

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

Preface:

This paper is intended to help you to configure, customize and use the From AFP to PDF Transform V1.2.2. It also provides information that is 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 PDF Transform Flows	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 rule	11
Method 1) Use Infoprint Server ISPF panels (typically 12.8) to create a printer definition and job selection rule to transform output to PDF.	11
Method 2) Use the PIDU utility to create the printer definition and job selection rule	15
Sample PDF Transform Scenarios	17
Scenario 1) Transform Simple Line Data to PDF.....	17
Scenario 2) Use AOPBATCH to transform simple line data to PDF	18
Scenario 3) Use the UNIX shell command line to transform an AFP input file to a PDF with Bookmarks	19
Additional Reference Information	21
How to Restart the aopxfd Daemon:	21
Create the links to the transform code	21
Advanced Transform Scenarios) Use the UNIX Shell command line to transform an AFP input file to an encrypted PDF.....	23
Other IBM References:	24

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 to configure Infoprint Server as well as the transform product:

- 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 PDF for Infoprint Server for z/OS V1R2M2](#)' GI11-9844-03, that describes how to install the product.

AFP to PDF Transform Flows

The first figure shows how to send an AFP document as an encrypted PDF document in an email.

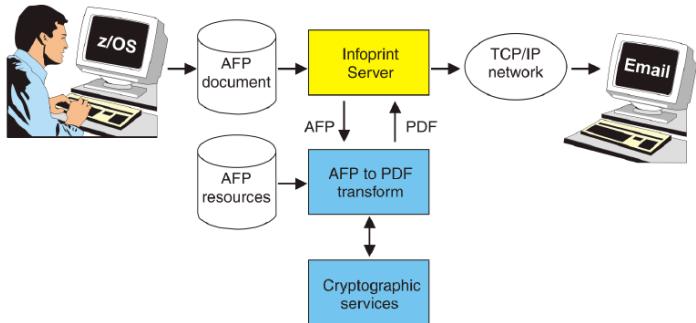
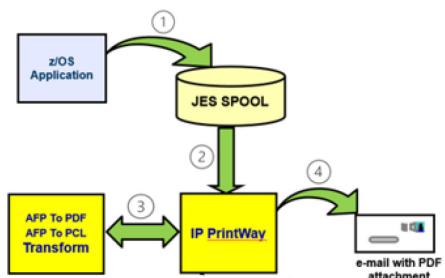


Figure 2. Sending AFP documents as encrypted PDF documents in emails

The second figure shows an application can submit a JCL job to produce output data on JES Spool (1), IP Printway can select that output from JES Spool (2), Printway can send the input to the transform to be transformed to PDF (3) and IP Printway can use sendmail to email the PDF output (4) to the email recipient.



Customer wants to print a utility bill form a z/OS application and send it to an end user as a PDF attachment

