRANSPORT	The TCP/IP job name associated with the TCP/IP stack you want to use. This variable is optional; however, you must set it if your installation has defined multiple TCP/IP stacks on the z/OS system and you start the Print Interface LPD. (The Print Interface LPD starts by default.) Refer to *z/OS UNIX System Services Planning* for more information about this environment variable.
AOP_64K_JOBID	If this environment variable is defined, Print Interface limits the job IDs assigned to print jobs on the JES spool to the range of 1 to 65535. If this environment variable is not defined, the job ID can be in the range of 1 to 999999. **Note:** This variable is defined for use by the z/OS SMB server. IBM recommends that other applications do *not* define this variable because system performance might be degraded.
AOP_SAP2AFP_RESOURCES	Path where resources used by the SAP to AFP transform are located. This environment variable is optional; however, you must set it if you did *not* install Infoprint Server Transforms resources in the default directory, **/usr/lpp/Printsrv/sap2afp** and you use the SAP to AFP transform. **Default:** `/usr/lpp/Printsrv/sap2afp`
AOPCONF	The full path name of the Infoprint Server configuration file. This environment variable is optional; however, you must set it if you did *not* create the configuration file in **/etc/Printsrv/aopd.conf**. **Default:** `/etc/Printsrv/aopd.conf`
AOPMAILER	Specifies the full path name of the z/OS UNIX sendmail command. This environment variable is optional; however, you must set it if you did *not* install the sendmail executable file in the default directory, **/usr/sbin**. **Default:** AOPMAILER=/usr/sbin/sendmail
AOPOPTIONS	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 AOPOPTIONS environment variable before the options you specify on the command line, a user can override the values of this variable. This variable is optional; however, you might want to set it to specify job attributes that are constant for all print jobs. Refer to *z/OS Infoprint Server User's Guide* for the format of job attributes and values.
AOPPATH	Path where job attribute files, used by the **lp** command and AOPPRINT procedure, are located. This environment variable is optional; however, you must set it if you create attribute files for the **lp** command. Refer to *z/OS Infoprint Server User's Guide* for information about attribute files.

AOPSAPD_CONF

Full path name of the SAP Callback daemon configuration file. This environment variable is optional; however, you must set it if you print from SAP R/3 and the configuration file is *not* in **/etc/Printsrv/aopsapd.conf**.

Default: `/etc/Printsrv/aopsapd.conf`

AOPXFD_CONF

The full path name of the transform configuration file. This environment variable is optional; however, you must set if if you use data transforms and did *not* create the transform configuration file in **/etc/Printsrv/aopxfd.conf**.

Default: `/etc/Printsrv/aopxfd.conf`

CLASSPATH

The full path names of Infoprint Server IPP files. This environment variable is optional; however, you must set it if you did *not* install Infoprint Server IPP files in the default directory, **/usr/lpp/Printsrv** and you start the Print Interface IPP server. If you installed Infoprint Server IPP files in a different directory, add the following values to any existing values in this variable:

- ***directory*/classes/ipp.jar**
- ***directory*/classes/ippserver.jar**
- ***directory*/classes/ippreal.jar**

directory is the directory that contains the files.

Default: `/usr/lpp/Printsrv/classes/ipp.jar:`
`/usr/lpp/Printsrv/classes/ippserver.jar:`
`/usr/lpp/Printsrv/classes/ippreal.jar`

JAVA_HOME

The path used to locate Java 1.3 files. This environment variable is optional; however, you must set this environment variable if you did *not* install Java 1.3 files in the default Java directories and you start the Print Interface IPP server.

Default: `/usr/lpp/java/IBM/J1.3`

LIBPATH

The path used to locate dynamic link library (DLL) files. This environment variable is *required*. 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 that directory to any existing values.

Also, if you run the SAP Output Management System (OMS), add the directory that contains the SAP remote function calls DLL (**librfc.dll**).

Default: `/usr/lpp/Printsrv/lib`
(This default applies only to the **aopstart** EXEC.)

MANPATH

The path of directories that contain the man pages. Infoprint Server man pages are available only in English. Add **/usr/lpp/Printsrv/man/En_US** to the values in this variable. This variable is *required*.

Note: Add the new directory path *before* **/usr/man/%L** in the MANPATH environment variable, so that the Infoprint Server versions of the **lp**, **lpstat**, and **cancel** man pages are displayed.

NLSPATH The path of directories that contain message catalogs. This environment variable is *required*. 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:

English	**/usr/lpp/Printsrv/En_US/%N**
Japanese	**/usr/lpp/Printsrv/Ja_JP/%N**

%L represents the value of the LANG environment variable. **%N** is the catalog filename.

Default: `/usr/lpp/Printsrv/%L/%N:/usr/lpp/Printsrv/En_US/%N:/usr/lib/nls/msg/%L/%N`
(This default applies only to the **aopstart** EXEC.)

PATH The path used to locate executables. This environment variable is *required*. If you installed Infoprint Server executables in the default directory, add **/usr/lpp/Printsrv/bin** to the values in this variable. Add the directory *before* **/bin** in the PATH environment variable to ensure that the Infoprint Server versions of the **lp**, **lpstat**, and **cancel** commands are invoked.

Default: `/usr/lpp/Printsrv/bin:/bin`
(This default applies only to the **aopstart** EXEC.)

PRINTER or LPDEST
The name of the default printer for the **lp** command. The value in LPDEST overrides the value in PRINTER.

This variable is optional. You can, instead, use Infoprint Server ISPF panels to define the name of the default printer; see *z/OS Infoprint Server Operation and Administration* for information. The printer named in either LPDEST or PRINTER overrides the printer named on the ISPF configuration panel.

STEPLIB The LE and C++ run-time libraries. This environment variable is optional; however, you must specify it if you did not add the *hlq*.SCEERUN and *hlq*.SCLBDLL data sets to system LNKLST.

Appendix D. AOPBATCH Program

This appendix describes the AOPBATCH utility program provided by Infoprint Server. AOPBATCH lets you use MVS job control language (JCL) to run a program that resides in a hierarchical file system (HFS).

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.

Format

JCL:

```
EXEC PGM=AOPBATCH,PARM='/program_name options'
```

Description

AOPBATCH lets you submit an MVS batch job to run an executable program that resides in an HFS. AOPBATCH uses the **spawn**() function to create a process that runs in the same address space as AOPBATCH under these conditions:

- The set-user-ID and 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 program object. This attribute is set by default.

If these conditions are not met, AOPBATCH spawns the process in a separate address space.

With AOPBATCH you can specify the following optional DD statements to set up the UNIX environment and to allocate **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 STDIN or STDOUT when you run a transform command because transform input and transformed output is not text data.

STDENV

Specifies environment variables for use by program *program_name*. You can specify the environment variables in-stream in the JCL, in a UNIX file, or in an MVS data set. Specify the environment variables in the format *variable*=*value*, with one environment variable per 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

© Copyright IBM Corp. 1995, 2002

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

STDERR
Specifies a SYSOUT data set, UNIX file, or MVS data set.

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

STDOUT
Specifies a SYSOUT data set, UNIX file, or MVS data set.

Parameters

AOPBATCH accepts the name of the executable program file you want to run followed by options for that program.

/ The optional slash indicates that the PARM data that follows is input to AOPBATCH. If you omit the initial slash, your PARM data might be interpreted as C^{++} run-time options. You *must* include the initial slash if any of the PARM data itself includes a slash. For example, if the transform name is `/mylib/ps2afp`, specify: `PARM='//mylib/ps2afp ...'`.

program_name
The name of an 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 path. 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 or Infoprint Server Transforms, which you can run using AOPBATCH are:

- **pidu**, the Printer Inventory Definition Utility program
- **aopmig**, the Printer Inventory migration program
- Transforms provided by Infoprint Server Transforms: **afp2pcl, afp2pdf, afp2ps, pcl2afp, pdf2afp, ps2afp**, and **sap2afp**.

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:

- For the **pidu** program, refer to *z/OS Infoprint Server Operation and Administration*.
- For the **aopmig** program, refer to *z/OS Infoprint Server Migration*.
- For the transforms, refer to *z/OS Infoprint Server User's Guide*.

Usage Notes

- Sequence numbers in columns 73 - 80 in data specified with the STDIN DD statement are treated as part of the data. Therefore, if you use the ISPF editor to edit the data, consider turning sequence numbers off by typing NUMBER OFF on the command line before you type any data; if sequence numbers already exist, type UNNUM to remove them and set number mode off.
- Sequence numbers in columns 73 - 80 in data specified with the STDENV DD statement are ignored and *not* treated as part of the data.

- AOPBATCH is installed in SYS1.LINKLIB; therefore, you do not need to include a STEPLIB DD statement in your JCL.
- If you have not added the Language Environment run-time library (CEE.SCEERUN) or the C++ run-time library (CBC.SCLBDLL) to the system LNKLST, concatenate these data sets to a STEPLIB DD statement.
- 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`.
- When you run a transform, you must specify the transform input data set or file as a transform argument, and you must specify the **-o** transform option to identify where you want the transform to write its output. This is because when you use AOPBATCH, the transform cannot read input from standard input and cannot write its output to standard output. To identify the transform input and transform output data set or file, you can specify either a DD statement name or a data set name. You must specify a DD statement name if you want the transform to write output to an MVS data set that does not already exist, and you must include the DD statement to allocate the output data set.
- When you specify the name of a DD statement in the PARM parameter, use the following format:

  `//DD:`*`DDname`*

  where *DDname* is the name of the DD statement.
- When you specify an MVS data set name in the PARM parameter, code two slashes before the data set name and enclose the data set name in two sets of single quotes if you specify a fully qualified data set name. If you do *not* enclose the data set name in quotes, then C/C++ assumes that the data set name is not fully qualified and adds a high-level qualifier to the name you specify, as follows:
  - If you are running under TSO (batch or interactive), C/C++ appends the TSO user prefix.
  - If you are running under MVS batch or IMS (batch or online), C/C++ appends the RACF user ID.
- Do not specify the _BPX_SHAREAS environment variable with the STDENV DD statement. AOPBATCH will set it appropriately.

Return Codes

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

Examples

The following examples show how to use the AOPBATCH program. The environment variables shown in the first example are set to default values. If you installed and created Infoprint Server files in default directories, you do not need to specify these environment variables, and you can omit the STDENV DD statement.

For more examples that show how to use AOPBATCH to run the transforms, refer to *z/OS Infoprint Server User's Guide*.

1. This example shows how to use the AOPBATCH program to run 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/lib/nls/msg/%L/%N
/*
```

2. This example shows how to use the AOPBATCH program to run the Infoprint Server **ps2afp** command. In this example, input to the command is in file **/tmp/input.ps**, and output is written to MVS data set HLQ.OUTPUT.AFP.

```
//AOPBATCH JOB  ...
//PRINT    EXEC PGM=AOPBATCH,PARM='/ps2afp -o //DD:OUTPUT /tmp/input.ps'
//OUTPUT   DD   DSN=HLQ.OUTPUT.AFP,DISP=SHR
//STDERR   DD   SYSOUT=*
```

Sample Jobs

IBM provides the following sample jobs in SYS1.SAMPLIB, which use AOPBATCH:

- AOPPIDU, which runs the Printer Inventory Definition Utility, **pidu**
- AOPMIGRT, which runs the migration program, **aopmig**
- AOPJAUTH, which APF-authorizes Java run-time libraries

Appendix E. Application Programming Interface (API)

Infoprint Server provides an Application Programming Interface (API) that lets you code C or C++ applications to perform the following functions:

- Create an Infoprint Server print job on the z/OS Job Entry System (JES) spool
- Cancel an Infoprint Server print job
- Query the status of an Infoprint Server print job
- Query the status of all Infoprint Server print jobs for a printer
- Query information about all printers defined in the Infoprint Server Printer Inventory
- Connect to a transform daemon

This appendix describes the following topics for application programmers who want to use the Infoprint Server API:

- "API Files"
- "API Data Structures"
- "API Functions" on page 278
- "Programming Considerations" on page 291
- "Customization Requirements" on page 293

API Files

The following table summarizes the files provided with the API. For each file, the table lists the default directory into which the file is installed.

File	Directory	Description
`aopapi.dll`	`/usr/lpp/Printsrv/lib`	Executable code
`aopapi.h`	`/usr/lpp/Printsrv/samples`	Include file containing data structures and function declarations
`libaopapi.a`	`/usr/lpp/Printsrv/lib`	Static library containing function stubs

API Data Structures

The following table summarizes the data structures provided with the Infoprint Server API. After the table, each data structure is described more fully in alphabetical order.

Data Structure	Include File	Description
`DocumentInfo`	`aopapi.h`	Information about a document in an Infoprint Server print job
`JobInfo`	`aopapi.h`	Information about an Infoprint Server print job and each document in the job
`PrinterInfo`	`aopapi.h`	Information about an Infoprint Server printer
`JobToken`	`aopapi.h`	A token that uniquely identifies an Infoprint Server print job
`QueryToken`	`aopapi.h`	A token that uniquely identifies an Infoprint Server query request

© Copyright IBM Corp. 1995, 2002

Data Structure	Include File	Description
Transform	aopapi.h	Information about a connection between your transform program and a transform daemon.

DocumentInfo - Information about a document in a job

The DocumentInfo data structure contains information about a document. It is included in the JobInfo data structure. The status and format fields are described by the DocumentStatus and DocumentFormat enumerations.

```
typedef struct
{
     char           status;          /* Current document status           */
     char           format;          /* Format of the document            */
     char           reserved[2];
     unsigned int   size;            /* Size in bytes of document on      */
                                     /*   spool                           */
     char*          name;            /* Filename of document              */
} DocumentInfo;
```

The DocumentStatus enumeration identifies the status of the print job on the JES spool. For an explanation of the job states in this data structure, refer to the description of the lpstat command in the *z/OS Infoprint Server User's Guide*. The job states in the DocumentStatus structure are the same as those returned by the lpstat command. (The selected state in DocumentStatus is the same as the processing state of the lpstat command.)

```
typedef enum
{
   status_completed    = 0,
   status_pending      = 1,
   status_selected     = 2,
   status_held         = 3,
   status_failed       = 4,
   status_purged       = 5
} DocumentStatus;
```

The DocumentFormat enumeration identifies the format of the input data stream. It is included in the DocumentInfo data structure. Infoprint Server determines the format automatically unless the **document-format** job attribute was specified as an attribute to the CreatePrintFile function. For an explanation of the data formats in this data structure, refer to the description of the **document-format** job attribute in the *z/OS Infoprint Server User's Guide*.

```
typedef enum
{
   format_line         = 0,
   format_modca        = 1,
   format_other        = 2,
   format_pcl          = 3,
   format_postscript   = 4,
   format_text         = 5,
   format_pdf          = 6,
   format_sap          = 7
} DocumentFormat;
```

JobInfo - Information about an Infoprint Server print job and each document in the job

This data structure contains information about an Infoprint Server print job and each document in the print job. It is returned by the GetJobInfo and EnumJobs functions. The documents field is described by the DocumentInfo data structure.

```
typedef struct
{
     unsigned int   id;              /* Infoprint Server job number          */
     char*          printer_name;    /* The printer on which the job was     */
                                     /*   submitted.                         */
     char*          user_name;       /* The owner of the job                 */
     char*          name_text;       /* The value of the name-text attribute */
     int            priority;        /* Priority at which job was submitted  */
     time_t         submitted;       /* The time the job was submitted       */
                                     /*   in seconds since epoch.            */
     unsigned int   ndocuments;      /* The number of documents contained    */
                                     /*   in the job.                        */
     DocumentInfo** documents;       /* One DocumentInfo structure for each  */
                                     /*    document in the job.              */
} JobInfo;
```

JobToken - Identifies an Infoprint Server print job

This opaque data structure uniquely identifies a request to create an output data set on the JES spool. The `CreatePrintFile` function returns the `JobToken` data structure when it receives a request to create an output data set. It is then input to the `WritePrintFile`, `ClosePrintFile`, and `AbortPrintFile` functions. `JobToken` is no longer valid after the `ClosePrintFile` function is called to close the output data set or the `AbortPrintFile` function is called to terminate the request to create an output data set.

```
typedef struct { void* p[2]; } JobToken;
```

PrinterInfo - Information about an Infoprint Server printer

This data structure contains information about an Infoprint Server printer that is described by a printer definition in the Infoprint Server Printer Inventory. It is returned by the `GetPrinterInfo` and `EnumPrinters` functions. The fields in this structure are:

printer_name The name of the printer definition

description The description specified in the printer definition or a NULL value if no description is specified

location The location specified in the printer definition or a NULL value if no location is specified

njobs Number of Infoprint Server jobs for this printer definition that are in the pending, selected, and held states

status Described by the `PrinterStatus` enumeration

```
typedef struct
{
     char*        printer_name;    /* Printer definition name                    */
     char*        description;     /* Printer description or NULL if not supplied */
     char*        location;        /* Printer location or NULL if not supplied   */
     unsigned in  njobs;           /* Number of jobs queued on this printer      */
     char         status;          /* Current printer status                     */
 } PrinterInfo;
```

The `PrinterStatus` enumeration identifies the status of the printer. It is included in the `PrinterInfo` data structure. Currently, the status of all printers is enabled.

