

How to use the From AFP to PCL Transform on z/OS

Preface:

This paper is intended to help you to configure, customize and use the From AFP to PCL Transform V1.2.0 based on our experience.

For full configuration information of the transform, refer to the [Transform User Guide](#).

Contents

Assumptions before starting this task:.....	3
AFP to PCL Transform Flow	4
One-time configuration actions	5
Register the Transform product.....	5
Run aokcheck	6
Verify the links to the transform code.....	6
Verify the transform parts are installed.....	7
Customization configuration actions	9
Configure aopxfd.conf	9
Verify Transform Daemons have been Started	10
Configure Printer Definition and job selection rules	11
Method 1) Use Infoprint Server ISPF panels (typically 12.8) to create a printer definition and job selection rule to transform output to PCL.....	11
Method 2) Use the PIDU utility to create the printer definition and job selection rule	14
Sample PCL Transform Scenarios.....	17
Scenario 1) Transform Simple Line Data to PCL	17
Scenario 2) Use AOPBATCH to transform simple line data to PCL	18
Scenario 3) Use the UNIX shell command line to transform an AFP input file to an A4 paper size PCL file.....	19
Additional Reference Information	20
How to Restart the aopxfd Daemon:	20
Create the links to the transform code	20
Selecting Direct Sockets protocol vs LPR.....	21
AOP_PAPER, AOP_TRAYID and AOP_PJL environment variables	22
IBM documentation references:	25

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing, or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS," without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Authors:

Anthony Mingo – Advisory Software Engineer, z/OS Print and Transform Development,
Rochester, MN. Email comments to mingo@us.ibm.com

Tariq Choudhry – Senior Software Engineer, z/OS Print and Transform Development, Rochester,
MN

Assumptions before starting this task:

The following assumptions are needed before starting the tasks discussed in this paper:

- Familiar with Unix System Services
- User id configuring the transform has an OMVS segment
- User id configuring the transform is a member of the AOOPER and AOPADMIN RACF groups
- Ability to run from a user ID that is defined to z/OS UNIX System Services, and has the following attributes:
 - o UID(0) or READ access or higher to BPX.SUPERUSER in the FACILITY class.
 - o READ access or higher to BPX.FILEATTR.PROGCTL, BPX.FILEATTR.APF, and BPX.FILEATTR.SHARELIB in the FACILITY classes.
- Infoprint Server is installed, product registration is enabled, and base configuration is setup. For more information refer to the [Infoprint Server Customization Guide](#).
- The Transform products can be ordered from SHOPz Included with this product is the '[Program Directory for IBM Print Transform from AFP to PCL for Infoprint Server for z/OS V1R2M2](#)' GI11-9843-01, that describes how to install the product.

NOTE: The focus for this document is transforming to PCL, however if you would like to transform to Postscript (assuming you have this transform installed), these examples could easily be modified to use Postscript by changing the afpxpcl references to afpxps.

AFP to PCL Transform Flow

The figure shows how to print an AFP document on a PCL printer.

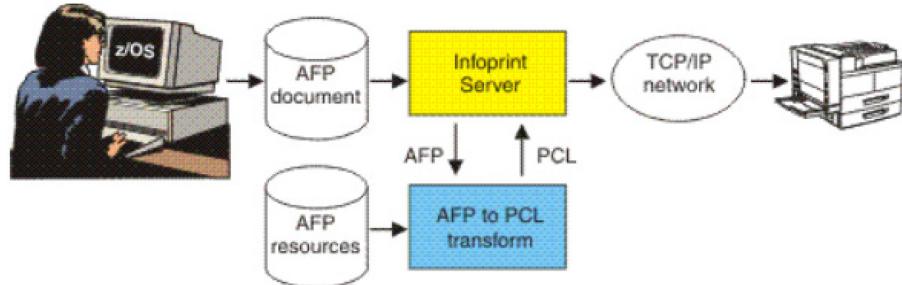


Figure 1. Printing AFP documents on PCL printers

One-time configuration actions

Register the Transform product.

If not already completed, add this text to your IFAPRDxx parmlib member where the z/OS enablement policies reside.

```
/* THIS IS FOR FROM AFP TO PCL TRANSFORM */
PRODUCT OWNER('IBM CORP')
NAME('PRINT TRANSFORMS')
ID(5655-TF2)
VERSION(*) RELEASE(*) MOD(*)
FEATURENAME('AFXPCL')
STATE(ENABLED)
/* */
```

The parmlib member should have this entry as shown highlighted below.

BROWSE SYS1.PARMLIB(IFAPRD00) - 01.20	Line 0
MOD(*)	
FEATURENAME('Download Plus')	
STATE(ENABLED)	
PRODUCT OWNER('IBM CORP')	
NAME ('PSF for z/OS')	
ID(5655-M32)	
VERSION(*)	
RELEASE(*)	
MOD(*)	
FEATURENAME('ACIF')	
STATE(ENABLED)	
PRODUCT OWNER('IBM CORP')	
NAME('PRINT TRANSFORMS')	
ID(5655-TF1)	
VERSION(*) RELEASE(*) MOD(*)	
FEATURENAME('AFXPDF')	
STATE(ENABLED)	
PRODUCT OWNER('IBM CORP')	
NAME('PRINT TRANSFORMS')	
ID(5655-TF2)	
VERSION(*) RELEASE(*) MOD(*)	
FEATURENAME('AFXPCL')	
STATE(ENABLED)	
PRODUCT OWNER('IBM CORP')	
NAME('PRINT TRANSFORMS')	
ID(5655-TF3)	
VERSION(*) RELEASE(*) MOD(*)	
FEATURENAME('AFXPXS')	
STATE(ENABLED)	
PRODUCT OWNER('IBM CORP')	
NAME('PRINT TRANSFORMS')	
ID(5655-TF1)	
VERSION(*) RELEASE(*) MOD(*)	
FEATURENAME('AFPXACCESS PDF')	
STATE(ENABLED)	
/* PDF/S	*
PRODUCT OWNER('IBM CORP')	

Then, issue MVS command 'set prod=(xx)' where the xx matches the parmlib member suffix.

You can use the 'Display Prod, State' command on SDSF command line to check if the transform product is enabled on your system.

Example:

D PROD, STATE					
S OWNER	NAME	FEATURE	VERSION	ID	
E IBM CORP	PRINT TRANSFORMS AFPxPCL		* . * . *	5655-TF2	

Run aokcheck

This may have already been completed if you followed the steps in the program directory manual.

This REXX EXEC checks the setup of From AFP Transforms V1R2.

Review the results for any issues.

Run this exec from UNIX shell. /usr/lpp/IBM/PrintXform/V1R2/samples/aokcheck

Verify the links to the transform code.