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 PDF TRANSFORM */
PRODUCT OWNER('IBM CORP')
NAME('PRINT TRANSFORMS')
ID(5655-TF1)
VERSION(*) RELEASE(*) MOD(*)
FEATURENAME('AFXPDPDF')
STATE(ENABLED)
/* */
```

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

BROWSE	SYS1.PARMLIB(IFAPRD00) - 01.20	Line 01
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('AFXPDPDF')	
	STATE(ENABLED)	
PRODUCT	OWNER('IBM CORP')	
	NAME('PRINT TRANSFORMS')	
	ID(5655-TF2)	
	VERSION(*) RELEASE(*) MOD(*)	
	FEATURENAME('AFPXPCP')	
PRODUCT	OWNER('IBM CORP')	
	NAME('PRINT TRANSFORMS')	
	ID(5655-TF3)	
	VERSION(*) RELEASE(*) MOD(*)	
	FEATURENAME('AFPXPS')	
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				
IFA111I 11.53.30 PROD DISPLAY 117				
S OWNER	NAME	FEATURE	VERSION	ID
E IBM CORP	PRINT TRANSFORMS AFPXPDF		* .* .*	5655-TF1

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/lib/afpxcrypto.dll
lrwxrwxrwx 1 BPXR00T OMVSGRP 44 Feb 15 08:07 afpxgum.dll -> /usr/lpp/IBM/PrintXform/V1R2/lib/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 afpxpcld.dll -> /usr/lpp/IBM/PrintXform/V1R2/AFPxPCL/lib/afpxpcld.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 afpxpdfua.dll -> /usr/lpp/IBM/PrintXform/V1R2/AFPxPDF/lib/afpxpdfua.dll
lrwxrwxrwx 1 BPXR00T OMVSGRP 50 Feb 15 08:07 afpxps.dll -> /usr/lpp/IBM/PrintXform/V1R2/AFPxFPS/lib/afpxps.dll
lrwxrwxrwx 1 BPXR00T OMVSGRP 51 Feb 15 08:07 afpxpsd.dll -> /usr/lpp/IBM/PrintXform/V1R2/AFPxFPS/lib/afpxpsd.dll
lrwxrwxrwx 1 BPXR00T OMVSGRP 44 Feb 15 08:07 afpxxml.dll -> /usr/lpp/IBM/PrintXform/V1R2/lib/afpxxml.dll
lrwxrwxrwx 1 BPXR00T OMVSGRP 47 Feb 15 08:07 afpxxmlb0.dll -> /usr/lpp/IBM/PrintXform/V1R2/lib/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      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 afpxpdfl -> /usr/lpp/IBM/PrintXform/V1R2/AFPxPDF/bin/afpxpdfl
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
-rwxr-xr-x  2 BPXROOT  OMVSGRP  2215936 Sep 25  2019 aopchkinv
-rwxr-xr-x  2 BPXROOT  OMVSGRP      20480 Sep 25  2019 aopcstart
-rwxr--S--- 2 BPXROOT  AOPADMIN 3608576 Oct 26 12:54 aopd
-rwsr-x--- 2 BPXROOT  AOPOPER   32768 Oct 26 12:54 aopdemon
-rwxr--S--- 2 BPXROOT  AOPADMIN 294912 Oct 26 12:54 aopippd
-rwxr--x--- 2 BPXROOT  AOPADMIN 3231744 Oct 26 12:54 aoplogu
-rwxr--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
-rwxr--S--- 2 BPXROOT  AOPADMIN 184320 Oct 26 12:54 aopmetd
-rwxr--x--- 2 BPXROOT  OMVSGRP 3698688 Oct 26 12:54 aopnoms
```

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 COUB1.AFM
-rw-r--r--  2 BPXROOT  OMVSGRP     114454 Jan  9  2020 COUB1.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 PDF 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 PDF Transform - Basic Transform Class  
-----  
transform afpxpdf  
    start-command = "afpxpdfd"  
    min-active = 0  
    max-active = 2  
    maximum-idle-time = 300 # 5 minutes  
    environment = {  
        AOP_ANNOTATIONS -> no  
        AOP_FONTLIB -> "sys1.sfntlib 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"  
        _BPX_JOBNAME -> AFPXPDFD  
    }  
;  
  
# PDF classes with AOP_INDEX->yes  
transform afpxpdf_indexyes  
    start-command = "afpxpdfd"  
    min-active = 0  
    max-active = 2  
    maximum-idle-time = 300 # 5 minutes  
    environment = {  
        AOP_INDEX-> yes  
        AOP_ANNOTATIONS -> no  
        AOP_FONTLIB -> "sys1.sfntlib 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"  
        _BPX_JOBNAME -> AFPXPDFD  
    }
```

```

;