```
typedef enum
{
   status_enabled    = 0,            /* The printer is accepting new jobs         */
   status_disabled   = 1             /* The printer is not accepting new jobs     */
} PrinterStatus;
```

QueryToken - Identifies an Infoprint Server query request

This opaque data structure uniquely identifies a query request for an enumeration of print jobs for a printer. It is returned by the `BeginEnumJobs` function; it is input to the `EnumJobs` and `EndEnumJobs` functions. It is no longer valid after your application calls the `EndEnumJobs` function to end the enumeration.

```
typedef struct { void* p[2]; } QueryToken;
```

Transform - Information About a Transform Connection

This data structure describes a connection between your transform program and a transform daemon. It is returned by the `RequestTransform` function, and is input to the `ReleaseTransform` function.

```
typedef struct {
   const char*    class_name; /* name of transformer class (63 bytes maximum)     */
   const int      fd;         /* connection to transform daemon                   */
} Transform;
```

The fields in this structure are:

`class_name` The name of the transform and class in the format: *transformname*[_*transformclass*]. This value must match the name of a transform entry in the transform configuration file (**aopd.conf**). The format of *class_name* is: *transformname*[_*transformclass*].

`fd` A UNIX file descriptor used to send input and receive output from the transform daemon.

API Functions

The following tables summarize the C library functions provided with the API. Each function is listed according to the type of function it performs. After the tables, each function is described more fully in alphabetical order.

Initializing and Terminating the API

Function	Include File	Description
`InitAPI`	`aopapi.h`	Initializes the API
`TermAPI`	`aopapi.h`	Terminates the API

Creating a Print Job

Function	Include File	Description
`CreatePrintFile`	`aopapi.h`	Creates an Infoprint Server print job
`WritePrintFile`	`aopapi.h`	Writes data to an Infoprint Server print job
`ClosePrintFile`	`aopapi.h`	Closes an Infoprint Server print job
`AbortPrintFile`	`aopapi.h`	Terminates an Infoprint Server print job

Canceling a Print Job

Function	Include File	Description
`CancelJob`	`aopapi.h`	Cancels an Infoprint Server print job

Querying Print Jobs

Function	Include File	Description
GetJobInfo	aopapi.h	Returns status of an Infoprint Server print job
BeginEnumJobs	aopapi.h	Begins an enumeration of Infoprint Server print jobs for a printer
EnumJobs	aopapi.h	Returns an enumeration of Infoprint Server print jobs for a printer
EndEnumJobs	aopapi.h	Ends an enumeration of Infoprint Server print jobs for a printer

Querying Printers

Function	Include File	Description
GetPrinterInfo	aopapi.h	Returns information about an Infoprint Server printer
BeginEnumPrinters	aopapi.h	Begins an enumeration of Infoprint Server printers
EnumPrinters	aopapi.h	Returns an enumeration of Infoprint Server printers
EndEnumPrinters	aopapi.h	Ends an enumeration of Infoprint Server printers

Connecting to the Transform Manager

Function	Include File	Description
RequestTransform	aopapi.h	Establishes a connection with a transform daemon
ReleaseTransform	aopapi.h	Releases a connection with a transform daemon

Handling Errors

Function	Include File	Description
SetTerminationHandler	aopapi.h	Registers a function to be called at program termination
ErrorString	aopapi.h	Returns a pointer to an API error message
ErrorNumber	aopapi.h	Returns the API error code
BufferSizeRequired	aopapi.h	Returns the required buffer size after a buffer overflow error

AbortPrintFile -- Terminate an Infoprint Server print job

Format

```
#include <aopapi.h>
int AbortPrintFile( JobToken  token );
```

Description

The `AbortPrintFile` function removes an output data set from the JES spool that has not yet been closed by the `ClosePrintFile` function. The *token* argument must be a valid job token returned by the `CreatePrintFile` function. After you call the `AbortPrintFile` function, this job token is no longer valid and cannot be used by any other functions.

Returned Value

`AbortPrintFile` returns:

- Zero if the function was successful.
- A non-zero value if the job token is not valid or if the function was not successful for another reason. You can use the `ErrorNumber` function to obtain an error code, and you can use the `ErrorString` function to obtain a description of the error.

BeginEnumJobs -- Begin an enumeration of Infoprint Server print jobs for a printer

Format

```
#include <aopapi.h>
int BeginEnumJobs( QueryToken*  out_token,
                   const char*  printer_name );
```

Description

The `BeginEnumJobs` function begins an enumeration of all Infoprint Server print jobs for a printer. After calling `BeginEnumJobs`, call these functions:

1. `EnumJobs` to obtain the enumeration of print jobs.
2. `EndEnumJobs` to end the enumeration.

Input parameters are:

printer_name The name of a printer definition in the Infoprint Server Printer Inventory whose jobs you want to enumerate; printer definition names are case sensitive.

out_token A query token that is returned by this function. You must pass this token as an argument to the `EnumJobs` and `EndEnumJobs` functions.

Returned Value

`BeginEnumJobs` returns:

- Zero and sets *out_token* if the function was successful.
- A non-zero value if the function was not successful. You can use the `ErrorNumber` function to obtain an error code, and you can use the `ErrorString` function to obtain a description of the error.

BeginEnumPrinters -- Begin an enumeration of Infoprint Server printers

Format

```
#include <aopapi.h>
int BeginEnumPrinters( QueryToken*  out_token );
```

Description

The `BeginEnumPrinters` function begins an enumeration of all printers defined by printer definitions in the Infoprint Server Printer Inventory. After calling `BeginEnumPrinters`, call these functions:

1. `EnumPrinters` to obtain the enumeration of printers.
2. `EndEnumPrinters` to end the enumeration.

Argument *out_token* is a query token returned by this function. You must pass this token as an argument to the `EnumPrinters` and `EndEnumPrinters` functions.

Returned Value

`BeginEnumPrinters` returns:

- Zero and sets *out_token* if the function was successful.
- A non-zero value if the function was not successful. You can use the `ErrorNumber` function to obtain an error code, and you can use the `ErrorString` function to obtain a description of the error.

BufferSizeRequired -- Return the required size of the buffer after a buffer overflow error

Format

```
#include <aopapi.h>
unsigned int BufferSizeRequired();
```

Description

The `BufferSizeRequired` function returns the buffer size that is required to complete the last action that failed due to a buffer overflow on the current thread. Call this function when the system error code returned by the `ErrorNumber` function is `API_BUFFER_OVERFLOW`.

Returned Value

`BufferSizeRequired` returns:

- Required buffer size if a buffer overflow occurred.
- An undefined value if a buffer overflow did not occur.

CancelJob -- Cancel a print job

Format

```
#include <aopapi.h>
int CancelJob( unsigned int job_id,
               const char*  user_name );
```

Description

The `CancelJob` function cancels a print job that has been closed on the JES spool but has not yet been selected for printing. It removes all output data sets for the job on the JES spool. Input parameters are:

job_id The job ID returned by the `ClosePrintFile` function or the job ID returned by Infoprint Server if the job was created by some other method.

user_name The user name passed to the `CreatePrintFile` function or the submitter name if the job was created by some other method.

Note: Your application must be APF-authorized to use this function.

Returned Value

`CancelJob` returns:

- Zero if the job was successfully canceled.

- A non-zero value if the job was not successfully canceled. You can use the `ErrorNumber` function to obtain an error code, and you can use the `ErrorString` function to obtain a description of the error.

ClosePrintFile -- Close an Infoprint Server print job

Format

```
#include <aopapi.h>
int ClosePrintFile( JobToken      token,
                    unsigned int* job_id );
```

Description

The `ClosePrintFile` function closes an output data set on the JES spool. If the output data set cannot be closed but *token* refers to a valid job, the output data set is removed from the JES spool as if `AbortPrintFile` had been called. Input parameters are:

token The job token returned by the `CreatePrintFile` function.

job_id A job ID assigned by Infoprint Server and returned by this function. If you call the `GetJobInfo` or `CancelJob` functions, you must pass this job ID as an argument. This job ID is assigned to the output data set on the JES spool; the operator can display this job ID using System Display and Search Facility (SDSF) commands or a comparable product. (JES commands do not display this job ID.)

To limit the range of *job_id* to a number from 1 to 65535, define the environment variable AOP_64K_JOBID before calling the `CreatePrintFile` function.

Returned Value

`ClosePrintFile` returns:

- Zero and sets *job_id* if the output data set was closed successfully.
- A non-zero value if the output data set was not closed successfully. You can use the `ErrorNumber` function to obtain an error code, and you can use the `ErrorString` function to obtain a description of the error.

CreatePrintFile -- Create an Infoprint Server print job

Format

```
#include <aopapi.h>
int CreatePrintFile( const char*  printer_name,
                     const char* user_name,
                     const char* file_name,
                     const char* attributes,
                     JobToken*   out_token    );
```

Description

The `CreatePrintFile` function creates an Infoprint Server print job for a printer that is defined in the Infoprint Server Printer Inventory. This function lets you create only one document in each job. After calling `CreatePrintFile`, call these functions:

1. `WritePrintFile` to write the data.
2. `ClosePrintFile` to close the output data set.

For each successful call to `CreatePrintFile`, you must call either `ClosePrintFile` or `AbortPrintFile`; otherwise, storage is not freed, and the output data set remains open.

Input parameters are:

printer_name The name of a printer definition in the Infoprint Server Printer Inventory; printer definition names are case sensitive.

user_name The name of the owner of the print job. The name can be from 1 to 31 characters. If *user_name* is NULL, then `CreatePrintFile` uses the current user login name returned by the `_getlogin1()` function. The user name becomes the job name of the output data set on the JES spool.

file_name The name of the document. The name can be any length. If *file_name* is NULL, the job identifier is used.

attributes A null-terminated string of job attributes. If *attributes* is NULL, values specified in the printer definition, or default values, are used for all job attributes. Specify attributes in the format *attribute*=*value*, separated by spaces. For more information about the valid job attributes and how to specify them, refer to the description of the `-o` option of the `lp` command and the *Using Job Attributes* chapter of the *z/OS Infoprint Server User's Guide*

Note: If the data to be written is ASCII text data, specify an ASCII code page, such as `ISO8859-1` in the **document-codepage** job attribute.

out_token A job token returned by this function. You must pass this job token as an argument to the `WritePrintFile` , `ClosePrintFile` , and `AbortPrintFile` functions. You can pass the returned job token to functions from threads other than the thread that called the `CreatePrintFile` function.

Note: Your application must be APF-authorized to use this function.

Returned Value

`CreatePrintFile` returns:

- Zero and *out_token* if the print job was successfully created.
- A non-zero value if the function was not successful. You can use the `ErrorNumber` function to obtain an error code, and you can use the `ErrorString` function to obtain a description of the error.

EndEnumJobs -- End the enumeration of Infoprint Server print jobs for a printer

Format

```
#include <aopapi.h>
int EndEnumJobs  ( QueryToken   token );
```

Description

`EndEnumJobs` ends an enumeration of print jobs. The *token* argument must be a valid token returned by the `BeginEnumJobs` function.

Returned Value

`EndEnumJobs` returns:

- Zero if the function was successful.
- A non-zero value if the function was not successful. You can use the `ErrorNumber` function to obtain an error code, and you can use the `ErrorString` function to obtain a description of the error.

EndEnumPrinters -- End the enumeration of Infoprint Server printers

Format

```
#include <aopapi.h>
int EndEnumPrinters  ( QueryToken   token );
```

Description

`EndEnumPrinters` ends an enumeration of Infoprint Server printers. The *token* argument must be a valid token returned by the `BeginEnumPrinters` function.

Returned Value

`EndEnumPrinters` returns:

- Zero if the function was successful.
- A non-zero value if the function was not successful. You can use the `ErrorNumber` function to obtain an error code, and you can use the `ErrorString` function to obtain a description of the error.

EnumJobs -- Enumerate Infoprint Server print jobs for a printer

Format

```
#include <aopapi.h>
int EnumJobs     ( QueryToken       token,
                   void*            buffer,
                   unsigned int     buffer_size,
                   unsigned int     max_count,
                   JobInfo***       job_list );
```

Description

The `EnumJobs` function returns information about all print jobs created by Infoprint Server for a specified printer definition. Job information is returned in `JobInfo` data structures in the buffer you provide to this function. The `JobInfo` data structures are returned in ascending order of the binary value of the job names. The locale has no effect on the ordering.

You can call `EnumJobs` as many times as necessary to obtain information about all jobs for the printer definition; `EnumJobs` returns zero when no more jobs exist for the printer. You must call the `EndEnumJobs` function to end the enumeration of print jobs.

Input parameters are:

token A valid query token returned by the `BeginEnumJobs` function.

buffer A buffer in which this function can return the `JobInfo` data strutures.

buffer_size The number of bytes in the buffer.

max_count The maximum number of jobs for which you want `JobInfo` structures returned.

job_list A pointer to an array of pointers to `JobInfo` data structures. The array of pointers is in the buffer. Do not use the returned array of pointers after the buffer has been freed or modified. The returned value indicates the number of pointers in the array.

Returned Value

`EnumJobs` returns:

- The number of `JobInfo` structures created if the function was successful. Zero indicates that there are no more jobs for this printer definition.

- A negative value if the function was not successful. You can use the `ErrorNumber` function to obtain an error code, and you can use the `ErrorString` function to obtain a description of the error.

EnumPrinters -- Enumerate Infoprint Server printers

Format

```
#include <aopapi.h>
int EnumPrinters     ( QueryToken       token,
                       void*            buffer,
                       unsigned int     buffer_size,
                       unsigned int     max_count,
                       PrinterInfo***   printer_list );
```

Description

The `EnumPrinters` function returns information about all of the printers defined by printer definitions in the Infoprint Server Printer Inventory. Printer information is returned in `PrinterInfo` data structures in the buffer you provide to this function. The `PrinterInfo` data structures are returned in ascending order of the binary value of the printer definition names. The locale has no effect on the ordering.

You can call `EnumPrinters` as many times as necessary to obtain information about all printers; `EnumPrinters` returns zero when information about all printers has been returned. You must call the `EndEnumPrinters` function to end the enumeration of printers.

Input parameters are:

token A valid query token returned by the `BeginEnumPrinters` function.

buffer A buffer in which this function can return the `PrinterInfo` data strutures.

buffer_size The number of bytes in the buffer.

max_count The maximum number of printers for which you want `PrinterInfo` structures returned.

printer_list A pointer returned by this function to an array of pointers to `PrinterInfo` data structures. The array of pointers is in the buffer. Do not use the returned array of pointers after the buffer has been freed or modified. The returned value indicates the number of pointers in the array.

Returned Value

`EnumPrinters` returns:

- The number of `PrinterInfo` structures created if the function was successful. Zero indicates that information about all printers has been returned.
- A negative value if the function was not successful. You can use the `ErrorNumber` function to obtain an error code, and you can use the `ErrorString` function to obtain a description of the error.

ErrorNumber -- Return the API error code

Format

```
#include <aopapi.h>
int ErrorNumber();
```

Description

The `ErrorNumber` function returns a number that indicates the last error returned by the API for the calling thread. The following table lists names defined in the `aopapi.h` file for the error numbers and describes the error. This table does not list all possible error numbers returned by this function. For all errors, you can use the `ErrorString` function to obtain a description of the error.

Name	Description
`AOP_NO_ERROR`	No error occurred.
`AOP_PRINTER_NOT_FOUND`	The printer definition that was specified as an input argument to an API function does not exist in the Printer Inventory.
`AOP_JOB_NOT_FOUND`	The job ID that was specified as an input argument to an API function does not exist.
`AOP_JOB_LIMIT_EXCEEDED`	The `CreatePrintFile` function was called to create a new job, but no free job IDs exist to assign to the job.
`AOP_CANCEL_NOT_PERMITTED`	The specified user is not the owner of the print job. Only the owner is allowed to cancel a print job.
`AOP_INV_CONNECT_ERROR`	The Printer Inventory is not active.
`AOP_NOT_ENABLED`	The use of Infoprint Server is not enabled.
`AOP_NOT_AUTHORIZED`	The action on the specified object is not permitted.
`AOP_INVALID_TOKEN`	The job or query token that was specified as an argument to an API function is not valid.
`AOP_BUFFER_OVERFLOW`	The buffer provided to an API function was not big enough to complete the operation. You can obtain the required buffer size for the operation by calling the `BufferSizeRequired` function.
`AOP_API_INIT_FAILED`	The API could not be initialized. Ensure that `aopapi.dll` can be loaded from one of the directories specified in the LIBPATH environment variable and is accessible to this process.

Returned Value

`ErrorNumber` returns a number that indicates the last error.

ErrorString -- Return a pointer to the API error message

Format

```
#include <aopapi.h>
const char*  ErrorString();
```

Description

The `ErrorString` function returns a pointer to a string that describes the last error returned by the API for the calling thread. The message is a multi-byte string in the language indicated by the current locale when the locale is supported or in English for other locales. The following locales are supported: C, En_US, and Ja_JP.

Returned Value

`ErrorString` returns:

- A pointer to an error string if an error occurred.
- An empty string if no error occurred.

GetJobInfo -- Return the status of an Infoprint Server print job

Format

```
#include <aopapi.h>
JobInfo* GetJobInfo( unsigned int   job_id,
                     void*          buffer,
                     unsigned int   buffer_size );
```

Description

The `GetJobInfo` function returns information about the Infoprint Server print job identified by argument *job_id*. This function returns the `JobInfo` data structure, which describes the job and each document in the job. The `DocumentInfo` data structure, which is part of the `JobInfo` data structure describes each document in the job.

Input parameters are:

job_id The job ID returned by the `ClosePrintFile` function or the job ID returned by Infoprint Server if the job was created by some other method.

buffer A buffer in which this function returns the `JobInfo` data structure.

buffer_size The number of bytes in the buffer.

Returned Value

`GetJobInfo` returns:

- A pointer to a `JobInfo` structure if the function was successful. The address will be within the buffer identified by the *buffer* argument. Do not use this address after you free or modify the buffer.
- A null pointer if the function was not successful. You can use the `ErrorNumber` function to obtain an error code, and you can use the `ErrorString` function to obtain a description of the error.

Example

This example calls the `GetJobInfo` function to get information for the print job identified by job ID 152. It handles a buffer overflow error condition, prints the API error message if another error occurs, and if the `GetJobInfo` function was successful, prints information returned about the job and each document within the job.