If the aoksymlink.sh command was completed as documented in the Program Directory manual, your /usr/lpp/Printsrv/lib and /usr/lpp/Printsrv/lib directories should look like the following. NOTE: The screen shots below are examples from our development system where all transform products are installed. The list you see might be smaller.

```
=>cd /usr/lpp/Printsrv/lib
[MINGO @ TES3] /24AT3/usr/lpp/Printsrv/lib
=>ls -l
total 53744
drwxr-xr-x  2 BPXR00T OMVSGRP    8192 Sep 25  2019 IBM
lrwxrwxrwx  1 BPXR00T OMVSGRP      55 Feb 15 08:07 afpxcrypto.dll -> /usr/lpp/IBM/PrintXform/V1R2/AFPxPDF/11b/afpxcrypto.dll
lrwxrwxrwx  1 BPXR00T OMVSGRP      44 Feb 15 08:07 afpxgum.dll -> /usr/lpp/IBM/PrintXform/V1R2/11b/afpxgum.dll
lrwxrwxrwx  1 BPXR00T OMVSGRP      52 Feb 15 08:07 afpxpcl.dll -> /usr/lpp/IBM/PrintXform/V1R2/AFPxPCL/lib/afpxpcl.dll
lrwxrwxrwx  1 BPXR00T OMVSGRP      53 Feb 15 08:07 afpxpclid.dll -> /usr/lpp/IBM/PrintXform/V1R2/AFPxPCL/lib/afpxpclid.dll
lrwxrwxrwx  1 BPXR00T OMVSGRP      52 Feb 15 08:07 afpxpdf.dll -> /usr/lpp/IBM/PrintXform/V1R2/AFPxPDF/lib/afpxpdf.dll
lrwxrwxrwx  1 BPXR00T OMVSGRP      53 Feb 15 08:07 afpxpdfd.dll -> /usr/lpp/IBM/PrintXform/V1R2/AFPxPDF/lib/afpxpdfd.dll
lrwxrwxrwx  1 BPXR00T OMVSGRP      54 Feb 15 08:07 afpxpdfu.dll -> /usr/lpp/IBM/PrintXform/V1R2/AFPxPDF/lib/afpxpdfu.dll
lrwxrwxrwx  1 BPXR00T OMVSGRP      50 Feb 15 08:07 afpxps.dll -> /usr/lpp/IBM/PrintXform/V1R2/AFPxPS/lib/afpxps.dll
lrwxrwxrwx  1 BPXR00T OMVSGRP      51 Feb 15 08:07 afpxpsd.dll -> /usr/lpp/IBM/PrintXform/V1R2/AFPxPS/lib/afpxpsd.dll
lrwxrwxrwx  1 BPXR00T OMVSGRP      44 Feb 15 08:07 afpxxml.dll -> /usr/lpp/IBM/PrintXform/V1R2/11b/afpxxml.dll
lrwxrwxrwx  1 BPXR00T OMVSGRP      47 Feb 15 08:07 afpxxmlb0.dll -> /usr/lpp/IBM/PrintXform/V1R2/11b/afpxxmlb0.dll
lrwxrwxrwx  1 BPXR00T OMVSGRP      47 Feb 15 08:07 aokpdfexit.dll -> /usr/lpp/IBM/PrintXform/V1R2/lib/aokpdfexit.dll
-rwxr-xr-x  2 BPXR00T OMVSGRP  5726208 Oct 26 12:54 aop.so
-rwxr-xr-x  2 BPXR00T OMVSGRP   32768 Oct 26 12:54 aopapi.dll
-rwxr-xr-x  2 BPXR00T OMVSGRP  528384 Oct 26 12:54 aopapi2.dll
-rwxr-xr-x  2 BPXR00T OMVSGRP  5558272 Sep 25  2019 aopcop.dll
-rwxr-xr-x  2 BPXR00T OMVSGRP 1204224 Oct 26 12:54 aopdb.so
-rwxr-xr-x  2 BPXR00T OMVSGRP 1978368 Oct 26 12:54 aopeapi.dll
-rwxr-xr-x  2 BPXR00T OMVSGRP   20480 Oct 26 12:54 aopfiltr.so
-rwxr-xr-x  2 BPXR00T OMVSGRP  122880 Oct 26 12:54 aopipc.so
-rwxr-xr-x  2 BPXR00T OMVSGRP  3313664 Oct 26 12:54 aopipc2.so
-rwxr-xr-x  2 BPXR00T OMVSGRP  241664 Oct 26 12:54 aoprform.dll
```

```
-- --
[MINGO @ TES3] /24AT3/usr/lpp/Printsrv/bin
=ls -l
total 130032
drwxr-xr-x 2 BPXROOT OMVSGRP 6192 Sep 25 2019 IBM
lrwxrwxrwx 1 BPXROOT OMVSGRP 48 Feb 15 08:07 afpxpcl -> /usr/lpp/IBM/PrintXform/V1R2/AFPxPCL/bin/afpxpcl
lrwxrwxrwx 1 BPXROOT OMVSGRP 49 Feb 15 08:07 afpxpcld -> /usr/lpp/IBM/PrintXform/V1R2/AFPxPCL/bin/afpxpcld
lrwxrwxrwx 1 BPXROOT OMVSGRP 48 Feb 15 08:07 afpxpdf -> /usr/lpp/IBM/PrintXform/V1R2/AFPxPDF/bin/afpxpdf
lrwxrwxrwx 1 BPXROOT OMVSGRP 49 Feb 15 08:07 afpxpdfd -> /usr/lpp/IBM/PrintXform/V1R2/AFPxPDF/bin/afpxpdfd
lrwxrwxrwx 1 BPXROOT OMVSGRP 46 Feb 15 08:07 afpxps -> /usr/lpp/IBM/PrintXform/V1R2/AFPxPS/bin/afpxps
lrwxrwxrwx 1 BPXROOT OMVSGRP 47 Feb 15 08:07 afpxpsd -> /usr/lpp/IBM/PrintXform/V1R2/AFPxPS/bin/afpxpsd
-rwrxr-xr-x 2 BPXROOT OMVSGRP 2215936 Sep 25 2019 aopchkinv
-rwxr----- 2 BPXROOT OMVSGRP 20480 Sep 25 2019 aopcstart
-rwxr--S--- 2 BPXROOT AOPADMIN 3608576 Oct 26 12:54 aopd
-rwsr-x---
```

The following fonts should reside in /usr/lpp/IBM/PrintXform/V1R2/aokfonts.

```
=>cd /usr/lpp/Printsrv/aokfonts
[MINGO @ TES3] /24AT3/usr/lpp/IBM/PrintXform/V1R2/aokfonts
=ls -l
total 4144
-rw-r--r-- 2 BPXROOT OMVSGRP 28215 Jan 9 2020 COU.AFM
-rw-r--r-- 2 BPXROOT OMVSGRP 115319 Jan 9 2020 COU.PFB
-rw-r--r-- 2 BPXROOT OMVSGRP 28186 Jan 9 2020 COUB.AFM
-rw-r--r-- 2 BPXROOT OMVSGRP 119235 Jan 9 2020 COUB.PFB
-rw-r--r-- 2 BPXROOT OMVSGRP 28315 Jan 9 2020 COUBI.AFM
-rw-r--r-- 2 BPXROOT OMVSGRP 114454 Jan 9 2020 COUBI.PFB
-rw-r--r-- 2 BPXROOT OMVSGRP 28361 Jan 9 2020 COUI.AFM
-rw-r--r-- 2 BPXROOT OMVSGRP 115193 Jan 9 2020 COUI.PFB
-rw-r--r-- 2 BPXROOT OMVSGRP 28201 Jan 9 2020 HEL.AFM
-rw-r--r-- 2 BPXROOT OMVSGRP 103270 Jan 9 2020 HEL.PFB
-rw-r--r-- 2 BPXROOT OMVSGRP 28191 Jan 9 2020 HELB.AFM
-rw-r--r-- 2 BPXROOT OMVSGRP 105822 Jan 9 2020 HELB.PFB
-rw-r--r-- 2 BPXROOT OMVSGRP 28373 Jan 9 2020 HELBI.AFM
-rw-r--r-- 2 BPXROOT OMVSGRP 106454 Jan 9 2020 HELBI.PFB
-rw-r--r-- 2 BPXROOT OMVSGRP 28371 Jan 9 2020 HELI.AFM
-rw-r--r-- 2 BPXROOT OMVSGRP 105356 Jan 9 2020 HELI.PFB
-rw-r--r-- 2 BPXROOT OMVSGRP 28150 Jan 9 2020 TNR.AFM
-rw-r--r-- 2 BPXROOT OMVSGRP 123986 Jan 9 2020 TNR.PFB
-rw-r--r-- 2 BPXROOT OMVSGRP 28202 Jan 9 2020 TNRB.AFM
-rw-r--r-- 2 BPXROOT OMVSGRP 127761 Jan 9 2020 TNRB.PFB
-rw-r--r-- 2 BPXROOT OMVSGRP 28493 Jan 9 2020 TNRBI.AFM
-rw-r--r-- 2 BPXROOT OMVSGRP 138524 Jan 9 2020 TNRBI.PFB
-rw-r--r-- 2 BPXROOT OMVSGRP 28504 Jan 9 2020 TNRI.AFM
-rw-r--r-- 2 BPXROOT OMVSGRP 135768 Jan 9 2020 TNRI.PFB
-rw-r--r-- 2 BPXROOT OMVSGRP 22085 Jan 9 2020 TNR_S.AFM
-rw-r--r-- 2 BPXROOT OMVSGRP 78304 Jan 9 2020 TNR_S.PFB
-rw-r--r-- 2 BPXROOT OMVSGRP 9459 Jan 9 2020 ZD_____._AFM
-rw-r--r-- 2 BPXROOT OMVSGRP 41711 Jan 9 2020 ZD_____._PFB
[MINGO @ TES3] /24AT3/usr/lpp/IBM/PrintXform/V1R2/aokfonts
```

Verify the transform parts are installed.