# Class with encryption on
transform afpxpdf_en
  start-command = "afpxpdfd"
  min-active = 0
  max-active = 2
  maximum-idle-time = 300 # 5 minutes
  environment =
    AOP_ENCRYPT -> yes
    AOP_ANNOTATIONS -> no
    AOP_FONTLIB -> "sys1.sfntilib sys1.font300 sys1.fontlib sys1.fontlibb
sys1.sfonlplib"
    AOP_FORMDEFLIB -> "sys1.fdeflib common.formdef"
    AOP_PAGEDEFLIB -> "sys1.pdeflib common.pagedef"
    AOP_PAGESEGLIB -> "sys1.pseglib common.segment"
    AOP_OVERLAYLIB -> "sys1.overlay common.overlay"
    _BPX_JOBNAME -> AFPXPFD
}
;

;

```

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 PDF transform](#) for a list of environment variables to customize the AFP to PDF transform.

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
IPPIBM00	AOPIPPD	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 }
```

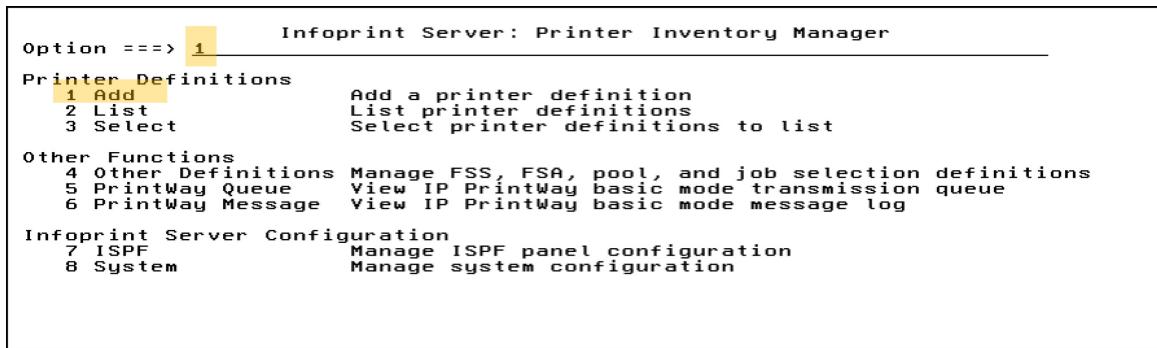
NOTE: If the InfoPrint Server ISPF, System (Option 8) Configuration panel, Operating mode parameter is set to z/OS 2.2 mode, this start-daemons parameter from aopd.conf is ignored.

Configure Printer Definition and job selection rule

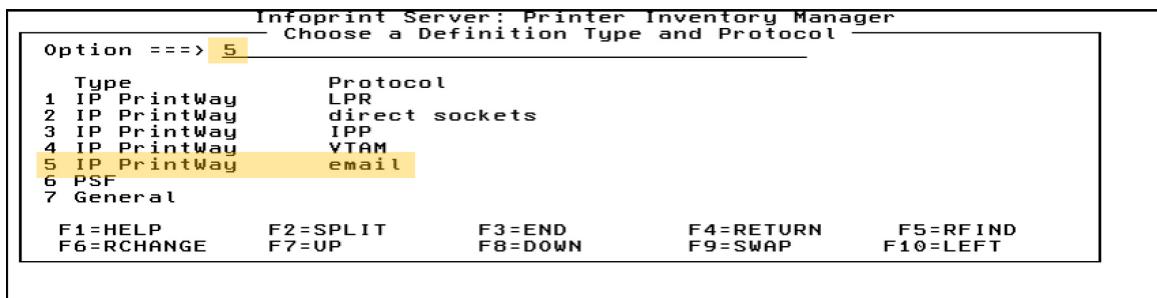
There are a couple ways to configure a printer definition, using IPSF panels or using the PIDU.