```
#include <aopapi.h>
#include <time.h>
#include <stdio.h>
#include <stdlib.h>
#define NULL_TEST(x) ((x) ? (x) : "(null)")
int main()
{
   int           i;
   JobInfo*      job;
   unsigned int  job_number = 152;
   int           buffer_size = 4096;
   void*         buffer = malloc(buffer_size);
   retry: job = GetJobInfo(job_number,
                           buffer,
                           buffer_size);
   if (job == 0) {
      if (ErrorNumber() == AOP_BUFFER_OVERFLOW) {
         buffer_size <<= 1;
         buffer = realloc(buffer, buffer_size);
         goto retry;
      } else {
```

```
            fprintf(stderr, "API error: %s\n", ErrorString());
            exit(1);
         }
      }
      printf("Job %-d:\n", job->id);
      printf("   user name:  %s\n", job->user_name);
      printf("   printer:    %s\n", job->printer_name);
      printf("   submitted:  %s", ctime(>submitted));
      printf("   priority:   %-d\n", job->priority);
      printf("   name text:  %s\n", NULL_TEST(job->name_text));
      for (i=0; i<job->ndocuments; ++i) {
         printf("   document:   %s\n", NULL_TEST(job->documents[i]->name));
         printf("      status:     %d\n", job->documents[i]->status);
         printf("      size:       %-d\n", job->documents[i]->size);
         printf("      format:     %-d\n", job->documents[i]->format);
      }
   }
```

GetPrinterInfo -- Return information about an Infoprint Server printer

Format

```
#include <aopapi.h>
PrinterInfo* GetPrinterInfo( const char*    printer_name,
                             void*          buffer,
                             unsigned int   buffer_size );
```

Description

The `GetPrinterInfo` function returns information about the printer definition in the Infoprint Server Printer Inventory identified by argument *printer_name*. This function returns the `PrinterInfo` data structure.

Input parameters are:

printer_name The name of the printer definition in the Infoprint Server Printer Inventory.

buffer An buffer in which this function returns the `PrinterInfo` data structure.

buffer_size The number of bytes in the buffer.

Returned Value

`GetPrinterInfo` returns:

- A pointer to a `PrinterInfo` structure if the function was successful. The address will be within the buffer identified by the *buffer* argument. Do not use this address after you free or modify the buffer.
- A null pointer if the function was not successful. You can use the `ErrorNumber` function to obtain an error code, and you can use the `ErrorString` function to obtain a description of the error.

InitAPI -- Initialize the API

Format

```
#include <aopapi.h>
int InitAPI();
```

Description

The `InitAPI` function initializes the API. You can call `InitAPI` multiple times from a single process and concurrently from multiple threads. You do not need to explicitly call `InitAPI` because it is called automatically on the first call to any other API

function if it has not been called explicitly. Use this function if your application needs to determine if the API can be successfully initialized separately from other calls.

Returned Value

`InitAPI` returns:

- Zero if the API was successfully initialized.
- A non-zero value if the DLL was not loadable or the API was not successfully initialized for another reason. You can use the `ErrorNumber` function to obtain an error code, and you can use the `ErrorString` function to obtain a description of the error.

ReleaseTransform -- Release a connection with a transform daemon

Format

```
#include <aopapi.h>

typedef enum {
   TRANSFORM_COMPLETE = 2,
   TRANSFORM_ABORT    = 3
} ReleaseType;
int        ReleaseTransform( Transform*, ReleaseType release_type );
```

Description

The `ReleaseTransform` function releases the connection with the transform daemon, which was established by the `RequestTransform` function. Input parameters are:

`Transform*` A pointer to the `Transform` data structure returned by the `RequestTransform` function.

release_type The status of the transform:

- TRANFORM_COMPLETE: This status indicates that the transform completed normally. Specify this status when the transform completes processing, whether successful or not, in a known good state.
- TRANSFORM_ABORT: This status indicates that the transform completed abnormally. Specify this status when the status of the transform is uncertain, due to an unexpected transform failure, a communication failure, and so on.

Returned Value

`ReleaseTransform` returns:

- Zero if the function was successful.
- A non-zero value if the function was not successful. You can use the `ErrorNumber` function to obtain an error code, and you can use the `ErrorString` function to obtain a description of the error.

RequestTransform -- Establish a connection with a transform daemon

Format

```
#include <aopapi.h>
Transform* RequestTransform( const char* transform_class );
```

Description

The `RequestTransform` function establishes a connection with a transform daemon. Call this function only when you are writing a transform that is managed by the Infoprint Server Transform Manager. After the connection is established, you can

use the identifier in field **fd** in the returned `Transform` data structure to receive data from and send data to the transform daemon.

transform_class must match the transform name and class specified in the **transform** attribute in the transform configuration file (**aopxfd.conf**). The format of *transform_class* is: *transformname*[_*transformclass*]. *transform_class* can be a maximum of 63 bytes in length. For information about the transform configuration file, see "Format of a Transform Entry" on page 89.

To release the transform connection, call the `ReleaseTransform` function.

Returned Value

`RequestTransform` returns:

- A pointer to a `Transform` structure if the function was successful.
- A null pointer if the function was not successful.

SetTerminationHandler -- Register a termination function

Format

```
#include <aopapi.h>
int SetTerminationHandler(TerminationHandler handler,
                          void* argument);
```

Description

The `SetTerminationHandler` function registers a function to handle termination of the API threads created within the current process (LE enclave). If the API encounters a condition on one of the threads it manages that would cause enclave termination, then the API calls the termination function pointed to by *handler* with argument *argument*. If the termination function returns without terminating the enclave, then the API terminates the enclave.

To remove the termination handler function, call `SetTerminationHandler` with a null argument for *handler*.

Note: See "Writing a Termination Handler Function" on page 292 for information about how to write a termination handler function.

Returned Value

`SetTerminationHandler` returns:

- Zero if the function was successful.
- A non-zero value if the function was not successful. You can use the `ErrorNumber` function to obtain an error code, and you can use the `ErrorString` function to obtain a description of the error.

TermAPI -- Terminate the API

Format

```
#include <aopapi.h>
int TermAPI();
```

Description

The `TermAPI` function terminates the API. When the number of `TermAPI` calls equals the number of `InitAPI` calls, then all resources associated with the API are freed and the `aopapi.dll` is unloaded. You do not need to explicitly call `TermAPI` because this function is performed automatically when your application terminates.

Returned Value

`TermAPI` returns:

- Zero if the API was successfully terminated.
- A non-zero value if the API was not successfully terminated. You can use the `ErrorNumber` function to obtain an error code, and you can use the `ErrorString` function to obtain a description of the error.

WritePrintFile -- Write data to an Infoprint Server print job

Format

```
#include <aopapi.h>
int WritePrintFile( JobToken      token,
                    const char*   buffer,
                    unsigned int  buffer_size );
```

Description

The `WritePrintFile` function writes data sequentially to an output data set on the JES spool. Infoprint Server determines the data format automatically unless the **document-format** job attribute is passed to the `CreatePrintFile` function. If the printer definition in the Printer Inventory specifies a filter for the data format, then Infoprint Server calls that filter to transform the data before writing it to the JES spool.

If the data format is text data, and if a printer code page is specified in the printer definition, Infoprint Server converts the data from the document code page to the printer code page before writing it to the JES spool. Because the default document code page is determined by the locale and is usually an EBCDIC code page, to print an ASCII document, specify an ASCII code page in the **document-codepage** job attribute when you call the `CreatePrintFile` function. Call the `WritePrintFile` function as many times as necessary to write all the data. After all data has been successfully written, you must call the `ClosePrintFile` function to close the output data set on the JES spool or the `AbortPrintFile` function to remove the data set from the JES spool.

If `WritePrintFile` is not successful for a valid *token*, then the spooling operation associated with *token* is terminated and the output data set is removed from the JES spool as if the `AbortPrintFile` function had been called.

Input parameters are:

token A valid job token returned by the `CreatePrintFile` function.

buffer A buffer containing the data to be written to the output data set on the JES spool.

buffer_size The number of bytes in the buffer.

Returned Value

`WritePrintFile` returns:

- Zero if the data is written successfully.
- A non-zero value if data is not successfully written.

Programming Considerations

This section describes considerations for writing and link-editing your C or C++ applications that use the API.

Redefining API Function Names

The names of the API functions are generic, for example `InitAPI` and `TermAPI`. These function names are mapped (using the `map` pragma) to names that are prefixed with AOP to reduce the chance of conflicts with your applications. For example:

```
#pragma map(InitAPI, "AOPINIT")
```

If the name of an API function conflicts with a function name defined in the same scope, you can redefine the API function name before including the API header file (`aopapi.h`). For example, you could redefine `InitAPI` to `AOP_InitAPI`, as shown in the following example:

```
#define InitAPI AOP_InitAPI
#include <aopapi.h>
```

Linking to API Functions

To link to the functions in this API, include the static library `libaopapi.a` in the link-edit for your application. The functions in `libaopapi.a` are stubs that call the actual code, which is in executable file `aopapi.dll`.

APF-Authorizing Your Application

Your application must run AFP authorized to use the `CancelJob` or `CreatePrintFile` functions. If your application is an HFS program object, use the z/OS UNIX command **extattr** to APF authorize it; also, link your program with the AC=1 attribute.

Writing a Termination Handler Function

If your application needs to perform cleanup before an API-created thread terminates abnormally, you can write your own termination handler function. The API calls your termination handler when an abnormal-termination situation occurs.

If you write a termination handler function, you must call the `SetTerminationHandler` function to register your termination handler.

Your termination handler receives three parameters:

handler_argument	The argument that was passed to the `SetTerminationHandler` function.
cleanup_handler	A pointer to a cleanup handler function that your termination handler must call.
cleanup_argument	An argument that the termination handler must pass to the cleanup handler.

Your termination handler function should perform these functions:

- Produce a dump or equivalent diagnostic of the process to help diagnose problems in your application and the API.
- Perform any desired cleanup actions. All actions permitted by an LE-condition handler are allowed in the termination handler.
- Call the cleanup handler with *cleanup_argument*. This call must be the last operation in the termination handler.
- Either (1) call the `exit` function to terminate the process or (2) simply return, which causes the function that initiated the API thread to fail but allows the process to continue.

Compile the executable code that contains the termination handler function with either the NODLL or DLL(CBA) option. The CBA sub-option is required if the code is a DLL because the cleanup handler function pointer passed to the termination handler is compiled with the NODLL option.

The `aopapi.h` header file contains the following definitions for the termination handler and cleanup handler functions:

```
typedef void (*CleanupHandler)     (void*              cleanup_argument );
typedef void (*TerminationHandler) (void*              handler_argument,
                                    CleanupHandler     cleanup_handler,
                                    void*              cleanup_argument );
```

Customization Requirements

Before you run an application that uses the API, you must perform the following tasks to customize Infoprint Server:

- Customize the Printer Inventory Manager component. See Chapter 3, “Customizing the Printer Inventory Manager” on page 25
- Customize the Print Interface component of Infoprint Server. You need to perform only the general customization tasks for the Print Interface component. See Chapter 4, “Customizing Print Interface” on page 49.
- Customize the Infoprint Server Transforms if you want to transform data from one format to another. Also customize the Transform Manager if the transform is one that is managed by the Transform Manager. See Chapter 5, “Customizing Infoprint Server Transforms and the Transform Manager” on page 83.

In addition, perform the following customization task related to the API:

- Optionally define environment variable `AOP_64K_JOBID`. If this environment variable is defined, the `CreatePrintFile` function limits the job ID of a new print job to the range of 1 to 65535. If this environment variable is not defined, the job ID can be in the range of 1 to 999999.

  Note: This variable is defined for use by the z/OS SMB server. IBM recommends that other applications do *not* define this variable because system performance might be degraded.

After customizing Infoprint Server, perform the following tasks described in *z/OS Infoprint Server Operation and Administration*.

1. Start the Printer Inventory Manager daemon. If you want to transform data from one format to another, and if the transform is one that is managed by the Transform Manager, also start the Transform Manager daemon.
2. Create a printer definition in the Infoprint Server Printer Inventory for each target printer and specify the printer attributes required by the Print Interface component. If you want to transform data, also specify the names of the transform filters in the printer definitions.

Appendix F. Accessibility

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully. The major accessibility features in z/OS enable users to:

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

Using Assistive Technologies

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

Keyboard Navigation of the User Interface

Users can access z/OS user interfaces using TSO/E or ISPF. Refer to *z/OS TSO/E Primer*, *z/OS TSO/E User's Guide*, and *z/OS ISPF User's Guide Volume I* for information about accessing TSO/E and ISPF interfaces. These guides describe how to use TSO/E and ISPF, including the use of keyboard shortcuts or function keys (PF keys). Each guide includes the default settings for the PF keys and explains how to modify their functions.

© Copyright IBM Corp. 1995, 2002

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

© Copyright IBM Corp. 1995, 2002

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
Mail Drop 001W
Boulder, CO 80301
U.S.A

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Programming Interfaces

This publication documents programming interfaces intended to allow you to write programs to obtain the services of Infoprint Server.

Trademarks

The following terms appear in this publication and are either trademarks or registered trademarks of International Business Machines Corporation in the United States, in other countries, or both:

AFP
AIX
Application System/400
CICS
eNetwork
IBM
IBM @server
IMS
IMS/ESA
Infoprint
Intelligent Printer Data Stream
IP PrintWay
IPDS
Language Environment
MVS/ESA
NetSpool
OpenEdition
Operating System/2
OS/2
OS/390
OS/400
Print Services Facility
PrintWay
RACF
RETAIN
RS/6000
S/390
VTAM
WIN-OS/2

z/OS
z/OS.e
z/VM
zSeries

The following terms appear in this publication and are either trademarks or registered trademarks of subsidiaries of International Business Machines Corporation in the United States, in other countries, or both:

- Freelance is a trademark of Lotus Development Corporation. Lotus is a registered trademark of Lotus Development Corporation.

The following terms appear in this publication and are either trademarks or registered trademarks of other companies in the United States, in other countries, or both:

- Java is a registered trademark of Sun Microsystems, Inc.
- MCS is a registered trademark of Intel.
- Microsoft, Windows, and Windows NT are registered trademarks of Microsoft Corporation.
- UNIX is a registered trademark of The Open Group in the United States and other countries.

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

Glossary

Sources

This glossary defines technical terms and abbreviations used in Infoprint Server documentation. If you do not find the term you are looking for, see the index of this publication or view *IBM Glossary of Computing Terms*, located at: http://www.ibm.com/ibm/terminology

Definitions reprinted from *American National Standard Dictionary for Information Systems*, ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI), are identified by the symbol (A) following the definition. Copies can be purchased from the American National Standards Institute, 11 West 42nd Street, New York, New York 10036.

Definitions reprinted from a published section of the International Organization for Standardization (ISO) *Vocabulary—Information Processing* or from a published section of *Vocabulary—Office Machines* developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Committee (ISO/IEC JTC1/SC1) are identified by the symbol (I) following the definition. Because many ISO definitions are also reproduced in *American National Standare Dictionary for Information Systems,* ISO definitions may also be identified by the symbol (A).

Definitions reprinted from working documents, draft proposals, or draft international standards of ISO Technical Committee 97, Subcommittee 1 (Vocabulary), Joint Technical Committee 1 are identified by the symbol (T) following the definition, indicating that final agreement has not yet been reached among its participating members.

This publication contains terms and terms and definitions from *Information Technology—Portable Operating System Interface (POSIX)* from the POSIX series of standards for applications and user interfaces to open systems, copyrighted by the Institute of Electrical and Electronics Engineers (IEEE). Copies of all POSIX drafts and standards may be purchased from IEEE at 1-800-678-IEEE.

- Definitions identified by (POSIX.0) are from *Part 0: Standards Project, Draft Guide to the POSIX Open System Environment* P1003.0 Draft 15 (June 1992), an unapproved draft subject to change.
- Definitions identified by (POSIX.1) are from *Part 1: System Application Program Interface (API) [C Language]* approved September 28, 1990, as IEEE Std 1003.1-1990 by the IEEE Standards Board, and adopted in 1990 as an International Standard (ISO/IEC 9945-1: 1990) by the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC).
- Definitions identified by (POSIX.2) are from *Part 2: Shell and Utilities* P1003.2.

Definitions that are specific to IBM products are so labeled; for example, “In TCP/IP,” or “In Infoprint Server.”

References

The following cross-references are used in this glossary:

Contrast with. This refers to a term that has an opposite or substantively different meaning.

See. This refers to multiple-word terms in which this term appears.

See also. This refers to related terms that have similar, but not synonymous, meanings.

Synonym for. This appears in the commentary of a less desirable or less specific term and identifies the preferred term that has the same meaning.

Synonymous with. This appears in the commentary of a preferred term and identifies less desirable or less specific terms that have the same meaning.

Numerics

3270 data stream. Data transferred from or to an allocated primary or tertiary device, or to the host system, as a continuous stream of data and 3270 Information Display System control elements in character form.

A

abend. Termination of a task before its completion because of an error condition that cannot be resolved by recovery facilities while the task is executing.

© Copyright IBM Corp. 1995, 2002

ACB. Access method control block.

access method control block (ACB). A control block that links an application program to VTAM.

ACIF. (1) AFP conversion and indexing facility. (2) A PSF utility program that converts a print file into AFP, MO:DCA-P, creates an index file for input data, and collects resources used by an AFP document into a separate file.

Advanced Function Presentation (AFP). A set of licensed programs, together with user applications, that use the all-points-addressable concept to print on presentation devices. AFP includes creating, formatting, archiving, retrieving, viewing, distributing, and printing information.

AFP. Advanced Function Presentation.

AFP Printer Driver for Windows. A printer driver that runs on a Windows workstation and creates output in AFP format for printing on AFP printers.

AFP Viewer plug-in for Windows. A program that runs on a Windows workstation and allows you to view files in AFP format from a Web browser.

AIX operating system. IBM's implementation of the UNIX operating system. The RS/6000® system, among others, runs the AIX operating system.

alphanumeric character. A letter or a number.

APPLID. Identifier of a VTAM application program defined by a VTAM APPL resource statement.

ASCII (American Standard Code for Information Interchange). The standard code, using a coded character set consisting of 7-bit coded characters (8-bit including parity check), that is used for information interchange among data processing systems, data communication systems, and associated equipment. The ASCII set consists of control characters and graphic characters. (A)

Note: IBM has defined an extension to ASCII code (characters 128–255).

B

banner page. A page printed before the data set is printed.

binary data. (1) Any data not intended for direct human reading. Binary data may contain unprintable characters, outside the range of text characters. (2) A type of data consisting of numeric values stored in bit patterns of 0s and 1s. Binary data can cause a large number to be placed in a smaller space of storage.

BIND. In SNA, a request to activate a session between two logical units (LUs).

broadcast. (1) Transmission of the same data to all destinations. (T) (2) Simultaneous transmission of data to more than one destination.

buffer. A portion of storage used to hold input or output data temporarily.

burst. To separate continuous-forms paper into single sheets.

C

carriage control character. An optional character in an input data record that specifies a write, space, or skip operation.

carriage return (CR). (1) A keystroke generally indicating the end of a command line. (2) In text data, the action that indicates to continue printing at the left margin of the next line. (3) A character that will cause printing to start at the beginning of the same physical line in which the carriage return occurred.

case-sensitive. Pertaining to the ability to distinguish between uppercase and lowercase letters.

catalog. (1) A directory of files and libraries, with reference to their locations. (2) To enter information about a file or a library into a catalog. (3) The collection of all data set indexes that are used by the control program to locate a volume containing a specific data set.

CICS. Customer Information Control System.

client. A functional unit that receives shared services from a server. See also *client-server*.

client-server. In TCP/IP, the model of interaction in distributed data processing in which a program at one site sends a request to a program at another site and awaits a response. The requesting program is called a client; the answering program is called a server.

code page. (1) A table showing codes assigned to character sets. (2) An assignment of graphic characters and control function meanings to all code points. (3) Arrays of code points representing characters that establish ordinal sequence (numeric order) of characters. (4) A particular assignment of hexadecimal identifiers to graphic elements.

code point. A 1-byte code representing one of 256 potential characters.

coexistence. Two or more systems at different levels (for example, software, service or operational levels) that share resources. Coexistence includes the ability of a system to respond in the following ways to a new function that was introduced on another system with which it shares resources: ignore a new function, terminate gracefully, support a new function.

connection. In TCP/IP, the path between two protocol applications that provides reliable data stream delivery service. In Internet communications, a connection extends from a TCP application on one system to a TCP application on another system.

copy group. (1) One or more copies of a page of paper. Each copy can have modifications, such as text suppression, page position, forms flash, and overlays. (2) An internal object in a form definition or a print data set that controls such items as modifications to a form, page placement, and overlays.

Customer Information Control System (CICS). An IBM licensed program that enables transactions entered at remote terminals to be processed concurrently by user-written application programs. It includes facilities for building, using, and maintaining databases.

D

daemon. A program that runs unattended to perform a standard service. Some daemons are triggered automatically to perform their task; others operate periodically.

data set. The major unit of data storage and retrieval, consisting of a collection of data in one of several prescribed arrangements and described by control information to which the system has access.

data stream. (1) All information (data and control commands) sent over a data link usually in a single read or write operation. (2) A continuous stream of data elements being transmitted, or intended for transmission, in character or binary-digit form, using a defined format.

DBCS. Double-byte character set.

default. A value, attribute, or option that is assumed when no alternative is specified by the user.

directory. (1) A type of file containing the names and controlling information for other files or other directories. Directories can also contain subdirectories, which can contain subdirectories of their own. (2) A file that contains directory entries. No two directory entries in the same directory can have the same name. (POSIX.1). (3) A file that points to files and to other directories. (4) An index used by a control program to locate blocks of data that are stored in separate areas of a data set in direct access storage.

DLL filter. A filter that provides one or more of these functions in a dynamic load library - init(), prolog(), process(), epilog(), and term(). See **cfilter.h** and **cfilter.c** in the **/usr/lpp/Printsrv/samples/** directory for more information. See also filter. Contrast with DLL filter.

dotted decimal notation. The syntactical representation for a 32-bit integer that consists of four 8-bit numbers written in base 10 with periods (dots) separating them. It is used to represent IP addresses.

double-byte character set (DBCS). A set of characters in which each character is represented by a two-bytes code. Languages such as Japanese, Chinese, and Korean, which contain more symbols than can be represented by 256 code points, require double-byte character sets. Because each character requires two bytes, the typing, display, and printing of DBCS characters requires hardware and programs that support DBCS. Contrast with *single-byte character set*.

download. To transfer data from one computer for use on another one. Typically, users download from a larger computer to a diskette or fixed disk on a smaller computer or from a system unit to an adapter.

Download for OS/390. A feature of PSF for OS/390 that allows the PSF program to automatically send data sets from the JES spool, without formatting them, directly to either Infoprint Manager for AIX or EDMSuite OnDemand, using the TCP/IP protocol. Infoprint Manager for AIX and EDMSuite OnDemand servers receive the data sets into files, which can be automatically formatted and printed by Infoprint Manager for AIX or loaded into EDMSuite OnDemand.

drain. An operator action to halt the flow of jobs to a printer, usually to stop the printer or to change print options.

E

EBCDIC. Extended binary-coded decimal interchange code. A coded character set consisting of 8-bit coded characters. (A)

encryption. In computer security, the process of transforming data into an unintelligible form in such a way that the original data either cannot be obtained or can be obtained only by using a decryption process.

environment variable. (1) A name associated with a string of characters, made available to the programs that you run. (2) A variable that describes the operating environment of the process and typically includes information about the home directory, command search path, the terminal in use, and the current time zone. (3) A variable included in the current software environment that is available to any called program that requests it.

F

file. (1) A set of related records treated as a unit. (2) A collection of related data that is stored and retrieved by an assigned name. (3) Linear data that can be opened, written, read, and closed. A file can also contain information about the file, such as authorization

information. The name used to obtain a file includes the directories in the path to the file. (4) Strings of characters with no additional structure. Structure is assumed only by the processing programs. Files can be located relative to the current directory or by an absolute pathname. (5) An object that can be written to, or read from, or both. A file has certain attributes, including access permissions and type. File types include regular file, character special file, block special file, FIFO special file, and directory. Other types of files may be defined by the implementation. (POSIX.1) In the z/OS UNIX System Services implementation, the file system does not support block special files, but it does support symbolic link files. (6) A collection of information or data that is organized by some method (relative, indexed, or serial, for example) and stored on a device such as a disk.

file system. (1) A collection of files and directories. (2) The collection of files and file management structures on a physical or logical mass storage device, such as a disk or disk partition. A single device can contain several file systems. (3) A mountable subtree of the directory hierarchy. (4) A collection of files and certain of their attributes. A file system provides a name space for file serial numbers referring to those files. (POSIX.1).