Run this exec from UNIX shell:

```
/usr/lpp/IBM/PrintXform/V1R2/samples/aokbuildinfo.sh
```

NOTE: The screen shot below is an example from our development system where all transform products are installed. The list you see might be smaller. The two highlighted parts should be there.

```
=>/usr/lpp/IBM/PrintXform/V1R2/samples/aokbuildinfo.sh
Wed Feb 15 11:05:45 MST 2023
This is build information for From AFP Transforms Product ID=5655TFX00 V1R2
Provide this information when submitting a problem record to IBM.
checking AFPxPCL in /usr/lpp/IBM/PrintXform/V1R2/AFPxPCL/lib/afpxpcld.dll level
  01/27/2023  OA64253 BUILD_ID=7672 20230127
checking AFPxPCL in /usr/lpp/IBM/PrintXform/V1R2/AFPxPCL/bin/afpxpcld level
  01/27/2023  OA64253 BUILD_ID=7672 20230127
checking AFPxPDF in /usr/lpp/IBM/PrintXform/V1R2/AFPxPDF/lib/afpxpdfd.dll level
  02/01/2023  OA64253 BUILD_ID=7686 20230201
checking Accessible PDF in /usr/lpp/IBM/PrintXform/V1R2/AFPxPDF/lib/afpxpdfua.dll level
  12/15/2022  OA63041 BUILD_ID=7639 20221215
checking Signed PDF in /usr/lpp/IBM/PrintXform/V1R2/AFPxPDF/lib/afpxcrypto.dll level
  12/15/2022  OA63041 BUILD_ID=7639 20221215
checking AFPxPDF in /usr/lpp/IBM/PrintXform/V1R2/AFPxPDF/bin/afpxpdfd level
  01/27/2023  OA64253 BUILD_ID=7672 20230127
checking AFPxPS in /usr/lpp/IBM/PrintXform/V1R2/AFPxPS/lib/afpxpsd.dll level
  01/27/2023  OA64253 BUILD_ID=7672 20230127
checking AFPxPS in /usr/lpp/IBM/PrintXform/V1R2/AFPxPS/bin/afpxpsd level
  01/27/2023  OA64253 BUILD_ID=7672 20230127
checking XML common in /usr/lpp/IBM/PrintXform/V1R2/lib/afpxxml.dll level
  12/15/2022  OA63041 BUILD_ID=7639 20221215
checking XML common in /usr/lpp/IBM/PrintXform/V1R2/lib/afpxxmlb0.dll level
  12/15/2022  OA63041 BUILD_ID=7639 20221215
checking GUM common in /usr/lpp/IBM/PrintXform/V1R2/lib/afpxgum.dll level
  12/12/2022  OA63041 BUILD_ID=7636 20221212
[MINGO @ TES3] /24AT3/usr/lpp/IBM/PrintXform/V1R2/samples
```

Customization configuration actions

Configure aopxfd.conf

The Transform User Guide Chapter 3 describes how to Customize the Transforms. The /etc/Printsrv/aopxfd.conf file is where the transform classes are specified and is needed for the transforms to function. For more general information for customizing transforms see [Chapter 5 in the InfoPrint Server Customization Manual](#).

The following examples are sample transform entries in the InfoPrint Server transform configuration file (/etc/Printsrv/aopxfd.conf) for the AFP to PCL transform. A sample aopxfd.conf was provided in the zip file bundled with this document. FTP (in ASCII) to /etc/Printsrv/aopxfd.conf and restart the aopxfd daemon.

aopxfd.conf transform class examples:

```
-----
# AFP to PCL Transform - Basic Transform Class
-----
transform afpxpcl
  start-command = "afpxpcld"
  min-active = 0
  max-active = 2
  maximum-idle-time = 300 # 5 minutes
  environment = {
    AOP_COLOR -> yes
    AOP_FONTLIB -> "sys1.font300 sys1.fontlib sys1.fontlibb sys1.sfondlib
sys1.sfntilib"
    AOP_FORMDEFLIB -> "sys1.fdeflib common.formdef"
    AOP_PAGEDEFLIB -> "sys1.pdeflib common.pagedef"
    AOP_PAGESEGLIB -> "sys1.pseglib common.segment"
    AOP_OVERLAYLIB -> "sys1.overlay common.overlay"
    AOP_PJL -> no
    AOP_PAPER -> "letter letter letter letter letter letter letter letter
letter"
    AOP_TRAYID -> "1 1 1 1 1 1 1 1 1"
    _BPX_JOBNAME -> AFPXPCLD
  }
;

-----
# AFP to PCL Transform - Color NO and Print on A4 size paper
-----
transform afpxpcl_colornoa4
  start-command = "afpxpcld -l"
  min-active = 0
  max-active = 2
  maximum-idle-time = 300 # 5 minutes
  environment = {
    AOP_COLOR -> no
    AOP_FONTLIB -> "sys1.sfntilib sys1.font300 sys1.fontlib sys1.fontlibb
sys1.sfondlib"
    AOP_FORMDEFLIB -> "sys1.fdeflib common.formdef"
```

```

AOP_PAGEDEFLIB -> "sys1.pdeflib common.pagedef"
AOP_PAGESEGLIB -> "sys1.pseglib common.segment"
AOP_OVERLAYLIB -> "sys1.overlaylib common.overlay"
AOP_PJL -> no
AOP_PAPER -> "a4 a4 a4 a4 a4 a4 a4 a4 a4"
AOP_TRAYID -> "1 1 1 1 1 1 1 1 1"
_BPX_JOBNAME -> AFPXPCLD
}
;

```

NOTE: The AOP_xxxxxxxLIB and AOP_FONTLIB parameters in the examples above are using resource libraries (sys1.fdeflib common.xxxx) might not exist on your system. If they do not exist, replace them with your resource libraries.

NOTE: Refer to the [Environment variables for the AFP to PCL transform](#) for a list of environment variables to customize the AFP to PCL transform. For the To PCL transform, some key environment variables are [AOP_PAPER](#), [AOP_TRAYID](#) and [AOP_PJL](#), described at the bottom of this document.

After adding or changing a transform class in aopxfd.conf, restart the aopxfd daemon. Users who start Infoprint Server must be members of the AOPOPER RACF group or have a UID of 0.

Verify Transform Daemons have been Started

Ensure that transform daemon, AOPXFD has been started. This can be verified using the 'aopstat' command at the z/OS UNIX shell command line. The AOPXFD daemon should be restarted when changes are made to the aopxfd.conf configuration file.