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

Select (1) Add a printer definition



The printer definition should have the following settings:



Give the printer definition a name and description

Allocation Panel

Choose a CLASS and DEST

Processing Panel

Set the dll name to process the input types. For the AFP to PDF transform, the dll name is afpxpdf.dll

<pre>Add Processing Command ==> _____ Printer definition name . PDFMAIL More: +</pre> <p>Document code page . . : IBM-1047</p> <p>Supported Data Formats and Associated Filters:</p> <pre>Data format: Filter: / Line data afpxpdf.dll (extend) / MO:DCA-P afpxpdf.dll (extend) / PostScript / Text aopfiltr.so (extend) / PCL / PDF / SAP / XML afpxpdf.dll (extend) / TIFF / JPEG / Other</pre> <p>_ Resubmit for filtering</p> <p>Transforms Error Handling:</p> <pre>Fail on error . . : - 1. No 2. Error 3. Warning Trailer error page: . - 1. No 2. Error 3. Warning</pre> <p>AFP to PDF Transform Encryption:</p> <pre>User identifier: _____ (extend) Owner identifier: _____ (extend) Encryption level 2 1. Low (40-bit key) 2. High (128-bit key) 3. AES (128-bit key) 4. AES (256-bit key)</pre> <p>Protected actions:</p> <pre>Restrict print: . . 2 1. Yes 2. No Restrict copy: . . 2 1. Yes 2. No Restrict update: . . 2 1. Yes 2. No</pre>
<input type="button" value="F1=HELP"/> <input type="button" value="F2=SPLIT"/> <input type="button" value="F3=END"/> <input type="button" value="F4=RETURN"/> <input type="button" value="F5=RFINID"/> <input type="button" value="F6=RCHANGE"/> <input type="button" value="F7=UP"/> <input type="button" value="F8=DOWN"/> <input type="button" value="F9=SWAP"/> <input type="button" value="F10=LEFT"/> <input type="button" value="F11=RIGHT"/> <input type="button" value="F12=RETRIEVE"/>

Protocol Panel

Set the email address to send the output to and the from name.

<pre>Add Email Protocol Command ==> _____</pre> <p>Printer definition name . PDFMAIL</p> <p>To addresses . . . tester@us.ibm.com (more)</p> <p>CC addresses . . . (more)</p> <p>BCC addresses . . . (more)</p> <p>From name tester</p> <p>Reply address . . .</p> <p>IP PrintWay Extended Mode:</p> <ul style="list-style-type: none"> - Embedded headers - Inline text and line data - Inline text attribute . . - Inline message . . . (extend)
<input type="button" value="F1=HELP"/> <input type="button" value="F2=SPLIT"/> <input type="button" value="F3=END"/> <input type="button" value="F4=RETURN"/> <input type="button" value="F5=RFINID"/> <input type="button" value="F6=RCHANGE"/> <input type="button" value="F7=UP"/> <input type="button" value="F8=DOWN"/> <input type="button" value="F9=SWAP"/> <input type="button" value="F10=LEFT"/> <input type="button" value="F11=RIGHT"/> <input type="button" value="F12=RETRIEVE"/>

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

Add	IP PrintWay Printer Definition	
Command ==> _____		
Printer definition name . PDFMAIL		
Description : PDF_email	(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)

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 F7=UP	F2=SPLIT F8=DOWN	F3=END F9=SWAP	F4=RETURN F10=LEFT	F5=RIND F11=RIGHT	F6=RCHANGE 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=R FIND          F6=R CHANGE
F7=UP            F8=DOWN           F9=SWAP           F10=LEFT           F11=RIGHT          F12=RETRIEVE
```

Add Job Selection Rule
Command ==> _____

Rule name . MAIL _____
Description . _____ (extend)
Operator security profile
. . .
DEST . . : MAIL _____
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 ‘tester@us.ibm.com’ to a valid

email address and ‘tester’ to the from username. 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.

NOTE: There is some configuration necessary to send output to email destinations. For more information, refer to [Infoprint Server Customization, Chapter 7, Customizing the sendmail bridge.](#)

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

pidu class example:

```
#-----
# Printer PDFMAIL to email tester@us.ibm.com
#-----
create printer PDFMAIL
    printer-codepage = IBM-1047
    print-page-header = no
    filters = {
        line -> afpxpdf.dll
        modca -> afpxpdf.dll
        text -> aopfiltr.so
        xml -> afpxpdf.dll
    }
    printer-type = ip-printway
    protocol-type = email
    destination = MAIL
    output-class = J
    mail-to-addresses = {
        "tester@us.ibm.com"
    }
    mail-from-name = tester
    dcf-routing = yes
    description = "PDF email"
;

create dcf-routing-key "MAIL      J"
    printer = PDFMAIL
;