filter. In Print Interface, a program that can add, delete, or modify input data before Print Interface writes the data to the JES spool. Print Interface provides support for two types of filter programs: DLL filters and UNIX filters. See also *DLL filter* and *UNIX filter*.

font. (1) A family or assortment of characters of a given size and style; for example, 9 point Bodoni Modern. (A) (2) One size and one typeface in a particular type family, including letters, numerals, punctuation marks, special characters, and ligatures. (3) A paired character set and code page that can be used together for printing a string of text characters. A double-byte font can consist of multiple pairs of character sets and code pages.

form definition. A resource used by PSF that defines the characteristics of the form, which includes such functions as overlays to be used (if any), paper source (for cut-sheet printers), duplex printing, text suppression, the position of MO:DCA-P data on the form, and the number and modifications of a page.

FSA. Functional subsystem application.

FSS. Functional subsystem.

functional subsystem (FSS). An address space uniquely identified as performing a specific function related to the JES.

functional subsystem application (FSA). The functional application program managed by the functional subsystem.

H

hexadecimal. (1) Pertaining to a selection, choice, or condition that has 16 possible different values or states. (I) (2) Pertaining to a fixed-radix numeration system, with radix of 16. (I) (3) Pertaining to a system of numbers to the base 16; hexadecimal digits range from 0 through 9 and A through F, where A represents 10 and F represents 15.

hiperspace. A range of up to two gigabytes of contiguous virtual storage addresses that a program can use as a buffer. Like a data space, a hiperspace can hold user data; it does not contain common areas or system data. Instructions do not execute in a hiperspace. Unlike an address space or a data space, data is not directly addressable. To manipulate data in a hiperspace, you bring the data into the address space in 4K byte blocks.

HFS file. An object that exists within a DFSMS mountable file system.

home directory. (1) The current directory associated with the user at the time of login. (POSIX.2) (2) A directory associated with an individual user. (3) The user's current directory on login or after issuing the **cd** command with no argument.

host. In the Internet suite of protocols, an end system. The end system can be any system; it does not have to be a mainframe.

host address. See *IP address*.

host name. In the Internet suite of protocols, the name given to a machine. Sometimes, ″host name″ is used to mean *fully qualified domain name*; other times, it is used to mean the most specific subname of a fully qualified domain name. For example, if *boulder.vnet.ibm.com* is the fully qualified domain name, either of the following may be considered the host name:

- boulder.vnet.ibm.com
- boulder

I

IMS. Information Management System

Infoprint Port Monitor for Windows. A component of Infoprint Server that runs on a Windows 95/98 or Windows NT workstation and sends a file for printing to Print Interface running on the z/OS system.

Infoprint Server. An element of z/OS V1R1 and higher that supports printing on z/OS printers, including local printers and remote printers in a TCP/IP network. Infoprint Server lets users submit print requests from remote workstations in a TCP/IP network, from z/OS UNIX System Services applications, from batch

applications, and from VTAM applications, such as CICS or IMS applications. Infoprint Server consists of the following components:

- IP PrintWay
- NetSpool
- Print Interface
- Printer Inventory Manager
- Transform Manager and Infoprint Server Transforms
- SNMP subagent
- Windows client: Infoprint Port Monitor for Windows

Information Management System (IMS). A database/data communication system that can manage complex databases and networks.

inline resource. A resource contained in the print data set.

internal copy group. A copy group within a print data set instead of within a form definition. See also *copy group*.

Internet. A wide area network connecting thousands of disparate networks in industry, education, government, and research. The Internet network uses TCP/IP as the protocol for transmitting information.

Internet Printing Protocol (IPP). An application-level protocol that enables distributed printing on the Internet. IPP uses a client/server architecture and defines the interactions between IPP clients (typically work-stations) and IPP servers.

Internet Protocol (IP). A protocol used to route data from its source to its destination in an Internet environment.

IP. Internet Protocol.

IP address. (1) In the Internet suite of protocols, the 32-bit address of a machine, expressed in dotted decimal notation, for example, 9.99.9.143. (2) Host name.

IPP. Internet Printing Protocol.

IP PrintWay. A component of Infoprint Server that transmits output data sets from the JES spool to printers in a TCP/IP and SNA network. Also called PrintWay.

J

JCL. Job control language.

JES. Job entry subsystem.

JES2. An z/OS subsystem that receives jobs into the system, converts them to internal format, selects them for execution, processes their output, and purges them from the system. In an installation with more than one processor, each JES2 processor independently controls its job input, scheduling, and output processing.`

JES3. An z/OS subsystem that receives jobs into the system, converts them to internal format, selects them for execution, processes their output, and purges them from the system. In complexes that have several loosely coupled processing units, the JES3 program manages processors so that the global processor exercises centralized control over the local processors and distributes jobs to them via a common job queue.

job control language (JCL). A language of control statements used to identify a computer job or describe its requirements to an operating system.

job entry subsystem (JES). An z/OS subsystem that receives jobs into the system, converts them to internal format, selects them for execution, processes their output, and purges them from the system.

K

Kanji. A Japanese ideographic alphabet. In Kanji, each character is represented by 2 bytes.

kilobyte (KB). (1) For processor storage, real and virtual storage, and channel volume, 1024 bits. (2) For disk storage capacity and communications volume, 1000 bytes.

L

LAN. local area network.

line data. Data prepared for printing on a line printer such as a 3800 Model 1 Printing Subsystem. Line data is usually characterized by carriage-control characters and table reference characters. Contrast with *MO:DCA-P data*.

line printer daemon (LPD). The printer server that allows other hosts to access its printer.

line printer requester (LPR). A client that allows the local host to submit a data set for printing on a remote printer server.

local area network (LAN). A computer network located on a user's premises within a limited geographical area. Communication within a local area network is not subject to external regulations; however, communication across the LAN boundary may be subject to some form of regulation.

locale. (1) A description of a cultural environment. (POSIX.0). (2) The definition of the subset of a user's environment that depends on language and cultural conventions. (POSIX.2).

logical printer. In NetSpool, the target of the VTAM print data, which acts as the secondary LU (SLU) on the session.

logical unit (LU). A type of VTAM network accessible unit that enables end users to gain access to network resources and communicate with each other.

logon mode. In VTAM, a subset of session parameters specified in a logon-mode table for communication with a logical unit. See also *session parameters*.

logon-mode table. In VTAM, a set of entries for one or more logon modes. Each logon mode is identified by a logon mode name.

LPD. line printer daemon

LPR. line printer requester

LU. Logical unit.

LU type. The classification of an LU in terms of the specific subset of SNA protocols and options it supports for a given session, namely:

- The mandatory and optional values allowed in the session activation request
- The usage of data stream controls, function management headers, request unit parameters, and sense data values
- Presentation services protocols such as those associated with FMH usage

LU types 0, 1, 2, 3, 4, 6.1, 6.2, and 7 are defined.

M

Management Information Base (MIB). A logical database made up of the configuration, status, and statistical information stored at a device.

megabyte (MB). (1) For processor storage, real and virtual storage, and channel volume, 1 048 576 bytes. (2) For disk storage capacity and communications volume, 1 000 000 bytes.

MIB. See *Management Information Base*.

migration. Activities that relate to the installation of a new version or release of a program to replace an earlier level. Completion of these activities ensures that the applications and resources on your system will function correctly at the new level.

MO:DCA-P data. Print data that has been composed into pages. Text formatting programs can produce composed text data consisting entirely of structured fields.

MVS/ESA™. Multiple Virtual Storage/Enterprise System Architecture.

N

NCP. Network Control Program.

NetSpool. A 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.

network. A collection of data processing products that are connected by communication lines for information exchange between locations.

Network Control Program (NCP). An IBM licensed program that provides communication controller support for single-domain, multiple-domain, and interconnected network capability.

Network Print Facility (NPF). In eNetwork™ Communications Server, a feature that routes VTAM, JES2, or JES3 printer output to printers in a TCP/IP network.

NPF. Network Print Facility.

NPM. Network Printer Manager.

Network Printer Manager (NPM) for the Web. IBM Network Printer Manager (NPM) for the Web lets network administrators monitor, control, and configure IBM network printers. NPM also lets network administrators monitor some aspects of printers controlled by PSF for OS/390 and other manufacturers' network printers that comply with RFC 1759.

O

OnDemand. A client/server application that you can use to replace hard copy reports and microfiche, and provide fast, online access to information. An OnDemand server manages electronic archives of reports and documents. You can use an OnDemand client program to search for information and view, print, and FAX copies of reports and documents.

OpenEdition®. See z/OS UNIX System Services.

options data set. In IP PrintWay prior to OS/390 V2R8, a VSAM data set containing one or more options entries. Each options entry contains transmission options used by IP PrintWay. to transmit data sets to a print queue in a TCP/IP network. Each options entry can also contain NetSpool parameters that specify data-set characteristics for use by NetSpool.

output writer. A part of the job entry subsystem that receives job output from the system spool.

output data set. (1) A data set that a program opens so that it can write to that file. (2) A file that contains the results of processing.

output writer. A part of the job entry subsystem that receives job output from the system spool.

P

page definition. A resource used by PSF and the Infoprint Server Transforms that defines the rules of transforming line data into MO:DCA-P data and text controls.

page-format table. In NetSpool prior to OS/390 V2R8, a table that defines page-formatting values NetSpool uses for SCS data streams that do not contain SHF (Set Horizontal Format) or SVF (Set Vertical Format) commands. The table can contain several entries, each entry containing a different set of page-formatting values.

parameter. Information that the user supplies to a panel, command, or function.

partitioned data set (PDS). A data set in direct access storage that is divided into partitions, called members, each of which can contain a program, part of a program, or data. Synonymous with program library.

pathname. (1) A filename specifying all directories leading to the file. (2) A filename specifying all directories leading to a file plus the filename itself. (3) A string that is used to identify a file. A pathname consists of, at most, (PATH_MAX) bytes, including the terminating null character. It has an optional beginning slash, followed by zero or more filenames separated by slashes. If the pathname refers to a directory, it may also have one or more trailing slashes. Multiple successive slashes are considered to be the same as one slash. A pathname that begins with two successive slashes may be interpreted in an implementation-defined manner, although more than two leading slashes shall be treated as a single slash. (POSIX.1). In the z/OS UNIX System Services implementation, the C/370™ functions **fopen()**, **freopen()**, **remove()**, and **rename()** interpret names with exactly two leading slashes, no leading blanks or other characters, and the third character not a slash to mean that the rest of the name refers to a traditional MVS™ data set.

pel. Picture element.

Picture element (pel, pixel). (1) In computer graphics, the smallest element of a display surface that can be independently assigned color and intensity. (T). (2) The area of the finest detail that can be reproduced effectively on the recording medium. (3) An element of a raster pattern about which a toned area on a photoconductor can appear. (4) The addressable unit on a 3800 Printing System Model 3 or 8.

PIDU. Printer Inventory Definition Utility.

PLU. Primary logical unit.

port. (1) A part of the system unit or remote controller to which cables for external devices (display stations, terminals, or printers) are attached. The port is an access point for data entry or exit. (2) A specific communications end point within a host. A port is identified by a port number.

POSIX. Portable Operating System Interface for Computer Environments, an interface standard governed by the IEEE and based on UNIX. POSIX is not a product; rather, it is an evolving family of standards describing a wide spectrum of operating system components ranging from C language and shell interfaces to system administration.

PostScript. A page description language with graphics capabilities that was developed by Adobe Systems, Incorporated.

primary logical unit (PLU). In SNA, the logical unit (LU) that sends the BIND to activate a session with its partner LU. Contrast with *secondary logical unit*.

print queue. A list of items waiting to be printed.

Print Services Facility™ (PSF). A licensed program that manages and controls the input data stream and output data stream required by supported IBM page printers. PSF combines print data with other resources and printing controls to produce AFP output.

Print Interface. A component of 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.

printer definition. In Infoprint Server, an entry in the Printer Inventory that contains information about a printer or set of printers that share the same characteristics. A printer definition contains information that Infoprint Server uses to print files.

Printer Inventory. In Infoprint Server, a set of files that contain printer definitions for printers. Each printer definition is identified with a unique printer name; the job submitter selects the printer name when printing a file. The printer definitions contain information that Infoprint Server uses to print files.

Printer Inventory Definition Utility. In Infoprint Server, a utility program that creates objects in the Printer Inventory.

Printer Port Monitor. See *Infoprint Port Monitor for Windows*.

PrintWay. See *IP PrintWay*.

protocol. A set of semantic and syntactic rules that determines the behavior of functional units in achieving communication.

PSF. Print Services Facility.

PSF/6000. An intelligent printer driver that provides AFP capabilities for the AIX operating system on the

RS/6000 system. AFP capabilities include electronic forms, images, graphics, and typographical fonts. Also called PSF for AIX.

R

RACF. Resource Access Control Facility

Request for Comments (RFC). In Internet communications, the document series that describes a part of the Internet suite of protocols and related experiments. All Internet standards are documented as RFCs.

request unit (RU). A message unit that contains control information, end-user data, or both.

resource. A collection of printing instructions used by Print Services Facility in addition to the print data set, to produce the printed output. PSF resources include coded fonts, font character sets, code pages, page segments, overlays, form definitions, and page definitions.

Resource Access Control Facility (RACF). An IBM-licensed product that provides for access control by identifying and verifying users to the system, authorizing access to protected resources, logging detected unauthorized attempts to enter the system, and logging detected accesses to protected resources.

response unit (RU). A message unit that acknowledges a request unit. It may contain prefix information received in a request unit. If positive, the response unit can contain additional information (such as session parameters in response to BIND SESSION). If negative, the response unit contains sense data defining the exception condition.

Restructured Extended Executor (REXX). A general-purpose, procedural language for end-user personal programming, designed for ease by both casual general users and computer professionals. It is also useful for application macros. REXX includes the capability of issuing commands to the underlying operating system from these macros and procedures. Features include powerful character-string manipulation, automatic data typing, manipulation of objects familiar to people, such as words, numbers, and names, and built-in interactive debugging.

retain time. In IP PrintWay, the length of time to keep a data set on the JES spool after either a successful transmission to the destination or a failed transmission, after retrying the transmission the number of times specified in the retry limit. You can specify a retain time for 2 different situations:

- Retain time for data sets that have been successfully transmitted
- Retain time for data sets whose transmission has failed

retry limit. In IP PrintWay, the maximum number of retries that IP PrintWay is to attempt.

retry time. In IP PrintWay, the time between two attempts to send the data set to its destination.

REXX. Restructured Extended Executor

RFC. Request for Comments.

routing data set. In IP PrintWay prior to OS/390 V2R8, a VSAM data set containing a routing entry for each print queue to which IP PrintWay can transmit output data sets. Each entry contains the name of the remote print queue, the IP address or name of the print queue's host system, the name of an options entry, and other routing information. Each routing entry can also define a NetSpool logical printer.

