

Configuring and Exploiting the New Infoprint Server FromAFP Transforms

Elizabeth Smith

elizabeth.smith@infoprint.com

Ricoh Production Print Solutions

Share 2012, Anaheim

August 7, 2012

Session 11543



New product: IBM Print Transforms from AFP V1.1



- IBM Print Transforms from AFP V1.1 allow you to transform AFP to the following data streams:
 - HP Printer Control Language (PCL)
(5655-TF2) Print Transform **from AFP to HP PCL** Print Transform for Infoprint Server for z/OS, V1.1.0
 - Adobe Portable Document Format (PDF)
(5655-TF1) Print Transform **from AFP to Adobe PDF** for Infoprint Server for z/OS, V1.1.0
 - Adobe PostScript
(5655-TF3) Print Transform **from AFP to Adobe PostScript** for Infoprint Server for z/OS, V1.1.0
- Prerequisite: (5694-A01) **Infoprint Server for z/OS**

What is Infoprint Server?

- **Printer Inventory**

- Provides a single set of printer definitions that all the components of Infoprint Server use.
- Contains printer customization information that Print Services Facility (PSF) uses.

- **Print Interface**

- Accepts print requests from z/OS UNIX System Services and from remote systems in your network, and allocates output data sets on the JES spool for printing on local or remote printers.

What is Infoprint Server?

- **NetSpool™**
 - Intercepts print output from VTAM applications and allocates output data sets on the JES2 or JES3 spool for printing on local or remote printers.
- **IP PrintWay™**
 - Transmits output data sets from the JES2 or JES3 spool to remote printers in a TCP/IP network or SNA network.
- **Infoprint Central**
 - A Web-based application that lets help desk operators and other authorized users or job submitters work with print jobs (output data sets) on the JES spool, printers controlled by IP PrintWay extended mode or PSF, and NetSpool logical units.
 - Lets operators see system status and printer definitions.

What is Infoprint Server?

- **Windows Client**

- Infoprint Port Monitor for Windows, which transmits documents and job attributes to Infoprint Server from Windows applications.

- **Transform Interface**

- Communicates with transform products that IBM and Ricoh provide. Transforms convert data from one format to another. For example, from PDF to AFP format, or from AFP to PCL format.

New functionality in IBM Print Transforms from AFP



- The new **IBM Print Transforms from AFP V1.1** products for Infoprint Server for z/OS replace the previous IBM Infoprint Transforms from AFP for z/OS V2.1
 - Full support for both 240-pel and 300-pel raster fonts.
 - Enhanced and simplified font mapping
 - Printing for global markets
 - Enhanced bar code support
 - Improved PDF processing
 - Better support for AFP PGD descriptor, which is helpful for jobs with mixed paper sizes
 - Enhanced error processing and message-handling
 - Better base for future functionality

Agenda



- History, PTFs and prereqs
- Details on new functions and enhancements
- Differences and Migration
- Tips and tricks

A short history, PTFs and prereqs

Transition to new transform product



- March 2011 : new **IBM Print Transforms from AFP V1.1** product (afp~~x~~pdf, afp~~x~~pcl, afp~~x~~ps) GA
- Starting in z/OS **V1R12**, V1.1 afpx* transforms are the only IBM from AFP Transform products that can be ordered
- Version 2.1 product (afp2pdf, afp2pcl, afp2ps) has been discontinued
 - Can no longer be ordered, but can be used if already installed
 - Still in service through September 2013

Software requirements

- The transforms run on all supported releases of z/OS.
- Additional software requirements:
 - PTF for **APAR OA38585** for Print Transforms from AFP for Infoprint Server for z/OS
 - PTF for APAR **OA37237** for PRMODE parameters on the OUTPUT JCL statement
 - If you run z/OS Infoprint Server V1R13 or earlier, PTF for APAR **OA36444** and PTF for APAR **OA38779**
 - If you run z/OS Infoprint Server V1R11 or earlier, PTF for APAR **OA32228** for enhanced message support
 - If you transform data from XML, IBM XML Toolkit for z/OS, C++ edition, V1.10 (5655-J51)
 - An AFP page definition, form definition, and font for formatting transform error messages
 - AFP resources that are not included inline in AFP documents