=>aopstat				
Member	Job ID	System	Status	State
AOPIBM00	AOPD	TES3	ACTIVE	Ready
XFDIBM00	AOPXFD	TES3	ACTIVE	Ready
LPDIBM00	AOPLPD	TES3	ACTIVE	Ready
IPPIIBM00	AOPIPDD	TES3	ACTIVE	Ready
SUBIBM00	AOPSUBD	TES3	ACTIVE	Ready
NETIBM00	AOPNETD	TES3	ACTIVE	Ready
SSIIBM00	AOPSSID	TES3	ACTIVE	Ready
WSMIBM00	AOPWSMD	TES3	ACTIVE	Ready
OUTIBM00	AOPOUTD	TES3	ACTIVE	Ready

The daemons that are started are specified in /etc/Printsrv/aopd.conf. For more information, refer to [Infoprint Server Customization, Chapter 3](#). At minimum you need to have 'outd' and 'xfd' daemons started.

```
start-daemons = { ippd lpd netd outd ssid subd xfd }
```

NOTE: If the Infoprint Server ISPF, System (Option 8) Configuration panel, Operating mode parameter is set to z/OS 2.2 mode, this start-daemons parameter from aopd.conf is ignored.

Configure Printer Definition and job selection rules

There are a couple of ways to configure a printer definition, using IPSF panels or using the PIDU.

Method 1) Use Infoprint Server ISPF panels (typically 12.8) to create a printer definition and job selection rule to transform output to PCL.

Select (1) Add a printer definition

Option ==> 1	Infoprint Server: Printer Inventory Manager
Printer Definitions	
1 Add	Add a printer definition
2 List	List printer definitions
3 Select	Select printer definitions to list
Other Functions	
4 Other Definitions	Manage FSS, FSA, pool, and job selection definitions
5 PrintWay Queue	View IP PrintWay basic mode transmission queue
6 PrintWay Message	View IP PrintWay basic mode message log
Infoprint Server Configuration	
7 ISPF	Manage ISPF panel configuration
8 System	Manage system configuration

The printer definition should have the following settings:

Infoprint Server: Printer Inventory Manager Choose a Definition Type and Protocol				
Option ==> 2				
Type	Protocol			
1 IP PrintWay	LPR			
2 IP PrintWay	direct sockets			
3 IP PrintWay	IPP			
4 IP PrintWay	VTAM			
5 IP PrintWay	email			
6 PSF				
7 General				
F1=HELP	F2=SPLIT	F3=END	F4=RETURN	F5=RFIN
F6=RCHANGE	F7=UP	F8=DOWN	F9=SWAP	F10=LEFT

Give the printer definition a name and description

Add			IP PrintWay Printer Definition
Command ==>			
Printer definition name . XFORMPCL			
Description . Transform Output to PCL and Send to Printer			(extend)
Location . .			(extend)
Section	Component name (enter to list)	Custom values (enter to customize)	
Allocation	=> _____	=> /	
Processing	=> _____	=> *	
NetSpool options	=> _____	=> —	
NetSpool end-of-file	=> _____	=> —	
IP PrintWay options	=> _____	=> —	
Protocol	=> _____	=> *	
/ Use DEST, CLASS, and FORMS for IP PrintWay printer selection			
NetSpool LU name . _____ LU classes . . — — — — — (extend)			

Allocation Panel

Choose a CLASS and DEST

Add	Allocation				
Command ==>	<hr/>				
Printer definition name .	XFORMPCL				
More: +					
Spool allocation values:					
CLASS	J	GROUPID . . .	<hr/>		
DEST.	XPCL	LINECT. . .	<hr/>		
JES node. . .	<hr/>	PRMODE. . .	<hr/>		
FCB . . .	<hr/>	PRTY. . .	<hr/>		
FLASH count .	<hr/>	THRESHLD. .	<hr/>		
FLASH name. .	<hr/>	UCS . . .	<hr/>		
FORMS	<hr/>	WRITER. . .	<hr/>		
USERDATA	<hr/>	(extend)			
BURST	-	1. Yes	2. No		
HOLD.	-	1. Yes	2. No		
OUTDISP	-	1. Purge (without printing)	2. Leave		
		3. Keep	4. Hold		
		5. Write			
Values for Separator Pages:					
Address	<hr/>	(extend)			
Building .	<hr/>				
Department .	<hr/>				
Name	<hr/>				
Room	<hr/>				
Title	<hr/>				
Resource Related Values:					
Form definition .	<hr/>				
Character sets .	<hr/>	Back . .	<hr/>		
Overlay front .	<hr/>	Back . .	<hr/>		
Input tray . . .	<hr/>				
Output bin .	<hr/>				
Page definition .	<hr/>				
Resource library.	<hr/>	(extend)			
Resource directories.					
Image shift x-direction front . .	<hr/>	Back . .	<hr/>		
y-direction front . .	<hr/>	Back . .	<hr/>		
Error Reporting Values:					
Print error reporting.	-	1. None	2. All		
F1=HELP	F2=SPLIT	F3=END	F4=RETURN	F5=RFIND	F6=RCHANGE
F7=UP	F8=DOWN	F9=SWAP	F10=LEFT	F11=RIGHT	F12=RETRIEVE

Processing Panel

Set the dll name to process the input types. For the AFP to PCL transform, the dll name is afpxpcl.dll

Add	Processing				
Command ==>	<hr/>				
Printer definition name .	XFORMPCL				
More: +					
Document code page . .					
Printer code page. . .	IS08859-1				
Supported Data Formats and Associated Filters:					
Data format:	Filter:				
/ Line data	afpxpcl.dll	(extend)			
/ M0:DCA-P	afpxpcl.dll	(extend)			
/ PostScript	<hr/>	(extend)			
/ Text	aopfiltr.so	(extend)			
/ PCL	<hr/>	(extend)			
/ PDF	<hr/>	(extend)			
/ SAP	<hr/>	(extend)			
/ XML	afpxpcl.dll	(extend)			
/ TIFF	<hr/>	(extend)			
/ JPEG	<hr/>	(extend)			
/ Other	<hr/>	(extend)			
<u>_ Resubmit for filtering</u>					
Transforms Error Handling:					
Fail on error	-	1. No	2. Error	3. Warning	
Trailer error page. . .	-	1. No	2. Error	3. Warning	
AFP to PDF Transform Encryption:					
User identifier .	<hr/>	(extend)			
Owner identifier.	<hr/>	(extend)			
Encryption level	2	1. Low (40-bit key)	2. High (128-bit key)		
		3. AES (128-bit key)	4. AES (256-bit key)		
Protected actions:					
Restrict print. . .	2	1. Yes	2. No		
Restrict copy . . .	2	1. Yes	2. No		
Restrict update . .	2	1. Yes	2. No		
Accessible PDF Transform:					
User accessibility control file .	<hr/>	(extend)			
F1=HELP	F2=SPLIT	F3=END	F4=RETURN	F5=RFIND	F6=RCHANGE
F7=UP	F8=DOWN	F9=SWAP	F10=LEFT	F11=RIGHT	F12=RETRIEVE

Protocol Panel

Set the IP address and TCP port number for the printer. The port number can vary for different printer models.

```
Add                               Direct Sockets Protocol
Command ==> _____
Printer definition name . XFORMPCL
Operator security profile
. .
Printer IP address . x.x.x.x
Port number. . . . 3100 (extend)
Printer Job Language (PJL) options:
- Record pages printed for accounting
- Restart printing after last successful page
```

Make sure **Use DEST, CLASS and Form for IP PrintWay printer selection** is checked

```
Add                               IP PrintWay Printer Definition
Command ==> _____
Printer definition name . XFORMPCL
Description . Transform Output to PCL and Send to Printer (extend)
Location. . . (extend)
Section          Component name
                 (enter to list)      Custom values
Allocation       => _____      => *
Processing       => _____      => *
NetSpool options => _____      => __
NetSpool end-of-file => _____      => __
IP PrintWay options => _____      => *
Protocol         => _____      => *
Use DEST, CLASS, and FORMS for IP PrintWay printer selection
NetSpool LU name . LU classes . . . . . . . . (extend)
```