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

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

```
=>pidu pidu.txt
AOP062I printer PDFMAIL was created.
AOP062I dcf-routing-key MAIL J was created.
AOP062I job-selection-rule MAIL 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 PDF Transform Scenarios

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

Scenario 1) Transform Simple Line Data to PDF

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 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=MAIL 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 PDF and email (“print”) the output PDF as an attachment to the specified email address.

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 PDF and email the result

Submit this JCL and it should create PDF output in data set XFORM.AOPBTEST.ONEPAGE.PDF. As you can see, it uses XFORM.AFPDOC.FB80(ONEPAGE) as input, the same simple line data example in Scenario 1. The second JCL step will route the PDF output to printer PDFMAIL using CLASS=J and DEST=MAIL and it will be emailed to the user specified in printer configuration. Using REGION=0M grants the job access to all available storage, dependent on any site-specific limits.

NOTE: There is some configuration necessary to send output to email destinations. For more information, refer to [InfoPrint Server Customization, Chapter 7, Customizing the sendmail bridge](#).

```
***** 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.PDF,
// 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 PDF output
//*****
//S1 EXEC PGM=AOPBATCH,
// PARM='/afpxpdf -j attr=/DD:ATTRS -o //DD:OUTPUT //DD:IN'
//*
//IN DD DSN=XFORM.AFPDOC.FB80(ONEPAGE),DISP=SHR
//OUTPUT DD DSN=XFORM.AOPBTEST.ONEPAGE.PDF,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=MAIL,
//                  PRTATTRS='document-format=pdf'
/*
//SYSUT1    DD  DSN=XFORM.AOPBTEST.ONEPAGE.PDF,DISP=SHR
//SYSUT2    DD  SYSOUT=(,),OUTPUT=(*.OUT1)
//SYSPRINT DD  SYSOUT=*
//SYSIN     DD  DUMMY
/*
***** Bottom of Data *****

```

You could then go to a UNIX shell command line and enter the following command to copy the output data set containing PDF data to onepage.pdf.

```
cp "://"XFORM.AOPBTEST.ONEPAGE.PDF"" <unix directory>/onepage.pdf
```

Or you can FTP the result to a destination where you can view PDF documents

Scenario 3) Use the UNIX shell command line to transform an AFP input file to a PDF with Bookmarks

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 afpxpdf command can be found.
- 2) The AFP binary input file baxter.aff exists in a location in your UNIX environment. The baxter.aff 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 BaxterIndex.pdf. Replace <unix directory> with the UNIX directory where you placed baxter.aff.