OA35704

- **UA63618** (1.10), **UA63619** (1.11), **UA63620** (1.12), **UA63621** (1.13)
- Related Infoprint Server APAR **OA36444**
 - ISPF help panel changes, updates to aopxfd.conf and Infoprint Central
- Continues support for remaining V2.1 functions:
 - TIFF and GIF object containers
 - All rotations and print directions
 - Color printed output
 - JCL parameters: OFFSETXB, OFFSETYB, OFFSETXF, OFFSETYF, OVERLAYB, OVERLAYF, PRMODE
 - Record-format line data
 - SOSI commands
 - XML line data

OA35704 continued support of V2.1 functions

- Existing transform environment variables:
 - AOP_ANNOTATIONS (AFP to PDF only)
 - AOP_ASSUME_RIDIC_UNPAD
 - AOP_COLOR
 - AOP_CUTSHEET (AFP to PCL and to PostScript only)
 - AOP_FAIL_ON_IMAGE_ERROR
 - AOP_LINEARIZE (AFP to PDF only)
 - AOP_LINKS (AFP to PDF only)
 - AOP_OFFSET_JOGGING (AFP to PCL and to PostScript only)
 - AOP_OUTLINES (AFP to PDF only)

OA35704 new function

- **New function** unique to V1.1
 - Additional bar codes – QR, USPS Four-State - Intelligent Mail Barcode (IMB)
 - True Type, World Type, and Open Type fonts in a single directory for UTF8 and UTF16 character input
 - A variety of error-handling enhancements
 - Additional transform environment variables:
 - AOP_FAIL_ON_ERROR
 - AOP_FONTMAP_TABLE
 - AOP_FONTPATH
 - AOP_TRAILER_ERROR_PAGE

OA38585

- **UA64998** (1.10), **UA64999** (1.11), **UA65000** (1.12), **UA65001** (1.13)
- Related Infoprint Server APAR **OA38779**
- Continues support for the remaining V2.1 functions:
 - Enhanced N_UP processing
 - Processing stand-alone page segments or overlays for complex processing requirements
 - Many miscellaneous fixes including:
 - Proper character positioning and rotation
 - Support for X'89' triplet on LND/RCD/XMD structured fields
 - Support for page definition PRINTLINE OVERLAY parameter
 - N_UP processing with duplex pages
 - GOCA ellipse processing

OA38585, new function

- **New function** unique to V1.1
 - Ability to process a PDF object container of up to one page, included in the input document, for all three transforms
 - Support for TrueType, World Type and Open Type (UTF8, UTF16, and EBCDIC), with option of multiple directories in a search path
 - Control over embedding outline fonts in PostScript output
 - Font mapping now processes wildcards as part of the raster font character set name
 - Support for record mode processing for Record length of 1 to 250

Latest level, OA39657

- **UA65899 (1.10), UA65900 (1.11), UA65901 (1.12), UA65902 (1.13)**
- Provides several fixes for XML processing
 - Correct data positioning on page
 - Correctly process conditional processing information in the page definition with XMD/XML.
 - Properly handle font elements so that characters are rendered correctly.
 - Processing for header and footer function changed to generate correct page numbers for the output document.
 - Checks have been added to process an empty input file and generate an empty output file with a successful return code.