Next, create the job selection rule for DEST and CLASS you used above.

```
InfoPrint Server: Printer Inventory Manager
Option ==> 4
Printer Definitions
 1 Add           Add a printer definition
 2 List          List printer definitions
 3 Select        Select printer definitions to list
Other Functions
 4 Other Definitions Manage FSS, FSA, pool, and job selection definitions
 5 PrintWay Queue View IP PrintWay basic mode transmission queue
 6 PrintWay Message View IP PrintWay basic mode message log
InfoPrint Server Configuration
 7 ISPF          Manage ISPF panel configuration
 8 System        Manage system configuration

F1=HELP          F2=SPLIT          F3=END            F4=RETURN          F5=RFIND          F6=RCHANGE
F7=UP           F8=DOWN           F9=SWAP           F10=LEFT           F11=RIGHT         F12=RETRIEVE
```

```

FSA, FSS, Pool, and Job Selection Rule Management
option ==> 10

FSA
  1 Add          Add an FSA
  2 List         List FSAs
  3 Select       Select FSAs to list

FSS
  4 Add          Add an FSS
  5 List         List FSSs
  6 Select       Select FSSs to list

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

Job Selection Rule
10 Add          Add a Job Selection Rule
  11 List        List Job Selection Rules

F1=HELP      F2=SPLIT      F3=END      F4=RETURN      F5=RFINDD      F6=RCHANGE
F7=UP        F8=DOWN       F9=SWAP      F10=LEFT       F11=RIGHT      F12=RETRIEVE

```

```

Add                                         Job Selection Rule
Command ==> _____
Rule name . . XPCL
Description . . _____ (extend)
Operator security profile
  .
  .
DEST. . . XPCL
CLASS . . J - - - - -
FORMS . . - - - - -
Creator . . _____
WRITER. . . _____
DEST IP address . . 3 1. Include 2. Exclude 3. Ignore
Record limit . . . . Lower _____ Upper _____
Page limit (AFP-only): Lower _____ Upper _____

```

Method 2) Use the PIDU utility to create the printer definition and job selection rule
An alternative to using ISPF panels to create printer definition and job selection rules is to use PIDU. This utility is recommended if you want to create several Printers or Job Selection rules. Place the following text in a text file such as 'pidu.txt.' Change printer-ip-address (x.x.x.x) and port-number to match your printer. Upload the file to a target directory on USS and use the 'pidu pidu.txt' command at the z/OS UNIX shell command line to add this printer definition to the Infoprint Server Inventory.

The following pidu example will create one direct-socket printer and one lpr printer. For the difference between the two, see '[Selecting Direct Sockets protocol vs LPR](#)' below.

NOTE: The user profile issuing the pidu command to add printers or printer definitions must be a member of the AOPADMIN RACF group.

pidu examples:

```
#-----
# Printer XFORMPCL - Transform to PCL and Send to Printer
#-----
create printer XFORMPCL
    printer-codepage = ISO8859-1
    filters = {
        line -> afpxpcl.dll
        modca -> afpxpcl.dll
        text -> aopfiltr.so
        xml -> afpxpcl.dll
    }
    printer-type = ip-printway
    protocol-type = direct-sockets
    port-number = 9100
    destination = XPCL
    output-class = J
    document-formats-supported = {
        line
        modca
        pcl
        xml
    }
    response-timeout = 120
    printer-ip-address = x.x.x.x
    dcf-routing = yes
    description = "Transform Output to PCL and Send to Printer"
;

create dcf-routing-key "XPCL J"
    printer = XFORMPCL
;

#-----
# job-selection-rule XPCL
#-----
create job-selection-rule XPCL
    output-class-list = {
        J
    }
    destination-pattern = XPCL
;

#-----
# Printer XA4PCL - Transform to PCL and LPR to Printer using A4 paper
#-----
create printer XA4PCL
    printer-codepage = ISO8859-1
    filters = {
        line -> "afpxpcl.dll -c colornoa4"
        modca -> "afpxpcl.dll -c colornoa4"
        text -> aopfiltr.so
        xml -> "afpxpcl.dll -c colornoa4"
    }
```

```
printer-type = ip-printway
protocol-type = lpr
destination = XA4PCL
output-class = J
description = "Print to PCL Printer Using LPR"
print-queue-name = text
printer-ip-address = x.x.x.x
dcf-routing = yes
;

create dcf-routing-key "XA4PCL J"
    printer = XA4PCL
;

#-----
# job-selection-rule XA4PCL
#-----
create job-selection-rule XA4PCL
    output-class-list = {
        J
    }
    destination-pattern = XA4PCL
;
```

The PIDU command output should show that the printer and job selection rules was created.

```
=>pidu pidu.txt
AOP062I printer XFORMPCL was created.
AOP062I dcf-routing-key XPCL J was created.
AOP062I job-selection-rule XPCL was created.
AOP062I printer XA4PCL was created.
AOP062I dcf-routing-key XA4PCL J was created.
AOP062I job-selection-rule XA4PCL was created.
```

NOTE: You can confirm the new Printer was created using option 2 (List Printer Definitions) from the Main Infoprint Server ISPF panel (typically 12.8.2). You can confirm the new Selection rule was created using option 4.11 (List Job Selection Rules) from the Main Infoprint Server ISPF panel (typically 12.8.4.11).

Sample PCL Transform Scenarios

Once the system is configured as explained, here are some ways to use the PCL transform.

Scenario 1) Transform Simple Line Data to PCL

Create or name an input file to pass into the transform

Input data can come in various forms, Line data, AFP, or XML. For this example, we will use a simple line data file.

Here is sample input data we are using for this simple example:

```
BROWSE XFORM.AFPDOC.FB80(ONEPAGE) - 01.12 Line 0000000000 Col 001 080
***** Top of Data *****
PAGE-01 LINE-001 ABCDEFGHIJKLMNOPQRSTUVWXYZ ABCDEFGHIJKLMNOPQRSTUVWXYZ REC-0001
PAGE-01 LINE-002 ABCDEFGHIJKLMNOPQRSTUVWXYZ ABCDEFGHIJKLMNOPQRSTUVWXYZ REC-0002
PAGE-01 LINE-003 ABCDEFGHIJKLMNOPQRSTUVWXYZ ABCDEFGHIJKLMNOPQRSTUVWXYZ REC-0003
PAGE-01 LINE-004 ABCDEFGHIJKLMNOPQRSTUVWXYZ ABCDEFGHIJKLMNOPQRSTUVWXYZ REC-0004
PAGE-01 LINE-005 ABCDEFGHIJKLMNOPQRSTUVWXYZ ABCDEFGHIJKLMNOPQRSTUVWXYZ REC-0005
PAGE-01 LINE-006 ABCDEFGHIJKLMNOPQRSTUVWXYZ ABCDEFGHIJKLMNOPQRSTUVWXYZ REC-0006
PAGE-01 LINE-007 ABCDEFGHIJKLMNOPQRSTUVWXYZ ABCDEFGHIJKLMNOPQRSTUVWXYZ REC-0007
PAGE-01 LINE-008 ABCDEFGHIJKLMNOPQRSTUVWXYZ ABCDEFGHIJKLMNOPQRSTUVWXYZ REC-0008
PAGE-01 LINE-009 ABCDEFGHIJKLMNOPQRSTUVWXYZ ABCDEFGHIJKLMNOPQRSTUVWXYZ REC-0009
PAGE-01 LINE-010 ABCDEFGHIJKLMNOPQRSTUVWXYZ ABCDEFGHIJKLMNOPQRSTUVWXYZ REC-0010
PAGE-01 LINE-011 ABCDEFGHIJKLMNOPQRSTUVWXYZ ABCDEFGHIJKLMNOPQRSTUVWXYZ REC-0011
PAGE-01 LINE-012 ABCDEFGHIJKLMNOPQRSTUVWXYZ ABCDEFGHIJKLMNOPQRSTUVWXYZ REC-0012
PAGE-01 LINE-013 ABCDEFGHIJKLMNOPQRSTUVWXYZ ABCDEFGHIJKLMNOPQRSTUVWXYZ REC-0013
PAGE-01 LINE-014 ABCDEFGHIJKLMNOPQRSTUVWXYZ ABCDEFGHIJKLMNOPQRSTUVWXYZ REC-0014
PAGE-01 LINE-015 ABCDEFGHIJKLMNOPQRSTUVWXYZ ABCDEFGHIJKLMNOPQRSTUVWXYZ REC-0015
PAGE-01 LINE-016 ABCDEFGHIJKLMNOPQRSTUVWXYZ ABCDEFGHIJKLMNOPQRSTUVWXYZ REC-0016
PAGE-01 LINE-017 ABCDEFGHIJKLMNOPQRSTUVWXYZ ABCDEFGHIJKLMNOPQRSTUVWXYZ REC-0017
PAGE-01 LINE-018 ABCDEFGHIJKLMNOPQRSTUVWXYZ ABCDEFGHIJKLMNOPQRSTUVWXYZ REC-0018
PAGE-01 LINE-019 ABCDEFGHIJKLMNOPQRSTUVWXYZ ABCDEFGHIJKLMNOPQRSTUVWXYZ REC-0019
PAGE-01 LINE-020 ABCDEFGHIJKLMNOPQRSTUVWXYZ ABCDEFGHIJKLMNOPQRSTUVWXYZ REC-0020
***** Bottom of Data *****
```