RU. Request/response unit.

S

SBCS. Single-byte character set.

SCS. SNA Character String.

SDSF. System Display and Search Facility.

secondary logical unit (SLU). In SNA, the logical unit (LU) that receives the BIND request to establish a session with its partner LU. Contrast with *primary logical unit*.

sense code. In SNA, the data sent with a negative response, indicating the reason for the response.

sequential data set. (1) A data set whose records are organized on the basis of their successive physical positions, such as on magnetic tape. (2) A data set in which the contents are arranged in successive physical order and are stored as an entity. The data set can contain data, text, a program, or part of a program. Contrast with *partitioned data set (PDS)*.

server. (1) On a network, the computer that contains the data or provides the facilities to be accessed by other computers on the network. (2) A program that handles protocol, queuing, routing, and other tasks necessary for data transfer between devices in a computer system.

Server Message Block (SMB). (1) A protocol for remote file and print access used by Windows clients. This protocol is also known as Common Internet File System (CIFS). (2) A program that handles protocol, queuing, routing, and other tasks necessary for data transfer between devices in a computer system.

session. A logical connection between two network accessible units that can be activated, tailored to provide various protocols, and deactivated, as requested.

session parameters. In SNA, the parameters that specify or constrain the protocols, for a session between two network addressable units (NAUs).

shell script. A file of shell commnds. If the file is executable; a user can run it by specifying the file's name as a shell command or as an operand on **sh** or on the TSO/E OMVS command. A shell script is like a TSO/E REXX program.

shift-out, shift-in (SOSI). Special EBCDIC or ASCII characters in the data stream that indicate switches between double-byte and single-byte fonts.

Simple Network Management Protocol (SNMP). A protocol that enables a management station to configure, monitor, and receive trap messages from network devices.

single-byte character set (SBCS). A set of characters in which each character is represented by a one-byte code. Contrast with *double-byte character set*.

SLU. Secondary logical unit.

SMB. Server Message Block.

SMF. System Management Facility.

SNA. Systems Network Architecture.

SNA Character String (SCS). In SNA, a character string composed of EBCDIC controls, optionally intermixed with end-user data, that is carried within a request/response unit.

SNMP. See *Simple Network Management Protocol*.

SNMP agent. Software that enables a device to respond to manager requests to view or update Management Information Base (MIB) data, and send traps reporting problems or significant events.

SNMP Manager. In SNMP, software in a network management station that enables the station to send requests to view or update MIB variables, to send and receive inform-requests, and to receive traps from an agent.

SOSI. See *shift-out, shift-in*.

spool. Simultaneous peripheral operation online.

startup procedure. JCL for a procedure to start an application (or, NetSpool and IP PrintWay).

superuser. A system user who operates without restrictions. A superuser has the special rights and privileges needed to perform administrative tasks.

syntax. The grammatical rules for constructing a command,

System Display and Search Facility (SDSF). An IBM-licensed program that provides a menu-driven full screen interface to obtain detailed information about the jobs and resources in an MVS/JES2 system.

System Management Facility (SMF). A standard feature of z/OS that collects and records a variety of system and job-related information.

Systems Network Architecture (SNA). The description of the logical structure, formats, protocols, and operational sequences for transmitting information units through, and controlling the configuration and operation of, networks.

T

table reference character (TRC). A numeric character corresponding to the order in which font character sets have been specified with the **chars** job attribute or in the page definition used to print a job. It is used to select a font character set during printing.

TCP. Transmission Control Protocol.

TCP/IP. Transmission Control Protocol/Internet Protocol.

Telnet. In the Internet suite of protocols, a protocol that provides remote terminal connection service. It allows users of one host to log on to a remote host and interact as directly attached terminal users of that host.

Time Sharing Option (TSO). An operating system option that provides interactive time sharing from remote terminals.

trace. A record of the execution of a computer program. It exhibits the sequences in which the instructions were executed. (A)

transform. A program that converts 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.

Transmission Control Protocol (TCP). A communications protocol used in Internet and in any network that follows the U.S. Department of Defense standards for inter-network protocol. TCP provides a reliable host-to-host protocol between hosts in packet-switched communications networks and in interconnected systems of such networks. It assumes that the Internet protocol is the underlying protocol.

transmission-queue data set. In IP PrintWay, a data set containing an entry for each data set that IP PrintWay is to transmit to the remote system or that IP PrintWay is retaining on the JES spool.

transparent data. (1) Data that is of no significance to the receiver. (2) Data that can contain any hexadecimal value.

trap. A message that reports a problem or a significant event.

TRC. Table reference character.

U

UCS. Universal character set.

universal character set (UCS). A printer feature that permits the use of a variety of character arrays. Synonymous with *font*.

UNIX. A highly portable operating system originally developed by Bell Laboratories that features multiprogramming in a multiuser environment. UNIX is implemented in the C language. UNIX was originally developed for use on minicomputers but has been adapted on mainframes and microcomputers. It is especially suitable for multiprocessor, graphics, and vector-processing systems. Many of the commands in the z/OS UNIX System Services shell are based on similar commands available with UNIX System V.

UNIX file. An object that exists within a hierarchical file system (HFS). UNIX files include files in a DFSMS HFS, a Network File System (NFS), a temporary file system (TFS), and so on.

UNIX filter. A filter that accepts input via **stdin**, and returns the output via **stdout**. Options and positional arguments can be specified. UNIX filters may be any shell executable, for example, a shell script, a REXX exec, a C program, and so on. See also *filter*. Contrast with *DLL filter*.

UNIX System Services. See *z/OS UNIX System Services*.

user port. In IP PrintWay, a port address that is outside the range of addresses defined in RFC 1179 for the LPR source port.

V

Virtual Telecommunications Access Method (VTAM). An IBM licensed program that controls communication and the flow of data in an SNA network. It provides single-domain, multiple-domain, and interconnected network capability.

VTAM. Virtual Telecommunications Access Method.

W

Workbench for OS/2 and Windows. An application that runs under Windows or WIN-OS/2® that enables you to browse and print AFP documents and resources on your workstation.

Z

z/FS. The z/OS Distributed File Service zSeries File System (zFS) is a UNIX System Services file system that can be used in addition to the Hierarchical File System (HFS). zFS file systems contain files and directories that can be accessed with the z/OS hierarchical file system file Application Programming Interfaces (APIs). zFS file systems can be mounted into the UNIX System Services hierarchy along with other local (or remote) file system types (for example, HFS, TFS, AUTOMNT, NFS, etc.). zFS does not replace HFS, rather is complementary to HFS.

z/OS UNIX System Services. z/OS services that support an environment within which operating systems, servers, distributed systems, and workstations share common interfaces. z/OS UNIX System Services supports standard application development across multivendor systems. It is required if you want to create and use applications that conform to the POSIX standard. z/OS UNIX System Services combines the personal power of the workstation, the flexibility of open systems, and the strength of MVS. It supports and fosters a superenvironment of larger operating systems or servers and of distributed systems and workstations that share common interfaces. Users can switch back and forth between the traditional TSO/E interface and the z/OS UNIX System Services interface. UNIX-skilled users can interact with the system, using a familiar set of standard commands and utilities. MVS-skilled users can interact with the system, using familiar TSO/E commands and interactive menus to create and manage hierarchical file system files and to copy data back and forth between MVS data sets and files. Application programmers and users have both sets of interfaces to choose from and, by making appropriate tradeoffs, can choose to mix these interfaces.

z/OS.e. z/OS.e is a specially priced offering of z/OS providing select z/OS function at a reduced price. z/OS.e is available on the IBM @server zSeries 800 (z800). It is intended to help z800 customers exploit the fast growing world of next-generation e-business by making the deployment of new Java and C/C++ applications on the z800 attractively priced. z/OS.e has the same code base as the z/OS operating system customized with new system parameters.

Bibliography

This section lists publications that may be helpful to you as you configure and use Infoprint Server.

Infoprint Server

Title	Order Number
Infoprint Server Transforms Licensed Program Specifications	G544-5797
z/OS Infoprint Server Customization	S544-5744
z/OS Infoprint Server Introduction	S544-5742
z/OS Infoprint Server Messages and Diagnosis	G544-5747
z/OS Infoprint Server Migration	G544-5743
z/OS Infoprint Server Operation and Administration	S544-5745
z/OS Infoprint Server User's Guide	S544-5746
Infoprint Server for z/OS Implementation Redbook	SG24–6234

Print Services Facility for OS/390

Title	Order Number
AFP Conversion and Indexing Facility: User's Guide	S544-5285
PSF for OS/390 & z/OS: Customization	S544-5622
PSF for OS/390 & z/OS: Diagnosis	G544-5623
PSF for OS/390 & z/OS: Download for OS/390	S544-5624
PSF for OS/390 & z/OS: Introduction	G544-5625
PSF for OS/390 & z/OS: Messages and Codes	G544-5627
PSF for OS/390 & z/OS: User's Guide	S544-5630

Advanced Function Presentation (AFP)

Title	Order Number
IBM Printing Systems: Printer Information	S544-5750
IBM Printing Systems: Printer Summary	S544-5749
AFP: Programming Guide and Line Data Reference	S544-3884
IBM AFP Fonts: Font Summary for AFP Font Collection	S544-5633
IBM AFP Fonts: Font Summary for AFP Font Collection	S544-5633
IBM Data Stream and Object Arcitectures: Bar Code Object Content Architecture Reference	S544–3766
IBM Data Stream and Object Arcitectures: IOCA Reference	SC31–6805
IBM Page Printer Formatting Aid: User's Guide	S544-5284

© Copyright IBM Corp. 1995, 2002

Infoprint Manager for AIX and Windows

Title	Order Number
IBM Infoprint Color 130 Plus Installation Planning Guide	G544-5771
IBM Infoprint Manager: Reference	S544-5475
IBM Infoprint Manager for AIX: Administrator's Guide	S544-5595

z/OS Version 1 Release 2

Title	Order Number
z/OS C/C++ Programming Guide	SC09-4765
z/OS C/C++ Run-Time Library Reference	SA22-7821
z/OS Communications Server: IP and SNA Codes	SC31-8791
z/OS Communications Server: IP Application Programming Interface Guide	SC31-8788
z/OS Communications Server: IP Configuration Guide	SC31-8775
z/OS Communications Server: IP Configuration Reference	SC31-8776
z/OS Communications Server: IP Migration	GC31-8773
z/OS Communications Server: SNA Diagnosis Vol 1 Techniques and Procedures	LY43-0088
z/OS Communications Server: SNA Diagnosis Vol 2 FFST Dumps and the VIT	LY43-0089
z/OS Communications Server: SNA Messages	SC31-8790
z/OS Communications Server: SNA Network Implementation Guide	SC31-8777
z/OS Communications Server: SNA Operation	SC31-8779
z/OS Communications Server: SNA Programming	SC31-8829
z/OS Communications Server: SNA Resource Definition Reference	SC31-8778
z/OS Distributed File Service SMB Administration	SC24-5918
z/OS Information Roadmap	SA22-7500
z/OS ISPF Dialog Developer's Guide and Reference	SC34-4821
z/OS JES2 Commands	SA22-7526
z/OS JES2 Initialization and Tuning Guide	SA22-7532
z/OS JES2 Initialization and Tuning Reference	SA22-7533
z/OS JES3 Commands	SA22-7540
z/OS JES3 Initialization and Tuning Guide	SA22-7549
z/OS JES3 Initialization and Tuning Reference	SA22-7550
z/OS Language Environment Debugging Guide	GA22-7560
z/OS Language Environment Run-Time Messages	SA22-7566
z/OS Language Environment Programming Guide	SA22-7561
z/OS MVS Diagnosis: Tools and Service Aids	GA22-7589
z/OS MVS Initialization and Tuning Guide	SA22-7591
z/OS MVS Initialization and Tuning Reference	SA22-7592
z/OS MVS JCL Reference	SA22-7597
z/OS MVS Product Management	SA22-7603

Title	Order Number
z/OS MVS Programming: Authorized Assembler Services Reference ALE-DYN	SA22-7609
z/OS MVS Programming: Authorized Assembler Services Reference ENF-IXG	SA22-7610
z/OS MVS Programming: Authorized Assembler Services Reference LLA-SDU	SA22-7611
z/OS MVS Programming: Authorized Assembler Services Reference SET-WTO	SA22-7612
z/OS and z/OS.e Planning for Installation	GA22-7504
z/OS Program Directory	GI10-0670
z/OS SDSF Operation and Customization	SA22-7670
z/OS Security Server RACF General User's Guide	SA22-7685
z/OS Security Server RACF Security Administrator's Guide	SA22-7683
z/OS Summary of Message Changes	SA22-7505
z/OS UNIX System Services Command Reference	SA22-7802
z/OS UNIX System Services Messages and Codes	SA22-7807
z/OS UNIX System Services Planning	GA22-7800
z/OS UNIX System Services User's Guide	SA22-7801

CICS

Title	Order Number
CICS Customization Guide	SC34-5706
CICS Diagnosis Reference	LY33-6097
CICS Resource Definition Guide	SC34-5722
CICS Supplied Transactions	SC34-5724

IMS/ESA® Version 6

Title	Order Number
IMS/ESA Administration Guide: System	SC26-8730
IMS/ESA Administration Guide: Transaction Manager	SC26-8731

3270 and SNA Data Streams

Title	Order Number
IBM 3270 Information Display System Data Stream Programmer's Reference	GA23–0059
IBM 3270 Information Display System 3274 Control Unit Description and Programmer's Reference	GA23–0061
IBM 3270 Information Display System Reference Summary	GX20–1878
IBM 3270 Kanji Data Streams	GA18–2980

Title	Order Number
IPDS and SCS Technical Reference	S544–5312
Systems Network Architecture: Sessions Between Logical Units	GC20–1868

Index

Special Characters

_BPX_JOBNAME environment variable 90, 250
_BPXK_SETIBMOPT_TRANSPORT environment variable 63, 268
/etc/Printsrv directory 26
/etc/Printsrv/aopd.conf, creating 28
/etc/Printsrv/aopsapd.conf, creating 73
/etc/profile, setting environment variables in 31
/etc/rc file
  setstack command 42
  setting STEPLIB variable 28
  starting Infoprint Server daemons 42
  starting sendmail 167
/usr/lpp/Printsrv/bin, specifying in PATH variable 32, 270
/var/Printsrv directory 26
% operator in filter options 55
& in configuration files 29, 75, 88, 248, 262, 265
# in configuration files 29, 75, 88, 248, 262, 265

Numerics

00000000.tab file 114
01200000.tab file 114
11000000.tab file 114
161 and 162 ports, unrestricting in TCP/IP data set 52
3270 data stream support in NetSpool 15
3270 data streams, NetSpool exits for 150
40010000.tab file 114
40020000.tab file 114
515 port number
  specified as default 64
  unrestricting in TCP/IP data set 52
631 port number
  specified as default 68
  unrestricting in TCP/IP data set 52
80000000.tab file 114

A

ABAP data, converting 106
abend D37 144, 180
AbortPrintFile function 279
accessibility 295
address spaces, running NetSpool in multiple 145
administrator group, for security 33
AFP Printer Driver for Windows, overview 4
AFP to PCL transform
  customizing 83, 115
  transform configuration file entry 116
AFP to PDF transform
  customizing 83, 115
  transform configuration file entry 116
AFP to PostScript transform
  customizing 83, 115
  transform configuration file entry 116
AFP Viewer plug-in for Windows, overview 5
AIX, increasing time-out value 66
AIX, transforms on 85
aliases file, sendmail 167
ANFDEAL sample JCL 179
ANFMFILE program 144, 180
ANFMIJCL sample JCL 180
ANFMMSG DD statement 184
ANFQUEUE DD statement 184
ANFUBTBL, tables for block letters 234
ANFUEXTP control block 197, 205, 211, 217, 223, 227
ANFULINK sample JCL 236
ANFUXRSP exit 228
ANFUXxxx exits, sample 235
ANFWPROC startup procedure 182
annotations, suppressing in the AFP to PDF transform 124, 255
AOP_64K_JOBID 268
AOP_64K_JOBID environment variable 293
AOP_ANNOTATIONS environment variable 124, 255
AOP_CHARS environment variable 124, 255
AOP_COLOR environment variable 125, 256
AOP_CUTSHEET environment variable 86, 125, 256
AOP_FLATE environment variable 86, 126, 257
AOP_FONTLIB environment variable 126, 257
AOP_FONTMAP environment variable 123, 127, 258
AOP_FORMDEF environment variable 128, 258
AOP_FORMDEFLIB environment variable 128, 258
AOP_HORIZONTAL_MARGINS environment variable 94
AOP_MSGFORMDEF environment variable 128, 259
AOP_MSGPAGEDEF environment variable 128, 259
AOP_OUTLINES environment variable 123, 128, 259
AOP_OVERLAYLIB environment variable 129, 260
AOP_PAGE_HEIGHT environment variable 94
AOP_PAGE_WIDTH environment variable 94
AOP_PAGEDEF environment variable 129, 260
AOP_PAGEDEFLIB environment variable 129, 260
AOP_PAGESEGLIB environment variable 129, 260
AOP_PAPER environment variable 86, 117, 130, 260
AOP_PJL environment variable 86, 130, 261
AOP_RESOLUTION environment variable 94
AOP_RESOURCE_PATH environment variable 100
AOP_SAP2AFP_RESOURCES environment variable 103, 268
AOP_TRAYID environment variable 116, 131, 261
AOP_VERTICAL_MARGINS environment variable 94
AOPADMIN group 33
AOPADMIN resource 33
AOPAPI 275
AOPBATCH utility program 271
AOPCONF environment variable 32, 268
aopd.conf
  ascii-codepage attribute 29
  base-directory attribute 30
  creating 28
  ebcdic-codepage attribute 30
  format 245, 249, 264
  inventory attribute 30

© Copyright IBM Corp. 1995, 2002

aopd.conf *(continued)*
ipp-port-number attribute 68
job-prefix attribute 54
lpd-port-number attribute 64
naming in AOPCONF envorment variable 32
snmp-community attribute 141
start-command attribute 93
start-daemons attribute 31, 64, 69, 87, 141
aopd.conf See also configuration file, aopd.conf 245
aopfiltr.so 59
AOPINIT EXEC 43