How to check current transforms APAR level

- The **amblist** command can be entered from the z/OS UNIX command line, and tells you the current level of support for a transform.

```
>echo " LISTIDR" | amblist  
/usr/lpp/Printsrv/lib/afpxpdfd.dll | grep -i  
Build_Id
```

Output similar to this:

```
10/16/2011 OA35704 BUILD_ID=3470 20111016
```

Details on new functions and enhancements

Double Byte support

- Now full double-byte character set (DBCS) font support:
 - 240 and 300-pel raster fonts (fixed and relative metrics)
 - AFP outline fonts
 - TrueType, OpenType, and WorldType fonts
 - Shift-out/shift-in: SOSI1, SOSI2, SOSI3, and SOSI4 options

Better font support

- Greek and Cyrillic fonts
- 240-pel fonts now supported for all transforms, which eliminates the need to use AOXCF30 to map them to 300-pel.
- The AFPXPCL transform includes all raster fonts in the output data stream to make sure they are available, and automatically rasterizes outline, OpenType and True Type fonts to 300-dpi raster fonts and includes them in the output data stream. You no longer have to set **AOP_FONTMAP=yes** to enable this.
- AFPXPCL can also now handle fonts created on the fly by font generator programs using names that have no similarity to the standard font naming convention.

Embedding outline fonts in PostScript output for improved output fidelity

- New **AOP_EMBED_OUTLINES** environment variable allows the AFPXPS transform to decide whether or not to include outline fonts in the PostScript output.
- Default=**yes**, which provides better output fidelity. If set to **no**, the transform will convert/rasterize outline fonts to equivalent 300-dpi raster fonts and include them in the PostScript data stream.
- Specified in aopxfd.conf for the PostScript transform entry:
`environment={AOP_EMBED_OUTLINES -> no}`
- Embedding outlines creates larger files: each included typeface increases the size of the PostScript output file by approximately 175 KB.

How to use WorldType, TrueType, and OpenType fonts

- Install IBM's WorldType Fonts for AFP Print Servers optional feature of Infoprint Fonts for Multiplatforms (Program Number 5648-E77) or a TrueType, OpenType, and WorldType font package from a company such as Agfa, Monotype, or Linotype.
- Use a resource installer program (such as the Font Installer for AFP Systems or the InfoPrint AFP Resource) to install and build Resource Access Tables (RATs) for the fonts.

How to use WorldType, TrueType, and OpenType fonts, continued

- Install the fonts in path directories and set read permissions, and identify them in JCL or the transform configuration file:
 - **USERPATH** parameter on the OUTPUT JCL statement.
`// USERPATH=(/usr/fonts/ttfonts)`
 - **AOP_FONTPATH** environment variable in **aopxfd.conf**
`environment={AOP_FONTPATH -> "/usr/lpp/fonts/myfonts
/jdoe/fonts/truetype"}`
- You should usually specify the same AFP resource directories in the transform configuration file as you currently specify in your PSF startup procedures.

Enhanced dynamic font mapping table support

- The font mapping table now resides in a Unix file, and is more straightforward to use.
- Each transform class can now choose to use a unique font mapping table.
- Automatic conversion of outline fonts to raster fonts is performed for PCL and PS, and can be controlled for AFPXPS with the **AOP_EMBED_OUTLINES** environment variable.
- Sample in </usr/lpp/Printsrv/samples/aokfontmap.samp>.

Wildcards in font mapping

- Raster font character set name can now be specified as a wildcard (*) for font mapping, which allows you to use a single statement to map a family of raster fonts to a corresponding family of outline fonts.
- Eliminates the need to map every raster font by pitch size to the corresponding outline font.
- Example from /usr/lpp/Printsrv/samples/aokfontmap.samp font mapping table:
; Map Specials Bookmaster 12 point normal raster to
; Bookmaster outline font.
mapfont RFONT=C0B20CB0, OFONT=CZB20C;
mapfont RFONT=C0B20C* OFONT=CZB20C;

Improved handling for jobs with mixed paper sizes.