Input data set XFORM.AFPDOC.FB80 is a PDS with Organization = PO, Record format = FB, Record length = 80, Block size = 12960

Next, create JCL to output the data above to JES Spool. Note, we are using CLASS=J and DEST=XPCL which matches the printer definition we created above. Once the JCL job is submitted, if all is working correctly, IP Printway should pick the input from JES spool and transform it to PCL and send it to the printer.

Note: This example is using a default form and page definition.

```
//ONEPAGE JOB (TST10000),ONEPAGE,CLASS=A,MSGCLASS=T,
// MSGLEVEL=(1,1),TIME=1440,REGION=0M
//*
//STEP01 EXEC PGM=IEBGENER,REGION=0M
//OUTPRT OUTPUT CLASS=J,
// FORMS=STD,
// DEST=MAIL
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DISP=SHR,DSN=XFORM.AFPDOC.FB80(ONEPAGE)
//SYSUT2 DD SYSOUT=(,),DCB=RECFM=FB,
// OUTPUT=(*.OUTPRT)
//SYSIN DD DUMMY
```

Scenario 2) Use AOPBATCH to transform simple line data to PCL

Submit this JCL and it should create PCL output in data set XFORM.AOPBTEST.ONEPAGE.PCL. As you can see, it uses XFORM.AFPDOC.FBA80(ONEPAGE) as input, the same simple line data example in Scenario 1. The second JCL step will route the PCL output to printer XFORMPCL using CLASS=J and DEST=XPCL. Using REGION=0M grants the job access to all available storage, dependent on any site-specific limits.

```
***** Top of Data *****
//AOPBTEST JOB (TEST0101),
//           CLASS=A,MSGCLASS=A,MSGLEVEL=(1,1),TIME=1440,REGION=0M
//***** 
//** DELETE THE TEMP OUTPUT FILE BEFORE STARTING
//***** 
//DELETE EXEC PGM=IEFBR14
//DD1      DD DSN=XFORM.AOPBTEST.ONEPAGE.PCL,
//           DISP=(MOD,DELETE),
//           SPACE=(TRK,(0)),UNIT=SYSALLDA
//***** 
//** Test Case:
//** This test case passes input and output files as //DD names
//** Expected results - Successful completion and a valid PCL output
//***** 
//S1      EXEC PGM=AOPBATCH,
// PARM='/afpxpcl -j attr=/DD:ATTRS -o //DD:OUTPUT //DD:IN'
//*
//IN       DD DSN=XFORM.AFPDOC.FBA80(ONEPAGE),DISP=SHR
//OUTPUT    DD DSN=XFORM.AOPBTEST.ONEPAGE.PCL,DISP=(NEW,CATLG,DELETE),
//           DCB=(RECFM=VB,LRECL=32756,BLKSIZE=32760),SPACE=(CYL,(5,5))
//ATTRS     DD *
fail-on-transform-error=error
trailer-transform-error-page=warning
/*
//STDOUT    DD SYSOUT=*
//STDERR    DD SYSOUT=*
//STDENV    DD *
PATH=/usr/lpp/Printsrv/bin:/bin:/usr/sbin
LIBPATH=/usr/lpp/Printsrv/lib
NLSPATH=/usr/lpp/Printsrv/%L/%N
/*
//S2      EXEC PGM=IEBGENER
//OUT1 OUTPUT CLASS=J,DEST=XPCL,
//           PRTATTRS='document-format=pcl'
//*
//SYSUT1    DD DSN=XFORM.AOPBTEST.ONEPAGE.PCL,DISP=SHR
//SYSUT2    DD SYSOUT=(,),OUTPUT=(*.OUT1)
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD DUMMY
***** Bottom of Data *****
```

Scenario 3) Use the UNIX shell command line to transform an AFP input file to an A4 paper size PCL file

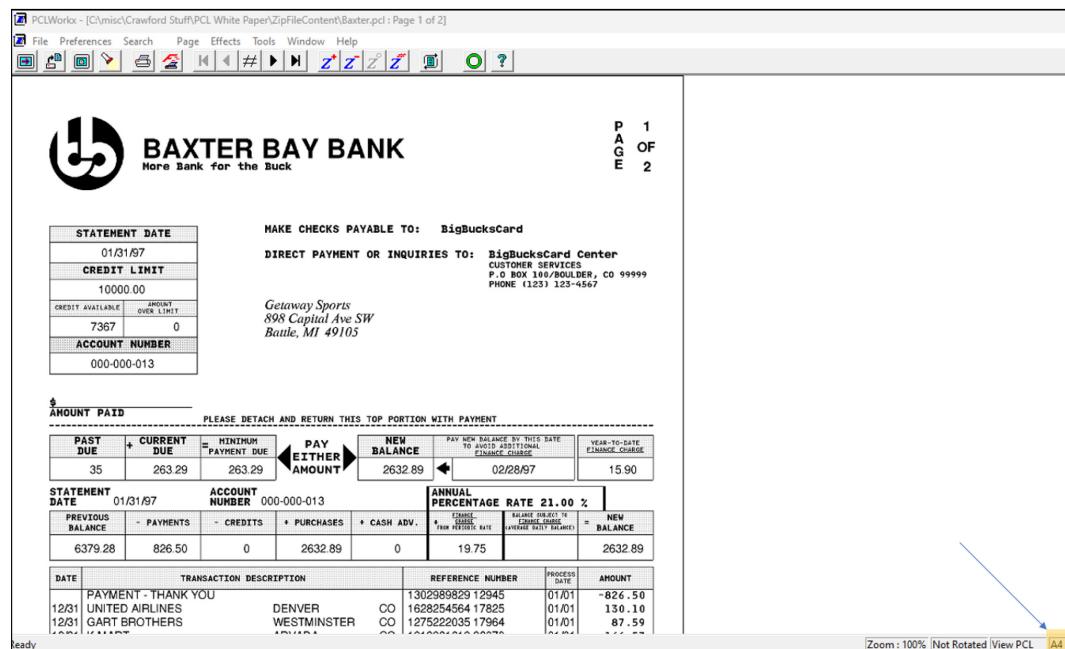
Assumptions:

- 1) Your UNIX PATH and LIBPATH environment variables should be set properly to find the transform components (/usr/lpp/Printsrv/bin and /usr/lpp/Printsrv/lib). This will ensure that the afpxpcl command can be found.
- 2) The AFP binary input file baxter.afp exists in a location in your UNIX environment. The baxter.afp file is provided in the zip file bundled with this document. FTP (in binary) to a UNIX directory location.