```
afpxpdf -o BaxterIndex.pdf -c indexyes <unix directory>/baxter.aff
```

Since the indexyes transform class is using AOP_INDEX-> yes and the baxter.aff input file has Tag Logical Elements (TLE) structured fields, the resulting PDF should have Bookmarks in the left pane...

BBB_PDFUONLYSMALLIndex.pdf - Adobe Acrobat Pro (32-bit)

File Edit View E-Sign Window Help

Home Tools Document Bookmarks

Share

NAME
ACCOUNT
CRD_DATE
BALANCE

BAXTER BAY BANK
More Bank for the Buck

P 1
A
G OF
E 1

STATEMENT DATE	
01/31/97	
CREDIT LIMIT	
16000.00	
CREDIT AVAILABLE	AMOUNT OVER LIMIT
14816	0
ACCOUNT NUMBER	
000-000-000	

MAKE CHECKS PAYABLE TO: Big Bucks Card

DIRECT PAYMENT OR INQUIRIES TO: Big Bucks Card Center
CUSTOMER SERVICES
P.O. BOX 100/BOULDER, CO 99999
PHONE (123) 123-4567

*Smith Cyclery Co
113 W. Maumee St
Adrian, MI 49221*

AMOUNT PAID _____

PLEASE DETACH AND RETURN THIS TOP PORTION WITH PAYMENT

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 afpxpcld.dll -> /usr/lpp/IBM/PrintXform/V1R2/AFPxPCL/lib/afpxpcld.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 afpxpdfd.dll -> /usr/lpp/IBM/PrintXform/V1R2/AFPxPDF/lib/afpxpdfd.dll
lrwxrwxrwx 1 BPXROOT OMVSGRP 54 Feb 15 08:07 afpxdfua.dll -> /usr/lpp/IBM/PrintXform/V1R2/AFPxPDF/lib/afpxdfua.dll
lrwxrwxrwx 1 BPXROOT OMVSGRP 50 Feb 15 08:07 afpxps.dll -> /usr/lpp/IBM/PrintXform/V1R2/AFPxPS/lib/afpxps.dll
lrwxrwxrwx 1 BPXROOT OMVSGRP 51 Feb 15 08:07 afpxpsd.dll -> /usr/lpp/IBM/PrintXform/V1R2/AFPxPS/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 aopipc.so
-rwxr-xr-x 2 BPXROOT OMVSGRP 3313664 Oct 26 12:54 aopipc2.so
-rwxr-xr-x 2 BPXROOT 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 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/AFPxPS/bin/afpxps
lrwxrwxrwx 1 BPXROOT OMVSGRP 47 Feb 15 08:07 afpxpsd -> /usr/lpp/IBM/PrintXform/V1R2/AFPxPS/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
-rwxr-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 aopmetd
-rwvxr-vv-v 2 BPXROOT OMVSGRP 3698688 Oct 26 12:54 aoproma
```

Advanced Transform Scenarios) Use the UNIX Shell command line to transform an AFP input file to an encrypted PDF

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 afpxpdf command is found.
- 2) The AFP binary input file baxter.afp exists in a location in your UNIX environment. The baxter.afp file was provided in the zip file bundled with this document. FTP (in binary) to a UNIX directory location.
- 3) A sample Password exit is provided in /usr/lpp/IBM/PrintXform/V1R2/lib/aokpdfexit.dll. This sample exit accepts the name of the password database as an optional argument (if none is specified, the sample exit uses /etc/Printsrv/aokpdfexit.db). A sample password database (aokpdfexit.db) was provided in the zip file bundled with this document. FTP (in ASCII) to /etc/Printsrv/aokpdfexit.db and restart the aopxfd daemon. This database has the 'Nurse-Lee' and 'Dr-Smith' entries needed for the encryption.

Run the following command from a UNIX shell. It should create an output file called 128BitEncrypted.pdf that has 128-bit encryption, and the PDF can be opened by using password 'Nurse-Lee' or 'Dr-Smith'. Replace <unix directory> with the UNIX directory where you placed baxter.afp.

```
afpxpdf -o 128BitEncrypted.pdf -c en -j " pdf-owner-identifier='Nurse-Lee' pdf-user-identifier='Dr-Smith' pdf-protect={copy print update}" <unix directory>/baxter.afp
```

Run the following command from a UNIX shell. It should create an output file called AES256Encrypted.pdf that has AES256 encryption, and the PDF can be opened by using password 'Nurse-Lee' or 'Dr-Smith'. Replace <unix directory> with the UNIX directory where you placed baxter.afp.

Note: For AES256 encryption, when ICSF is configured on the system, the transform will use it to perform a cryptographically superior random number generation. If ICSF is not configured on the system, message AOK00426W will be issued, and AES encryption will continue using a less secure alternative.

```
afpxpdf -o AES256Encrypted.pdf -c en -j " pdf-encryption-level=aes256 pdf-owner-identifier='Nurse-Lee' pdf-user-identifier='Dr-Smith' pdf-protect={copy print update}" <unix directory>/baxter.afp
```

Other IBM References:

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

[Program Directory for IBM Print Transform from AFP to PDF 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](#)