- **AOP_PAPER** environment variable describes the name of the paper that is typically installed in each AFP input tray. The transform uses this variable to format its output.
- If **AOP_PAPER** is set to the new value **AUTO**, the transform uses the page size in the Page Descriptor (PGD) structured field.
Note: If the to PCL transform is being used, the page size specified in the PGD must be equivalent to a standard PCL page size. If not, it uses letter size because PCL interpreters do not recognize non-standard paper sizes.
- Specified in the transform configuration file, **aopxfd.conf**:
AOP_PAPER ->
"auto legal letter letter auto auto letter letter
letter letter"

Issues with error-handling in previous transforms

- Previous transforms error-handling was not ideal. Transforms would either:
 - Ignore the problem and continue
 - Stop processing with error message appended to output document
- Return code of zero even for data stream errors
- No messages issued to the Infoprint Server message log and no ability to suppress trailer error message page on output.
- No convenient way for system programmer to monitor the message log or implement return code checking to prevent printing or distribution of bad output, except for certain specific IOCA decoding errors.

To fix these problems, error-handling enhancements have been added

- Now, with error-handling enhancements added in OA35704:
 - **Transform messages are issued to the Infoprint Server message log**, in addition to transform stderr file and (optionally) output data.
 - Ability to request whether or not the transform signals a **non-zero return code for data stream warnings and errors**.
 - Ability to tell the transform **whether or not to append a trailer message page** to output.
 - Ability to request **more granular MVS-style (0/4/8) return codes**, rather than UNIX-style (0/1)

New trailer error page option

- **Indicates when to attach trailer page of messages to output**

- Environment variable in aopxfd.conf transform entry

`AOP_TRAILER_ERROR_PAGE = no | error | warning`

- Job attribute

`trailer-transform-error-page = no | error | warning`

- Printer definition field option

`Trailer error page. . 1. No 2. Error 3. Warning`

no

No trailer page is attached.

error

Attach trailer page of error messages.

Warning (default)

Attach trailer page of warning and error messages

New fail on error option

- **Customize the error level that halts the transform.**

- Environment variable in **aopxfd.conf** transform entry
`AOP_FAIL_ON_ERROR = no | error | warning`
- Job attribute
`fail-on-transform-error = no | error | warning`
- Printer definition field option
`Fail on error 1. No 2. Error 3. Warning`

no

Transform attempts to continue processing when errors are encountered in the input AFP

error (default)

The transform stops processing when errors are encountered in the input (but continues processing on warnings)

warning

The transform stops processing when warnings or errors are encountered in the input

More granularity added to the return codes

- New environment variable to request more granular return codes with **AOP_MVS_RETURN_CODES**.

AOP_MVS_RETURN_CODES

no (default) UNIX exit values (0/1)

yes MVS return codes (0/4/8)

- **Not specified in aopxfd.conf.** Can specify:
 - Before the command on the z/OS UNIX command line
 - \$HOME/.profile or \$HOME/.tcshrc
 - In the STDENV data set for AOPPRINT or AOPBATCH

```
AOP_MVS_RETURN_CODES=yes afpxpcl -o output.pcl input.afp
```

Suggested settings for new error-related options

- Two phases when producing a document:
 - Creation and testing of a new document, where you want to quickly find and correct any problems in the input data.
fail-on-transform-error=no
trailer-transform-error-page=warning
 - Putting the document into production, when you don't want the document to be produced if there are errors in the input, and don't want error or warning messages appended to output:
fail-on-transform-error=warning or error
trailer-transform-error-page=no

AOPBATCH example

```
//TRANSFRM   EXEC PGM=AOPBATCH,
// PARM='/afpxpcl -o //DD:OUTPUT -jattr=//DD:ATTRS //DD:INPUT'
//OUTPUT     DD DSN=PRODUCTN.OUTPUT.PCL,DISP=(NEW,CATLG,DELETE),
//   DCB=(RECFM=VB,LRECL=1024),SPACE=(CYL,(1,1))
//INPUT      DD DSN=PRODUCTN.INPUT.DOCUMENT,DISP=SHR
//ATTRS      DD  *
form-def=flfdef
page-def=plpdef
fail-on-transform-error=error
trailer-transform-error-page=no
/*
//STDOUT     DD   SYSOUT=*
//STDERR     DD   SYSOUT=*
//STDENV DD  *

AOP_MVS_RETURN_CODES=YES
PATH=/usr/lpp/Printsrv/bin:/bin:/usr/bin
LIBPATH=/usr/lpp/Printsrv/lib:/lib:/usr/lib
NLSPATH=/usr/lpp/Printsrv/En_US/%N:/usr/lib/nls/msg/En_US/%N
AOPCONF=/etc/Printsrv/aopd.conf
/*
//*
//PRINTIT   EXEC PGM=IEBGENER,REGION=212K,COND=(0,LT,TRANSFRM)
```