AOPJAUTH job 69
AOPMAILER environment variable 185
AOPOPER group 33
AOPOPTIONS environment variable 60
AOPPATH environment variable 268
AOPPATH variable 60
AOPPRINT JCL procedure, customizing 60
aoprform.dll 85
AOPSAPD_CONF environment variable 75, 269
aopsapd.conf 264
client attribute 74, 264
creating 73
example 74
format 74
initial-target attribute 74, 264
language attribute 74, 264
naming in environment variable 75
password attribute 74, 264
syntax rules 75
system attribute 74, 264
userid attribute 74, 264
aopsetup shell script 35
aopstart EXEC, editing 36
AOPSTART procedure 39
AOPSTOP procedure 40
AOPTRACEON environment variable, setting in AOPSTART procedure 39
AOPXFD_CONF environment variable 91, 269
aopxfd.conf 249
AFP to PCL, PDF, and PostScript transform entry 116
creating 87
environment attribute 90, 94, 100, 249
example 91, 94, 100, 131
format 89, 98
maximum-active attribute 91, 94, 100, 250
maximum-idle-time attribute 90, 94, 100, 250
minimum-active attribute 90, 94, 100, 250
PCL to AFP transform entry 92
PDF to AFP transform entry 98
PostScript to AFP transform entry 98
start-command attribute 90, 99, 249
transform attribute 89, 93, 98, 249
AOXCF30 font-conversion program 135
AOXCF30J member of SYS1.SAMPLIB 137
APF authorizing Java libraries 69
APF-authorizing C++ applications that use API 292
APF-authorizing NetSpool exits 151
APF-authorizing SYS1.IMAGELIB 178
API 275
APIJPJCL startup procedure for NetSpool 147
APIJPTD1 161
APIJPTD2 161
APIMIJCL sample JCL 144
APIMMSG DD statement 149
APIPP1X2 control block 151
APIPP1X2 macro 161
APIPPTD1 exit, NetSpool 153
APIPPTD2 exit, NetSpool 157
APIUBF3 exit, NetSpool 155
APIUGEX exit, NetSpool 159
APPL id, specifying for IP PrintWay 186
APPL statement for IP PrintWay 185
Application Programming Interface 275
ascii-codepage attribute in configuration file 29
attributes, job, specifying in environment variable 60
AUTH parameter 186
AUTOSTOP parameter (JES2) 172

B

backend option, AIX 66
backing up the Printer Inventory 44
banner page, adding in exit 204, 216
banner page, controlling in sample exit 235
bar codes, mapping to BCOCA for SAP to AFP transform 104
barcode.tab configuration file 104
base-directory attribute in configuration file 30
batch job, using AOPBATCH 271
Begin Data Set exit, IP PrintWay
examples 210
input 205
installing 236
modifying 236
output 209
overview 204
programming considerations 191
sample 235
BeginEnumJobs function 280
BeginEnumPrinters function 280
Beginning of File exit for 3270 data, NetSpool (APIPUBF3)
description 155
examples 157
input 156
installing 161
modifying 162
output 157
Beginning of File exit for SCS data, NetSpool (APIPPTD1)
description 153
examples 155
input 154
installing 161
modifying 162
output 155
programming considerations 151
sample 161
binary data support in NetSpool 15
blank truncation 53, 143, 171, 174

BLNKTRNC parameter 53, 143
BLNKTRNC parameter (OUTCLASS) 171
Block Letter routine
  description of 234
  input 234
  link-editing with exit 236
  output 235
  sample exit for 235
  tables (ANFUBTBL) 234
BPXPRMxx member, updating
  for Inventory Manager 27
  for IP PrintWay 178
  for IPP server 68
  for Print Interface 54
  for Transform Manager 87, 96
broadcasting data 15
BufferSizeRequired function 281

C

C++ run-time options, changing in IP PrintWay 185
C++ run-time options, changing in NetSpool 150
callable services 275
CancelJob function 281
CARRIAGE parameter 170
cfilter.c 57
checklist of customization tasks 241
CICS, customization required to print from 23
CKJ support in transforms 110
class
  changing in exit 196
  CLASS parameter (JES2) 173
  QUEUE parameter (JES2) 173
  specifying as JES2 work-selection criteria 173, 174
  specifying as JES3 work-selection criteria 177
  WC parameter (JES3) 177
classes for transforms 88, 92, 93, 98, 116
CLASSPATH environment variable 69, 269
client attribute 74, 264
ClosePrintFile function 282
Coax Printer Support feature 16, 165
code page default for IP PrintWay, specifying in FSS definition 187
code page, default for IP PrintWay 186
code pages, default, specifying in configuration file 29, 30
coexistence, regarding Printer Inventory 25
color transform requirements 50
color, specifying for AFP to PCL, PostScript, and PDF transforms 125, 256
community name for SNMP in configuration file 141
compatibility filter 64
compression, specifying for AFP to PDF transform 126, 257
configuration file aopd.conf
  ascii-codepage attribute 29
  base-directory attribute 30
  creating 28
  ebcdic-codepage attribute 30
  format 245, 249, 264
  inventory attribute 30
configuration file aopd.conf *(continued)*
  ipp-port-number attribute 68
  job-prefix attribute 54
  lpd-port-number attribute 64
  naming in AOPCONF envorment variable 32
  snmp-community attribute 141
  start-command attribute 93
  start-daemons attribute 31, 64, 69, 87, 141
configuration file aopsapd.conf
  client attribute 74, 264
  creating 73
  example 74
  format 74
  initial-target attribute 74, 264
  language attribute 74, 264
  naming in environment variable 75
  password attribute 74, 264
  syntax rules 75
  system attribute 74, 264
  userid attribute 74, 264
configuration file aopxfd.conf
  AFP to PCL, PDF, and PostScript transform entry 116
  creating 87
  environment attribute 90, 94, 100, 249
  example 91, 94, 100, 131
  format 89, 98
  maximum-active attribute 91, 94, 100, 250
  maximum-idle-time attribute 90, 94, 100, 250
  minimum-active attribute 90, 94, 100, 250
  PCL to AFP transform entry 92
  PDF to AFP transform entry 98
  PostScript to AFP transform entry 98
  start-command attribute 90, 99, 249
  transform attribute 89, 93, 98, 249
configuration file, barcode.tab 104
configuration file, defcp.tab 106
configuration file, fonts.tab 109
configuration file, image.tab 111
configuration file, pagedef.tab 112
configuration file, xxxx0000.tab 114
configuration files for SAP to AFP transform 104
CONSNAME parameter, for NetSpool 149
consoles, for NetSpool 146, 149
control file first, sending with LPR 65
COPIES parameter, changing in exit 196
cp command, example of 29, 73
CreatePrintFile function 282
customer fonts 110
customizing
  AOPPRINT JCL procedure 60
  Infoprint Port Monitor for Windows 237
  Infoprint Server Transforms 83
  IP PrintWay 163
  LPD 61
  NetSpool 143
  Print Interface 49
  Print Interface IPP server 67
  Print Interface SAP R/3 OMS 71
  Printer Inventory Manager 25
  SNMP manager 142

customizing *(continued)*
SNMP subagent 139
Transform Manager 83
Windows client 237
z/OS SNMP agent 140
cutsheet printing, specifying for AFP to PCL, PostScript, and PDF transforms 125, 256

D

D37 abend 144, 180
daemons to start 31, 64, 69, 80, 87, 141
daemons, description of 35
DATASETPREFIX parameter, in TCPIP.DATA data set 166
DBCS, customizing for
code pages for SAP to AFP transform 108
conversion tables for SAP to AFP transform 103
default ASCII code page, specifying 29
default EBCDIC code page, specifying 30
mappling SAP code pages to AFP code pages 114
default printer, specifying 60, 270
defcp.tab configuration file 106
destination
changing in exit 196
DGROUP parameter (JES3) 176
ROUTECDE parameter (JES2) 173
specifying as JES2 work-selection criteria 174
specifying as JES3 work-selection criteria 177
DEVICE statement (JES3) 176
DGROUP parameter (JES3) 176
diagram of system flow 6
direct sockets protocol, IP PrintWay
customization tasks specific to protocol 164
directories for Infoprint Server 26
directory name for files, specifying in configuration file 30
disability 295
DLLs, specifying directory for 32, 75, 269
DocumentInfo data structure 276
Domain Name Server, customizing 167
downloading Infoprint Port Monitor 237
DTYPE parameter (JES3) 176
dynamic enablement of Infoprint Server 26
dynamic enablement of Infoprint Server Transforms 86
DYNAMIC parameter (JES3) 176

E

e-mail protocol, IP PrintWay
changing e-mail address in Routing exit 203
customization tasks specific to protocol 164
environment variables 184
sendmail customization 167
software required 165
ebcdic-codepage attribute in configuration file 30
echo command, example of 31
edge-to-edge printing, paper names for 118, 130, 261
enabling Infoprint Server 26
enabling Infoprint Server Transforms 86
End Data Set exit, IP PrintWay
examples 221
input 217
installing 236
modifying 236
output 221
overview 216
programming considerations 191
sample 235
EndEnumJobs function 283
EndEnumPrinters function 284
EnumJobs function 284
EnumPrinters function 285
environment attribute 90, 94, 100, 249
environment variables
_BPX_JOBNAME 90, 250
_BPXK_SETIBMOPT_TRANSPORT variable 63, 268
AOP_64K_JOBID variable 268, 293
AOP_ANNOTATIONS variable 124, 255
AOP_CHARS variable 124, 255
AOP_COLOR variable 125, 256
AOP_CUTSHEET variable 86, 125, 256
AOP_FLATE variable 86, 126, 257
AOP_FONTLIB variable 126, 257
AOP_FONTMAP variable 123, 127, 258
AOP_FORMDEF variable 128, 258
AOP_FORMDEFLIB variable 128, 258
AOP_HORIZONTAL_MARGINS variable 94
AOP_MSGFORMDEF variable 128, 259
AOP_MSGPAGEDEF variable 128, 259
AOP_OUTLINES variable 123, 128, 259
AOP_OVERLAYLIB variable 129, 260
AOP_PAGE_HEIGHT variable 94
AOP_PAGE_WIDTH variable 94
AOP_PAGEDEF variable 129, 260
AOP_PAGEDEFLIB variable 129, 260
AOP_PAGESEGLIB variable 129, 260
AOP_PAPER variable 86, 117, 130, 260
AOP_PJL variable 86, 130, 261
AOP_RESOLUTION variable 94
AOP_RESOURCE_PATH variable 100
AOP_SAP2AFP_RESOURCES variable 103, 268
AOP_TRAYID variable 116, 131, 261
AOP_VERTICAL_MARGINS variable 94
AOPCONF variable 32, 268
AOPMAILER variable 185, 268
AOPOPTIONS variable 60, 268
AOPPATH variable 60, 268
AOPSAPD_CONF variable 75, 269
AOPXFD_CONF variable 91, 269
CLASSPATH variable 69, 269
example of displaying with echo command 31
examples of setting with export command 31
for AFP to PCL, PDF, and PostScript transforms 124, 255
for IP PrintWay 184
for IPP server 69
for PCL to AFP transform 94
for PostScript/PDF to AFP transform 100
for Print Interface IPP server 69

environment variables *(continued)*
for Printer Inventory Manager 31
for SAP OMS 75
for SAP to AFP transform 103
for Transform Manager 91
for transforms 90, 249
JAVA_HOME variable 69, 269
LANG 184
LIBPATH 184
LIBPATH variable 32, 75, 269
LPDEST variable 60, 270
MANPATH variable 60, 91, 269
NLSPATH 184
NLSPATH variable 32, 270
PATH 185
PATH variable 32, 270
PRINTER variable 60, 270
setting in aopstart EXEC 36
STEPLIB variable 28, 270
summary of 267
error-handling for font conversion 137
ErrorNumber function 285
ErrorString function 286
examples
Begin Data Set exit 210
End Data Set exit 221
JES2 initialization 172
JES3 initialization 175
Message exit 228
Record exit 215
Routing exit 203
SAP Callback daemon configuration file 74
SMF exit 226
transform configuration file 91, 94, 100, 131
exit programs 150, 189
APF-authorizing NetSpool exits 151
APIPP1X2 control block 151
installing IP PrintWay exits 236
installing NetSpool exits 161
IP PrintWay Begin Data Set exit 204
IP PrintWay End Data Set exit 216
IP PrintWay Message exit 227
IP PrintWay Record exit 210
IP PrintWay Response Notification exit 228
IP PrintWay Routing exit 196
IP PrintWay SMF exit 222
modifying IP PrintWay exits 236
modifying NetSpool exits 162
NetSpool Beginning of File exit for 3270 data streams 155
NetSpool Beginning of File exit for SCS data 153
NetSpool Graphic Escape exit 159
NetSpool Transparent Data Control exit 157
order of invocation, IP PrintWay 190
Print Interface filters 55
programming considerations, IP PrintWay 191
programming considerations, NetSpool 151
sample IP PrintWay exits 235
sample NetSpool exits 161
STEPLIB for NetSpool exits 149
export command, example of 31
exporting the Printer Inventory 44
extattr command, using to authorize libraries 70

F

FACILITY class 35
FCB parameter (JES2) 173
FCB work-selection criterion 177
FCB, APF authorization for SYS1.IMAGELIB 178
FCB, default for IP PrintWay (JES3) 176
FILESYSTYPE parameter 87
filter programs, Print Interface
examples of % operator 56
input 57, 58
LPD compatibility 64
modifying 59
options for 55
output 57, 58
programming considerations 58
REXX examples 56
sample 59
source code, sample 57
testing 59
transform filters (IBM) 59
writing a DLL filter 57
writing a UNIX filter 58
firewalls, e-mail considerations 168
Flate compression, specifying 126, 257
font default, specifying for AFP to PCL, PostScript, and PDF transforms 124, 255
font libraries, specifying for AFP to PCL, PostScript, and PDF transforms 126, 257
font mapping, specifying for AFP to PCL, PostScript, and PDF transforms 123, 127, 258
font substitution for PostScript and PDF to AFP transform 101
font-conversion program (AOXCF30) 135
description 135
error handling 137
font, default for IP PrintWay (JES2) 170, 174
font, default for IP PrintWay (JES3) 170, 176
fonts, outline, specifying type of transform processing 123, 128, 259
fonts, pre-loading for PostScript and PDF to AFP transform 102
fonts.tab file 109
form definition for transform messages, specifying 128, 259
form definition libraries, specifying for AFP to PCL, PostScript, and PDF transforms 128, 258
form definition, specifying for AFP to PCL, PostScript, and PDF transforms 128, 258
form definitions, mapping to SAP Format names 112
form name
changing in exit 196
FORMS parameter (JES2) 173
specifying as JES2 work-selection criteria 174
specifying as JES3 work-selection criteria 177
Format names, mapping to page and form definitions 112
FSA definitions for IP PrintWay, creating 188

FSS definitions for IP PrintWay, creating 186
FSS initialization statement (JES2) 172
FSS initialization statement (JES3) 175
FSSNAME parameter (JES3) 175, 176
functional subsystem (FSS), IP PrintWay
  defining multiple FSAs 169
  defining multiple FSSs 169
  JES2 initialization statements 172
  JES3 initialization 175
  name, specifying in JES2 172
  name, specifying in JES3 175
  planning 169
functional subsystem application (FSA), IP PrintWay
  determining work-selection criteria 169
  how many IP PrintWay FSAs to create 169
  IP PrintWay configuration information in 188
  name, specifying in JES2 172
  name, specifying in JES3 176

G

GetJobInfo function 287
GetPrinterInfo function 288
GF10 default font (JES2) 170, 174
Graphic Exit, NetSpool (APIUGEX)
  description 159
  examples 161
  input 160
  installing 161
  modifying 162
  output 161
  sample 161
GS10 default font (JES3) 170

H

HFS directories 26
hiperspace blocks for IP PrintWay 187
HRI character 105
human readable interface (HRI) character 105

I

IAZJSPA control block 192
IBM-1047, specified as default 30
IDCAMS parameters to allocate transmission-queue data sets 179
IFAPRDxx member of SYS1.PARMLIB 26, 86
image data in SAP to AFP transform 111
image.tab file 111
IMS, customization required to print from 23
Infoprint Color 130 Plus printer, requirements 50
Infoprint Port Monitor for Windows, overview 4
Infoprint Port Monitor, customizing 237
Infoprint Server configuration file aopd.conf
  ascii-codepage attribute 29
  base-directory attribute 30
  creating 28
  ebcdic-codepage attribute 30
  format 245, 249, 264
  inventory attribute 30
Infoprint Server configuration file aopd.conf *(continued)*
  ipp-port-number attribute 68
  job-prefix attribute 54
  lpd-port-number attribute 64
  naming in AOPCONF envorment variable 32
  snmp-community attribute 141
  start-command attribute 93
  start-daemons attribute 31, 64, 69, 87, 141
Infoprint Server Transforms
  components to customize 23
  customizing 83
  default font, specifying to JES2 174
  default font, suppressing in JES2 170
  default font, suppressing in JES3 170, 176
  default page definition, specifying to JES2 173
  default page definition, suppressing in JES2 170
  default page definition, suppressing in JES3 170, 176
  dynamic enablement 86
  remotely on AIX or Windows using Infoprint Manager 85
  temporary files in /var/Printsrv 27
  transform DLLs 59
Infoprint Server Transforms, overview 12
InitAPI function 288
initial-target attribute 74, 264
input data streams, types supported 14
installing
  NetSpool exits 161
installing IP PrintWay exits 236
interfaces, programming 55
INTERVAL parameter 52
INV parameter in IP PrintWay startup procedure 184
INV parameter in NetSpool startup procedure 149
inventory attribute in configuration file 30, 80
Inventory, Printer
  backing up 44
  naming in configuration file 30
  naming in IP PrintWay startup procedure 184
  naming in NetSpool startup procedure 149
  restoring 44
  sharing 25
IP address, changing in exit 196
IP PrintWay
  APF-authorizing SYS1.IMAGELIB 178
  APPL statement 185
  customizing 163
  exits, writing 189
  FSA definitions for 188
  FSS and FSA definitions 186
  FSS, defining 169
  increasing allowed number of processes 178
  LE run-time options 185
  sharing AFP printers with PSF for OS/390 236
  software requirements 165
  startup procedure for 181
  TCPIP.DATA data set, customizing for 166, 167
  transmission-queue data set, allocating 178
IP PrintWay, overview 16
IPP protocol, IP PrintWay
  customization tasks specific to protocol 164

IPP protocol, IP PrintWay *(continued)*
environment variables 184
IPP server in Print Interface, customizing 67
APF authorizing libraries 69
environment variables for 69
IPP workstation client, customizing 70
specifying port number for 68
starting daemon for 69
URI used by IPP workstation client 70
ipp-port-number attribute in configuration file 68
ISO8859-1, specified as default 29
ISPF panels, Infoprint Server
defining libraries in TSO logon procedure 42
displaying Infoprint Server option 43
enabling 42
modifying AOPINIT EXEC 43

J

Japanese, ISPF panels 42
Japanese, messages 40, 43, 145, 187
Java run-time libraries, APF authorizing 69
Java software requirements 50
JAVA_HOME environment variable 69, 269
JES work-selection criteria
determining 169
specifying in JES2 174
specifying in JES3 177
JES2
FSS statement example 172
initialization statements 171
PRTnnnnn statement 172
JES3
DEVICE statement 176
FSS statement 175
initialization statements 174
JESNEWS data set
indicator in ANFUEXTP control block 194
suppressing SMF type-6 record for 223, 235
JNAME parameter (JES3) 176
job attributes, specifying in environment variable 60
job control language (JCL)
sample to link-edit exit 236
startup, for Infoprint Server daemons 39
startup, IP PrintWay 183
startup, NetSpool 144
job name for transforms, changing 90, 250
job separator page data area (IAZJSPA) 192
job-prefix attribute in configuration file 54
JobInfo data structure 276
JobToken data structure 277
JSPA control block 192

K

Kanji AFP Print feature 12, 83
keyboard 295

L

LANG environment variable 184
language attribute 74, 264
language for IP PrintWay messages 186
LANGUAGE parameter, in NetSpool startup procedure 149
large files, sending to LPD 65
LE libraries, making available to Infoprint Server 28
LE run-time libraries, making available 27
LE run-time options, changing in IP PrintWay 185
LE run-time options, changing in NetSpool 150
LIBPATH environment variable 32, 75, 184, 269
libraries, defining in TSO logon procedure (Print Interface) 42
logon procedure for ISPF panels 42
LOMS for SAP R/3
defining 75
modifying 76