Run the following command on the UNIX shell command line. It should create an output file called baxter.pcl. Replace <unix directory> with the UNIX directory where you placed baxter.afp.

```
afpxpcl -o baxter.pcl -c colornoa4 <unix directory>/baxter.afp
```

Since the colornoa4 transform class is using A4 paper sizes, the resulting PCL output should have a paper size of A4.



Additional Reference Information

How to Restart the aopxfd Daemon:

o Do one of these to stop aopxfd:

- Use the AOPSTOP JCL procedure on the SDSF command line:

```
START AOPSTOP,OPTIONS='-d xfd'
```

- At a UNIX command prompt enter:

```
aopstop -d xfd
```

o Do one of these to restart aopxfd:

- Use the AOPSTART JCL procedure on the SDSF command line:

```
START AOPDEMON,TYPE=XFD
```

- At a UNIX command prompt enter:

```
aopstart
```

Note: Using the AOPSTART JCL procedure is the preferred method

for starting the daemons because it lets you set the

TIME=NOLIMIT or REGION=<size> JCL parameters.

Create the links to the transform code

If Step 3 above was not successful Run the following exec from a UNIX shell to create links from /usr/lpp/Printsrv to /usr/lpp/IBM/PrintXform/V1R2.

```
/usr/lpp/IBM/PrintXform/V1R2/samples/aoksymlink.sh -f slinks -i  
/usr/lpp/IBM/PrintXform/V1R2 -b /usr/lpp/Printsrv
```

Once this step is completed, your /usr/lpp/Printsrv/lib and /usr/lpp/Printsrv/lib directories should look like the following. NOTE: The screen shots below are examples from our development system where all transform products are installed. The list you see might be smaller.

```
=>cd /usr/lpp/Printsrv/lib
[MINGO @ TES3] /24AT3/usr/lpp/Printsrv/lib
=>ls -l
total 53744
drwxr-xr-x 2 BPXROOT OMVSGRP 8192 Sep 25 2019 IBM
lrwxrwxrwx 1 BPXROOT OMVSGRP 55 Feb 15 08:07 afpxcrypto.dll -> /usr/lpp/IBM/PrintXform/V1R2/AFPxPDF/lib/afpxcrypto.dll
lrwxrwxrwx 1 BPXROOT OMVSGRP 44 Feb 15 08:07 afpxgum.dll -> /usr/lpp/IBM/PrintXform/V1R2/lib/afpxgum.dll
lrwxrwxrwx 1 BPXROOT OMVSGRP 52 Feb 15 08:07 afpxpcl.dll -> /usr/lpp/IBM/PrintXform/V1R2/AFPxPCL/lib/afpxpcl.dll
lrwxrwxrwx 1 BPXROOT OMVSGRP 53 Feb 15 08:07 afpxpcl.dll -> /usr/lpp/IBM/PrintXform/V1R2/AFPxPCL/lib/afpxpcl.dll
lrwxrwxrwx 1 BPXROOT OMVSGRP 52 Feb 15 08:07 afpxpdf.dll -> /usr/lpp/IBM/PrintXform/V1R2/AFPxPDF/lib/afpxpdf.dll
lrwxrwxrwx 1 BPXROOT OMVSGRP 53 Feb 15 08:07 afpxpdf.dll -> /usr/lpp/IBM/PrintXform/V1R2/AFPxPDF/lib/afpxpdf.dll
lrwxrwxrwx 1 BPXROOT OMVSGRP 54 Feb 15 08:07 afpxpdfua.dll -> /usr/lpp/IBM/PrintXform/V1R2/AFPxPDF/lib/afpxpdfua.dll
lrwxrwxrwx 1 BPXROOT OMVSGRP 50 Feb 15 08:07 afpxps.dll -> /usr/lpp/IBM/PrintXform/V1R2/AFFxPS/lib/afpxps.dll
lrwxrwxrwx 1 BPXROOT OMVSGRP 51 Feb 15 08:07 afpxpsd.dll -> /usr/lpp/IBM/PrintXform/V1R2/AFFxPS/lib/afpxpsd.dll
lrwxrwxrwx 1 BPXROOT OMVSGRP 44 Feb 15 08:07 afpxxml.dll -> /usr/lpp/IBM/PrintXform/V1R2/lib/afpxxml.dll
lrwxrwxrwx 1 BPXROOT OMVSGRP 47 Feb 15 08:07 afpxxmlb0.dll -> /usr/lpp/IBM/PrintXform/V1R2/lib/afpxxmlb0.dll
lrwxrwxrwx 1 BPXROOT OMVSGRP 47 Feb 15 08:07 aokpdfexit.dll -> /usr/lpp/IBM/PrintXform/V1R2/lib/aokpdfexit.dll
-rwxr-xr-x 2 BPXROOT OMVSGRP 5726208 Oct 26 12:54 aop.so
-rwxr-xr-x 2 BPXROOT OMVSGRP 32768 Oct 26 12:54 aopapi.dll
-rwxr-xr-x 2 BPXROOT OMVSGRP 528384 Oct 26 12:54 aopapi2.dll
-rwxr-xr-x 2 BPXROOT OMVSGRP 5558272 Sep 25 2019 aopcop.dll
-rwxr-xr-x 2 BPXROOT OMVSGRP 1204224 Oct 26 12:54 aopdb.so
-rwxr-xr-x 2 BPXROOT OMVSGRP 1978368 Oct 26 12:54 aopeapi.dll
-rwxr-xr-x 2 BPXROOT OMVSGRP 20480 Oct 26 12:54 aopfiltr.so
-rwxr-xr-x 2 BPXROOT OMVSGRP 122880 Oct 26 12:54 aopicpc.so
-rwxr-xr-x 2 BPXROOT OMVSGRP 3313664 Oct 26 12:54 aopicc2.so
-rwxr-xr-x 2 BPXROOT OMVSGRP 241664 Oct 26 12:54 aopform.dll
```

```
[MINGO @ TES3] /24AT3/usr/lpp/Printsrv/bin
=>ls -l
total 130032
drwxr-xr-x 2 BPXROOT OMVSGRP 8192 Sep 25 2019 IBM
lrwxrwxrwx 1 BPXROOT OMVSGRP 48 Feb 15 08:07 afpxpcl -> /usr/lpp/IBM/PrintXform/V1R2/AFPxPCL/bin/afpxpcl
lrwxrwxrwx 1 BPXROOT OMVSGRP 49 Feb 15 08:07 afpxpcld -> /usr/lpp/IBM/PrintXform/V1R2/AFPxPCL/bin/afpxpcld
lrwxrwxrwx 1 BPXROOT OMVSGRP 48 Feb 15 08:07 afpxpdf -> /usr/lpp/IBM/PrintXform/V1R2/AFPxPDF/bin/afpxpdf
lrwxrwxrwx 1 BPXROOT OMVSGRP 49 Feb 15 08:07 afpxpdfd -> /usr/lpp/IBM/PrintXform/V1R2/AFPxPDF/bin/afpxpdfd
lrwxrwxrwx 1 BPXROOT OMVSGRP 46 Feb 15 08:07 afpxps -> /usr/lpp/IBM/PrintXform/V1R2/AFFxPS/bin/afpxps
lrwxrwxrwx 1 BPXROOT OMVSGRP 47 Feb 15 08:07 afpxpsd -> /usr/lpp/IBM/PrintXform/V1R2/AFFxPS/bin/afpxpsd
-rwxr-xr-x 2 BPXROOT OMVSGRP 2215936 Sep 25 2019 aopchkinv
-rwx----- 2 BPXROOT OMVSGRP 20480 Sep 25 2019 aopcstart
-rwx---S--- 2 BPXROOT AOPADMIN 3608576 Oct 26 12:54 aopd
-rwvx---x--- 2 BPXROOT AOPOPER 32768 Oct 26 12:54 aopdemon
-rwx---S--- 2 BPXROOT AOPADMIN 294912 Oct 26 12:54 aopippd
-rwxr-x--- 2 BPXROOT AOPADMIN 3231744 Oct 26 12:54 aoplogu
-rwx---S--- 2 BPXROOT AOPADMIN 307200 Oct 26 12:54 aoplpd
-rwxr-xr-x 2 BPXROOT OMVSGRP 5812224 Oct 26 12:54 aopmig
-rwxr-xr-x 2 BPXROOT OMVSGRP 2506752 Oct 26 12:54 aopmigns
-rwxr-xr-x 2 BPXROOT OMVSGRP 1880064 Oct 26 12:54 aopmigpw
-rwx---S--- 2 BPXROOT AOPADMIN 184320 Oct 26 12:54 aopnetd
-rwxr-xr-x 2 BPXROOT OMVSGRP 3698888 Oct 26 12:54 aopoms
```