Automatic transform trace facility

- The transforms automatically generate a flight recorder-type entry for every transform that is performed, and place it into transform directory stderr file:
 - Transform start and stop time
 - Job and file name
 - Printer definition name being used
 - AOKxxxxx info, warning and error messages from the transform
 - CPU time in seconds for the transform
 - I/O service units consumed by the transform
- In addition to helping service and system programmers locate transform history and isolate problems, also allows you to monitor and observe transform activity and progress.

Automatic transform trace facility

- How to invoke, if you want optional information about sysout-dataset-name, sysout-job-id, sysout-job-name:

```
afpxpdf -o test.pdf -j "sysout-dataset-name='DATASET1'
sysout-job-id='JOBID001' sysout-job-name='POSE732' "
"///' PSFMVS.FVT.R410.LINEDATA.FBA(MULTIPAG) ' "
```

- Output:

```
Starting Transform @ 2012/06/25 16:28:07.986160
```

```
Performing Transform for Job POSE732 - JOBID001
```

```
for Sysout Dataset DATASET1
```

```
Transform Terminated @ 2012/06/25 16:28:09.993369,
used 0.521529 cpu seconds, 2503 I/O service units.
```

Migration

Transform name change

- Transforms naming scheme has changed, from AFP2P* to **AFPXP***
 - Transform command: afp2pdf -> afpxpdf
 - Transform DLL: afp2ps.dll -> afxps.dll
 - Transform classes: afp2pcl_us -> afxpcl_us
 - Transform daemon names: afp2pdfd -> afxpdfd
- Name change needs to be reflected in:
 - aopxfd.conf
 - Infoprint Server printer definitions that reference the transforms in their Processing section
`/ MO:DCA-P afxpdf.dll -c encrypt %filter-options (extend)`
 - AOPBATCH and AOPPRINT JCL
 - Unix shell scripts

Easing migration of printer definitions

- To save time for installations with many printer definitions, new shell script tool `/usr/lpp/Printsrv/samples/aokmodfilter.sh` can be used to modify their filter section.
- **Steps:**
 - 1. Run `aokmodfilter.sh`

```
>aokmodfilter.sh  
Migrating afp2 filters to afpx  
AOP065I 137 definitions were exported to  
/tmp/inventory.dump.  
modify statements written to /tmp/filter.defs  
removing file /tmp/inventory.dump
```

Easing migration of printer definitions

- 2. Examine output in **/tmp/filter.defs** to verify that the modify commands are correct, and remove any you do not want changed.

```
modify printer DUP0010_AFP2PCL filters={line ->  
"afpxpcl.dll -c us %filter-options" modca ->  
"afpxpcl.dll -c us %filter-options" text -> aopfiltr.so  
};
```

- 3. Use the **pidu** tool to modify the original printer definitions.

```
>pidu < /tmp/filter.defs  
AOP063I printer DUP0010_AFP2PCL was replaced.
```

Using symbolic links to manage name change

- The transform command names are used by the AOPBATCH and AOPPRINT JCL and any Unix shell scripts that execute the transform, so these also need to change to point to the new transforms.
- Instead of editing each JCL file and Unix script that references the transform commands, you can use symbolic links.