selecting for an output device 77
LPD compatibility filter 64
LPD control file first, sending with LPR 65
lpd_compat.so 64
lpd-port-number attribute in configuration file 64
LPD, customizing 61
command codes supported in 65
running TCP/IP LPD on same system 63
specifying port number for 64
starting daemon 64
LPDEST environment variable 60, 270
LPR protocol, IP PrintWay
customization tasks specific to protocol 164
LUCLASS parameter in NetSpool startup procedure 148

M

man pages, specifying directory for 60, 91, 269
MANPATH environment variable 269
MANPATH variable 60, 91
margins, specifying for PCL to AFP transform 94
MAXASSIZE parameter 68, 96
maximum-active attribute 91, 94, 100, 250
maximum-idle-time attribute 90, 94, 100, 250
MAXPROCSYS parameter 54, 178
MAXPROCUSER parameter 54, 178
MCS console 146
Message exit, IP PrintWay
examples 228
input 227
installing 236
modifying 236
output 228
overview 227
programming considerations 191
sample 235
message form definition, specifying for AFP to PCL, PostScript, and PDF transforms 128, 259
message page definition, specifying for AFP to PCL, PostScript, and PDF transforms 128, 259

message-log data set, IP PrintWay
  allocating space 180
  defining in startup procedure 184
  initializing 180
message-log data set, NetSpool 144
migration program, overview 4
minimum-active attribute 90, 94, 100, 250
MODE parameter
  in DEVICE statement (JES3) 176
  in PRTnnnnn statement (JES2) 173
ModWidth keyword 105
MSGDEST parameter (JES3) 175
multiple TCP/IP stacks 63

N

NetSpool
  customizing 143
  exits for transparent data, writing 150
  JES customization 143
  LE run-time options 150
  LU classes, specified in startup procedure 148
  message log, allocating 144
  startup procedure for 144
NetSpool, overview 14
Network Printer Manager (NPM) for the Web, customizing 142
network printers, customizing PSF for OS/390 for 236
NIFCB parameter (JES2) 170, 171
NIUCS parameter (JES2) 170, 171
NLSPATH environment variable 32, 184, 270
NLVs for NetSpool 145
NOBODY user ID 97
NOGROUP group 97
NPM for the Web, customizing 142

O

operator group, for security 33
OTF data, converting fonts in 109
OUTCLASS statement (JES2) 171
outline fonts, specifying for AFP to PCL, PostScript, and PDF transforms 123, 128, 259
output devices for SAP R/3, creating 77
overlay library, specifying for AFP to PCL, PostScript, and PDF transforms 129, 260
overview of Infoprint Server components 1
  AFP Printer Driver for Windows 4
  AFP Viewer plug-in for Windows 5
  Infoprint Port Monitor for Windows 4
  Infoprint Server Transforms 12
  IP PrintWay 16
  NetSpool 14
  Print Interface 5
  Printer Inventory 3
  Printer Inventory Manager 3
  Transform Manager 13
  Windows client 4

P

page definition default for IP PrintWay (JES2) 173
page definition default, specifying for AFP to PCL, PDF, and PostScript transforms 129, 260
page definition default, specifying for AFP to PCL, PostScript, and PDF transforms 129, 260
page definition for transform messages, specifying 128, 259
page definition, default for IP PrintWay (JES2) 170
page definition, default for IP PrintWay (JES3) 170, 176
page definitions, mapping to SAP Format names 112
page height, specifying for PCL to AFP transform 94
page segment library, specifying for AFP to PCL, PostScript, and PDF transforms 129, 260
page width, specifying for PCL to AFP transform 94
pagedef.tab file 112
panels, ISPF
  defining libraries in TSO logon procedure 42
  displaying Infoprint Server option 43
  enabling 42
  modifying AOPINIT EXEC 43
paper dimensions used by transforms 117
paper sizes, specifying for AFP to PCL, PostScript, and PDF transforms 117, 130, 260
parameters
  JES2 initialization statements 171
  JES3 initialization statements 174
  LANGUAGE parameter 149
  TRACE parameter 149
password attribute 74, 264
PATH environment variable 32, 185, 270
PCL to AFP transform
  customizing 83, 92
  environment variables 94
  transform configuration file entry 92
PDEFAULT parameter (JES3) 170, 176
PDF to AFP transform
  customizing 83, 95
  environment variable 100
  font substition, enabling 101
  fonts, preloading 102
  preload.ps resource file 102
  security for 97
  transform configuration file entry 98
  transform resource files 101
  UserInit resource file 101
performance and the TRKCELL parameter (JES2) 173
PIDU, used to backup and restore Printer Inventory 44
PJL, specifying for AFP to PCL, PostScript, and PDF transforms 130, 261
PNAME parameter (JES3) 175
port number
  for Print Interface IPP server 68
  for Print Interface LPD 64
  used by LPR 65
PORT statement in TCP/IP 52
ports, unrestricting in TCP/IP 52
PostScript to AFP transform
  customizing 83, 95
  environment variable 100

PostScript to AFP transform *(continued)*
font substition, enabling 101
fonts, preloading 102
preload.ps resource file 102
security for 97
transform configuration file entry 98
transform resource files 101
UserInit resource file 101
prefix for job identifier 54
preload.ps resource file 102
PRESELCT parameter (JES2) 173
Print Interface
AOPPRINT JCL procedure, customizing 60
common customization required for all functions 51
configuration file attributes 54
customizing 49
environment variables for z/OS UNIX commands 60
filters, writing 55
increasing allowed number of processes 54
IPP server customization 67
JES customization 53
LPD customization 61
PTFs required 50
SAP customization 71
software requirements 50
subsystem customization 79
TCP/IP customization 51
Print Interface, overview 5
print queue, changing in exit 196
PRINTDEF statement (JES2) 170, 171
PRINTER environment variable 60, 270
Printer Inventory
backing up 44
naming in configuration file 30
naming in IP PrintWay startup procedure 184
naming in NetSpool startup procedure 149
restoring 44
sharing 25
Printer Inventory FSS and FSA definitions 186
Printer Inventory Manager
backing up and resotoring Printer Inventory 44
configuration file 28
creating startup and shutdown procedures for 35
customizing 25
dynamic enablement of Infoprint Server 26
enabling ISPF panels for 42
establishing security for 33
HFS directories for 26
making run-time libraries available for 27
setting environment variables for 31
verifying software requirements 25
Printer Inventory Manager, overview 3
Printer Inventory, overview 3
Printer Port Monitor compared to other Windows printing protocols 237
printer setup string
adding in a filter program 55
printer setup string, adding in exit 204
printer sharing, customizing PSF for OS/390 for 236
printer, setting a default 60, 270
PrinterInfo data structure 277
PrintWay
APF-authorizing SYS1.IMAGELIB 178
APPL statement 185
customizing 163
exits, writing 189
FSA definitions for 188
FSS and FSA definitions 186
FSS, defining 169
increasing allowed number of processes 178
LE run-time options 185
sharing AFP printers with PSF for OS/390 236
software requirements 165
startup procedure for 181
TCPIP.DATA data set, customizing for 166, 167
transmission-queue data set, allocating 178
PrintWay, overview 16
PROC parameter (JES2) 172
procedure, startup
for Infoprint Server daemons 39
IP PrintWay 181
sample 183
specifying name of in JES2 172
specifying name of in JES3 175
NetSpool 144
sample 147
profile, TCP/IP 51
PROFILE.TCPIP data set, modifying for SNMP subagent 140
programming considerations for filter programs 58
programming considerations for IP PrintWay exits 191
programming considerations for NetSpool exit programs 151
programming interfaces 55
PRTnnnnn initialization statement (JES2) 172
PRTOPTNS parameter, changing in exit 196
PRTQUEUE parameter, changing in exit 196
PS, default job prefix 54
PSF for OS/390, components to customize 23
PSF for OS/390, customizing for printer sharing 236
PSF for OS/390, enabling SNMP reporting in Printer Inventory 141
PSF for OS/390, relationship to Infoprint Server 2, 4
PSF for OS/390, roadmap of components to customize 23
PSP bucket xxi
publications, related 311
PW.SRC file 140

Q

QueryToken data structure 278
QUEUE parameter (JES2) 173

R

RACF
aopsetup shell script 35
security for IP PrintWay transmission-queue data set 180
security for PostScript and PDF to AFP transform 97

RACF *(continued)*
security for Printer Inventory and operator commands 33
Ratio keyword 105
Record exit, IP PrintWay
examples 215
input 211
installing 236
modifying 236
output 214
overview 210
sample 235
record format
input to Begin Data Set exit 207
input to End Data Set exit 219
input to Routing exit 199
region size, IP PrintWay 184
region size, NetSpool 148
registers
for Begin Data Set exit 205
for Beginning of File exit 154, 156, 160
for Block-Letter program 234
for End Data Set exit 217
for Message exit 227
for Record exit 211
for SMF exit 223
for the Routing exit 197
for Transparent Data exit 158
ReleaseTransform function 289
rembak command, AIX 66
remote transforms 85
RequestTransform function 289
requirements, software 25, 85, 139
resolution default for SAP to AFP transform 111
resolution, specifying for PCL to AFP transform 94
Resource Access Control Facility (RACF)
aopsetup shell script 35
security for IP PrintWay transmission-queue data set 180
security for PostScript and PDF to AFP transform 97
security for Printer Inventory and operator commands 33
resource path for transforms 100, 103
Response Notification exit 228
restoring the Printer Inventory 44
resubmit for filtering
customizing IP PrintWay to use font and page definition in printer definition 170
exit processing, description 190
specifying for transforms 84
suppressing SMF record 222
RETAINF parameter, changing in exit 196
RETAINS parameter, changing in exit 196
RETRYL parameter, changing in exit 196
RETRYT parameter, changing in exit 196
return codes for font conversion 137
REXX examples, for filters 56
RFC 1179, LPD command codes supported 65
roadmap of components to customize 23
ROMS for SAP R/3, defining 75
ROUTECDE parameter (JES2) 173, 174
routing criteria, changing in exit 196
Routing exit, IP PrintWay
examples 203
input 197
installing 236
modifying 236
output 201
overview 196
programming considerations 191
sample 235
run-time libraries for LE, making available 27
run-time options for C++ LE , changing in IP PrintWay 185
run-time options for C++ LE , changing in NetSpool 150

S

S2STG DSECT 151
SACONFIG statement 140
SAF (system authorization facility)
aopsetup shell script 35
security for IP PrintWay transmission-queue data set 180
security for PostScript and PDF to AFP transform 97
security for Printer Inventory and operator commands 33
sample exits 235
SAP Callback daemon configuration file, aopsapd.conf
client attribute 74, 264
creating 73
example 74
format 74
initial-target attribute 74, 264
language attribute 74, 264
naming in environment variable 75
password attribute 74, 264
syntax rules 75
system attribute 74, 264
userid attribute 74, 264
SAP R/3 printing
configuration file 73
configuration file for Callback daemon 73
creating an SAP R/3 output device 77
customization tasks 71
environment variable for configuration file 75
ROMS and LOMS for 75
software requirements 50
user for Callback daemon 73
SAP R/3 printing, overview 9, 10
SAP to AFP transform
configuration files 104
converting ABAP data 106
customizing 83, 102
environment variable 103
mapping bar codes to BCOCA 104
mapping Format names to page and form definitions 112
mapping SAP code pages to AFP code pages 114

SAP to AFP transform *(continued)*
mapping SAP OTF fonts to AFP fonts 109
printing image data 111
resolution default, specifying 111
scaling 240–pel fonts 135
SCEERUN data set, making it available 27
scheduler work block text unit (SWBTU) 192
SCLBDLL data set, making it available 27
SCS data stream support in NetSpool 15
SCS data streams, NetSpool exits for 150
search order for TCPIP.DATA data set 166
security, establishing
aopsetup shell script 35
security for IP PrintWay transmission-queue data set 180
security for PostScript and PDF to AFP transform 97
security for Printer Inventory and operator commands 33
sendmail aliases file 167
sendmail, customizing 167
separator page, adding in exit 204, 216
Server Message Block (SMB), customizing Print Interface support for 66
Server Message Block compared to other printing protocols 237
setstack command 42
SetTerminationHandler function 290
setup string, adding in exit 204
setup strings for printers
adding in a filter program 55
shared printers, SMB 67
sharing IP PrintWay transmission-queue data set 179
sharing of printers, customizing PSF for OS/390 for 236
shortcut keys 295
shutdown procedure, for Infoprint Server daemons 40
SMB compared to other printing protocols 237
SMB software requirements 50
SMB, customizing Print Interface support for 66
SMF exit, IP PrintWay
examples 226
input 223
installing 236
modifying 236
output 225
overview 222
programming considerations 191
sample 235
SMF record
replacing in exit 222
suppressing in exit 222
smiley face printed 130, 261
SNMP agent for z/OS, customizing and starting 140
SNMP manager, customizing 142
SNMP reporting, enabling in Printer Inventory 141
SNMP subagent
configuration file statements 141
customizing 139
enabling SNMP reporting in Printer Inventory 141
NPM for the Web, customizing 142
SNMP subagent *(continued)*
SNMP manager, customizing 142
specifying community name for 141
starting daemon for 141
z/OS SNMP agent, customizing and starting 140
SNMP subagent, overview 19
snmp-community attribute in configuration file 141
SNMPD.CONF file 140
software requirements 25, 85, 139
special characters in configuration files 29, 75, 88, 248, 262, 265
SPOOLDEF statement (JES2) 170, 171
stacks, multiple TCP/IP 63
START parameter (JES2) 173
start-command attribute 90, 93, 99, 249
start-daemons attribute in configuration file 31, 64, 69, 80, 87, 141
starting daemons, specifying names in configuration file 31, 64, 69, 80, 87, 141
startup procedure
for Infoprint Server daemons 39
for IP PrintWay 181
sample 183
specifying name of in JES2 172
specifying name of in JES3 175
for NetSpool 144
sample 147
Printer Inventory Manager 39
STDENV DD statement 184
STEPLIB environment variable 28, 270
subsystem, Print Interface
customizing 79
software requirements 51
SWBTU control block 192
syntax rules
for SAP Callback configuration file 75
SYS1.PARMLIB 26, 86
SYS1.PARMLIB, updating
for Inventory Manager 27
for IP PrintWay 178
for IPP server 68
for Print Interface 54
for Transform Manager 87, 96
SYSOUT statement (JES3) 174
SYSTCPD DD statement 184
system attribute 74, 264
system authorization facility (SAF)
aopsetup shell script 35
security for IP PrintWay transmission-queue data set 180
security for PostScript and PDF to AFP transform 97
security for Printer Inventory and operator commands 33
system flow diagram 6
SYSTEM parameter (JES3) 175

T

TCP/IP
customizing sendmail 167

TCP/IP *(continued)*
error when initialization is not finished 42
modifying the hlq.PROFILE.TCPIP data set 51
modifying the TCPIP.DATA data set 166
multiple stacks, configuring 63
specifying the name of 166, 187
standard translation table, using in IP PrintWay 187
SYSTCPD DD statement in startup procedure 184
TCPCONFIG statement in TCP/IP 52
TCPIP.DATA data set 166
DATASETPREFIX parameter 166
TCPIPJOBNAME parameter 166
TCPIPJOBNAME parameter 166
TCPRCVBUFRSIZE parameter 52
TCPSENDBFRSIZE parameter 52
TERM parameter (JES3) 175
TermAPI function 290
termination handler, writing 292
time-out value, increasing with sending large files to LPD 66
TRACE parameter, in NetSpool startup procedure 149
trace, specifying for IP PrintWay 188
TRACEOPT DD statement 149
tracing parameters for IP PrintWay 187
tracing, turning on in AOPSTART procedure 39
transform attribute 89, 93, 98, 249
transform classes 88, 92, 93, 98, 116
transform configuration file, aopxfd.conf
AFP to PCL, PDF, and PostScript transform entry 116
creating 87
environment attribute 90, 94, 100, 249
example 91, 94, 100, 131
format 89, 98
maximum-active attribute 91, 94, 100, 250
maximum-idle-time attribute 90, 94, 100, 250
minimum-active attribute 90, 94, 100, 250
PCL to AFP transform entry 92
PDF to AFP transform entry 98
PostScript to AFP transform entry 98
start-command attribute 90, 99, 249
transform attribute 89, 93, 98, 249
Transform data structure 278
Transform Manager
configuration file 87, 98
customizing 83
environment variable 91
Transform Manager, overview 13
transforms
components to customize 23
customizing 83
dynamic enablement 86
remotely on AIX or Windows using Infoprint Manager 85
temporary files in /var/Printsrv 27
transform DLLs 59
translation table, standard 187
transmission-queue data set, IP PrintWay
allocating space 178
defining in startup procedure 184
initializing 178, 180
transmission-queue data set, IP PrintWay *(continued)*
restricting access 180
sharing between IP PrintWay FSSs 169, 179
Transparent Data Control exit, NetSpool
description 157
examples 159
input 158
installing 161
modifying 162
output 158
programming considerations 151
sample 161
transparent data exits in NetSpool, writing 150
transparent data support in NetSpool 15
tray numbers, specifying for AFP to PCL, PostScript, and PDF transforms 116, 131, 261
TRKCELL parameter
JES2 173
SPOOLDEF 171
TRUNC parameter 53, 143, 174
TSO logon procedure, defining libraries in 42
TYPE parameter (JES3) 175
types of LU sessions supported 14

U

UCS parameter (JES2) 174
UNRESTRICTLOWPORTS parameter 52
URI for printer 70
URL for printer 70
userid attribute 74, 264
UserInit resource file 101

V

VTAM APPL statement for IP PrintWay 185
VTAM applications, customization required to print from 23
VTAM protocol, IP PrintWay
APPL id, specifying 186
customization tasks specific to protocol 164
FSS definition, creating 186
software required 165

W

WC parameter (JES3) 177
width of barcode 105
Windows client, customizing 237
Windows client, overview 4
Windows NT and 2000, transforms on 85
Windows printing, customization required 23
WritePrintFile function 291
WS parameter
in DEVICE statement (JES3) 177
in PRTnnnnn statement (JES2) 174

X

xxxx0000.tab file 114

Z

z/OS SNMP agent, customizing and starting 140

Readers' Comments — We'd Like to Hear from You

z/OS
Infoprint Server Customization

Publication No. S544-5744-03

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	☐	☐	☐	☐	☐

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	☐	☐	☐	☐	☐
Complete	☐	☐	☐	☐	☐
Easy to find	☐	☐	☐	☐	☐
Easy to understand	☐	☐	☐	☐	☐
Well organized	☐	☐	☐	☐	☐
Applicable to your tasks	☐	☐	☐	☐	☐

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? ☐ Yes ☐ No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Name _____ Address _____

Company or Organization _____

Phone No. _____

Readers' Comments — We'd Like to Hear from You
S544-5744-03

IBM

Cut or Fold
Along Line

Fold and Tape | **Please do not staple** | Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
Information Development
IBM Printing Systems Division
Department H7FE Building 004M
Boulder, CO 80301-9817

Fold and Tape | **Please do not staple** | Fold and Tape

S544-5744-03

Cut or Fold
Along Line

Program Number: 5694-A01 5697-F51 5655-G52

S544-5744-03

Spine information:

IBM z/OS Infoprint Server Customization S544-5744-03