Selecting Direct Sockets protocol vs LPR

In an IP PrintWay printer definition, you can select the transmission protocol that IP PrintWay uses to transmit output data sets from the JES spool to the printer, print server, or email destination. IP PrintWay supports these transmission protocols: LPR, direct sockets, IPP, VTAM®, and email.

Select the IP PrintWay direct sockets protocol if you want IP PrintWay to use the TCP/IP direct sockets printing protocol to transmit data sets directly to a designated port on a printer or print server. When you select this protocol, the printer or print server must support direct sockets printing.

Tips:

- Some printers support the direct sockets printing protocol as well as other protocols, such as the LPR protocol. For large data sets, the direct sockets printing protocol can provide better performance. However, you might want to select the LPR protocol to take advantage of the

formatting options that IP PrintWay can specify in the LPD control file. For example, printing a banner page.

- When you select the direct sockets protocol, IP PrintWay extended mode can record the number of printed pages in the SMF type 6 record and can restart printing after the last page that printed successfully. For information, see Tracking the number of printed pages (extended mode).

When you select the direct sockets protocol, IP PrintWay transmits data to the printer or print server at the IP address (or host name) and port number that you specify in the printer definition. A job submitter can override the IP address and port number specified in the printer definition by specifying the IP address in the DEST=IP: parameter and the port number in the PORTNO parameter on the OUTPUT JCL statement.

Because the job submitter can override the IP address and port number, you can create one printer definition for several printers that share the same attributes. To use this printer definition, the job submitter must specify the name of the printer definition, the IP address, and the port number on the OUTPUT JCL statement. If the job submitter does not specify the name of the printer definition on the OUTPUT JCL statement, IP PrintWay uses printer attributes specified in the default IP PrintWay printer definition described in Creating an IP PrintWay default printer definition. For more information about job submission, see z/OS Infoprint Server User's Guide.

IP PrintWay prints multiple copies by transmitting the data set to the printer the requested number of times. This is because the direct sockets printing protocol cannot print multiple copies of a single data set.

[AOP_PAPER, AOP_TRAYID and AOP_PJL environment variables](#)

AOP_PAPER

The name of the paper that is typically installed in each AFP input tray. The transform formats the PDF output for the paper in the first AFP input tray ID the document selects. The position (1 through 9) of each paper name represents the number of the AFP input tray. The 10th position represents any AFP input tray number greater than 9.

You can specify from 1 to 10 paper names. Separate each name with a space. If you specify fewer than 10 paper names, the transform uses the paper name in position 1.

Valid paper names and their sizes (width x height) are:

Paper Name

Size

a3

297 X 420 mm

a4

210 x 297 mm

a5

148.3 x 210 mm

b4

250 x 353 mm

b5

176 x 250 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)

ledger

11 x 17 in (279 x 432 mm)

legal

8.5 x 14.0 in (216 x 356 mm)

letter

8.5 x 11.0 in (216 x 279 mm)

monarch

3.875 x 7.5 in. (98.4 x 190.5 mm)

Default: AOP_PAPER -> "letter letter letter letter letter letter letter letter letter"

Example: The backslash in the example indicates that the text within the quotation marks continues on the next line:

```
environment={AOP_PAPER -> "letter legal letter letter \
letter letter letter letter letter letter"}
```

In this example, if the document to be transformed specifies:

- AFP input tray 1, the transform formats the output for letter size paper.
- AFP input tray 2, the transform formats the output for legal size paper.
- AFP input tray greater than 9, the transform formats the output for letter size paper.

AOP_TRAYID

A mapping of AFP input tray numbers to PCL tray IDs. The position (1 - 9) of each PCL tray ID corresponds to the AFP tray number. The 10th position corresponds to all AFP input tray numbers greater than 9.

Specify 1 to 10 PCL tray IDs, separating each number with a space. Number 0 (zero) indicates that an input tray is not installed in the printer. If the input document requests an input tray that is not installed, the transform writes an error message in the output file and uses the tray ID in position 1.

PCL tray IDs typically mean:

- PCL tray 1: Feed paper from a printer-specific tray.
- PCL tray 2: Feed paper from manual input.
- PCL tray 3: Feed envelope from manual input.
- PCL tray 4: Feed paper from lower tray.
- PCL tray 5: Feed paper from optional paper source.
- PCL tray 6: Feed envelope from optional envelope source.

The AFP to PCL transform codes the PCL tray ID specified in AOP_TRAYID in the PCL Paper Source command. For more information about PCL tray IDs, see the description of the Paper Source command in Hewlett Packard's PCL documentation. Because the implementation of paper tray IDs can vary from printer to printer, also see the documentation for your printer.

Rule: Specify a nonzero value in the first position. This rule applies because the transform uses tray 1 as the default tray.

Tips:

For more information about PCL tray IDs, see the description of the Paper Source command in Hewlett Packard's PCL documentation. Because the implementation of paper tray IDs can vary from printer to printer, also see the documentation for your printer. They may differ as follows:

1. PCL tray IDs do not match the tray numbers that are embossed on the actual printer trays.
2. If the usual PCL tray IDs do not work, specify different tray IDs in the AOP_TRAYID variable until the printer selects paper from the desired paper tray.

Default: AOP_TRAYID → "1 4 0 0 0 0 0 0 2"

Example: environment={AOP_TRAYID → "1 4 1 1 1 1 1 1 2"}

This example shows that if the document to be transformed specifies:

- AFP input tray 1, the transform uses PCL tray ID 1.
- AFP input tray 2, the transform uses PCL tray ID 4.
- AFP input tray 3 through 9, the transform uses PCL tray ID 1.
- AFP input tray greater than 9, the transform uses PCL tray ID 2.

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.

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 create PJL commands.

Default: AOP_PJL → no

Example: environment={AOP_PJL → yes}

Tip: If you use IP PrintWay extended mode, specify AOP_PJL → no because a PJL JOB command in the document can conflict with the PJL JOB command that IP PrintWay adds to the document to track the number of pages that print successfully.

IBM documentation references:

[IBM Print Transforms from AFP for Infoprint Server for z/OS](#)

[Program Directory for IBM Print Transform from AFP to PCL for Infoprint Server for z/OS V1R2M2](#)

[z/OS Infoprint Server Introduction](#)

[z/OS Infoprint Server User's Guide](#)

[z/OS Infoprint Server Customization](#)

[z/OS Infoprint Server Messages and Diagnosis](#)

[z/OS Infoprint Server Operation and Administration](#)