Using symbolic links to manage name change

- To create a symbolic link from **afp2pdf** to **afpxpdf**, using the directory **/usr/bin**:
 - Create a symbolic link from **afp2pdf** to **/usr/lpp/Printsrv/bin/afpxpdf** in **/usr/bin**.

```
>ln -s /usr/lpp/Printsrv/bin/afpxpdf /usr/bin/afp2pdf
```
 - Modify the PATH environment variable:
 - For JCL, in the **aopstart EXEC**

```
>n=n+1; env.n='PATH=/usr/bin:' install_path' /bin:/bin'
```
 - For Unix scripts, in the shell running the script.

```
>export PATH=./usr/bin:$PATH
```
 - Stop and restart the Transform Manager daemon (**aopxfd**).

Using both transforms at once

- The new product was designed to coexist with the old product, to ease transition.
- If you do not use symbolic links to point to the new transforms, you can use both transform products at the same time because the names of the transform commands, filters, and daemons differ between versions.
- The Infoprint Server transform configuration file (**aopxfd.conf**) can contain entries for both Print Transforms from AFP V1.1 and Infoprint Transforms from AFP V2.1.

Edge-to-edge printing differences

- AOP_PAPER environment variable is still supported, but paper names ending in 'ee' (**letteree** and **a4ee**) are not. If you specify these values, the transforms map:
 - **letteree -> letter**
 - **a4ee -> a4.**
- For AFP2PCL edge-to-edge printing, specify the new **AOP_EDGETOEDGE** environment variable:
 - yes The AOP_PAPER values will be sent to the printer and adjusted for edge-to-edge printing.
 - no No adjustment is made. (**default**)

Edge-to-edge printing differences

- `environment={AOP_EDGE TOEDGE -> yes}`
- **Note that AOP_EDGE TOEDGE applies to all paper sizes in all input trays for a particular transform class.**
For edge-to-edge printing for only certain paper sizes or trays, you can create separate transform classes.

PDF encryption with passwords

- If you encrypt the transformed PDF documents, you may need to make changes to the **AOP_PASSWORD_EXIT**
- The exit is the same, but file names for exit and password database have changed.
- **Password exit:** aoppdfexit.dll -> aokpdfexit.dll
- **Password database:** aoppdfexit.db -> aokpdfexit.db
- **AOP_PASSWORD_EXIT** default:
/usr/lpp/Printsrv/lib/aoppdfexit.dll -> /usr/lpp/Printsrv/lib/aokpdfexit.dll
- **NOTE:** No additional products are required to use encryption for V1.1 AFPXPDF. V2.1 AFP2PDF requires that OCSF be installed.

DATAACK and print-error-reporting functionality replaced

- The DATAACK JCL parameter and **print-error-reporting** attribute are no longer supported, but the **AOP_TRAILER_ERROR_PAGE** environment variable and **trailer-transform-error-page** job attribute provide a similar set of options for requesting whether or not to attach error and warning messages to the output.
- Specify the PRTATTRS JCL parameter in the OUTPUT JCL statement to specify the **trailer-transform-error-page** job attribute as a replacement:
 - To not attach a trailer page (corresponds to DATAACK=BLOCK):
`//OUTDS OUTPUT PRTATTRS='trailer-transform-error-page=no'`
 - To attach a trailer page (corresponds to DATAACK=UNBLOCK, DATAACK=BLKCHAR, and DATAACK=BLKPOS):
`//OUTDS OUTPUT PRTATTRS='trailer-transform-error-page=warning'`

Font mapping

- Note that the new font mapping file, **/usr/lpp/Printsrv/samples/aokfontmap.samp**, includes all of the mappings in SYS1.SAMPLIB(AOXFONTS).
- If you have customized font mappings specified in the old SYS1.SAMPLIB mapping tables, move them to the new font mapping file, and add the **AOP_FONTMAP_TABLE** environment in transform entries that should use the mapping file.

```
environment={AOP_FONTMAP_TABLE ->  
"/usr/lpp/Printsrv/samples/aokfontmap.samp" }
```

Font mapping

- **AOP_FONTMAP** is no longer used, but will be ignored if left in aopxfd.conf transform entries.
- When viewing the output PDF document, if the fidelity of the font characters is not as sharp as it was when using the previous transform product, this might be because the transform is inserting raster font characters rather than outline fonts into the output PDF. You can remedy this by including a font-mapping table.

Creating a font-mapping table

- Copy the sample font-mapping table, **/usr/lpp/Printsrv/samples/aokfontmap.samp**, to another directory.
- Edit the font-mapping entries in the font-mapping table, and set file permissions so that it is readable by everyone.
- Set the **AOP_FONTMAP_TABLE** environment variable in **aopxfd.conf** transform class entries to point to the file path for the font-mapping table.
- Stop and start Infoprint Server Transform Manager (**aopxfd**) with AOPSTOP and AOPSTART.
- Check for error messages in transform's **/var/Printsrv/xfd stderr** file, and fix any errors, remembering to continue issuing AOPSTOP AND AOPSTART

New environment variables

- **AOP_EDGETOEDGE** (only AFPXPCL)
 - Indicates whether the transform adjusts the x,y left-edge offset position for printing on printers configured for edge-to-edge printing.
- **AOP_PAPER=auto value**
 - Auto means that the transform uses the page size in the Page Descriptor (PGD) structured field in the AFP data.
- **AOP_FAIL_ON_ERROR**
 - Controls whether the transform fails when it detects a data stream warning or error.
- **AOP_TRAILER_ERROR_PAGE**
 - Indicates whether or not to attach a trailer page with warning and/or error messages.

New environment variables

- **AOP_FONTPATH**
 - Specifies the directories that contain TrueType, WorldType, and OpenType fonts that the transform uses to process data sets.
- **AOP_FONTMAP_TABLE**
 - Specifies the file path to a font-mapping table that maps raster fonts to outline fonts.
- **AOP_EMBED_OUTLINES** (only AFPXPS)
 - Indicates whether the transform includes outline fonts in the PostScript output.

Some more obscure V2.1 environment variables no longer supported

- These 2.1 environment variables not supported in Print Transforms V1.1
 - AOP_FONTMAP (replaced by AOP_FONTMAP_TABLE)
 - AOP_POSITIONING_METHOD
 - AOP_RESOLUTION
 - AOP_SIMULATE_240_DPI

-r no longer supported

- Print Transforms from AFP does not support the transform **-r** option, because the transforms automatically scale 240 pel fonts to 300 pel.
- If a printer definition specifies this transform option in the **Filter** field, you can delete it or leave it as-is. Print Transforms from AFP ignores **-r** without error.

Default error handling has changed

- The V2.1 transforms continue processing and give a 0 return code even when input errors occur, stopping only when a critical error is reached.
- New V1.1 transforms stop processing by default whenever an error occurs in the input document.
You can change this by setting
AOP_FAIL_ON_ERROR/fail-on-transform-error/Fail on error.

Init files for non-default installations

- If you install the transform daemons in a directory other than the default **/usr/lpp/Printsrv/bin** directory,
 - When processing PDF objects, there is a special **/usr/lpp/Printsrv/aokfonts** subdirectory that contains necessary fonts. Make sure that this subdirectory is in your new directory structure.
 - Make sure that the **.aokcinit** file is in the same directory since all of the transforms use this file.

Possible error if not:

```
AOP003E afpxps.dll: AOK03227E Transform Config  
File "/u/user/Printsrv/bin/.aokcinit" is  
inaccessible: EDC5129I No such file or directory.
```

Tips and Tricks

Don't get the transforms confused

- **Make sure that you know which transform you want!**
- There are different transform products (to AFP vs. from AFP), they use different resources, and they need to be configured differently.
- Common problems:
 - Specifying an **afp2*** or **afpx*** (which transform **from** AFP **to** PS/PDF/PCL) in the processing section of the printer definition for data other than modca, line, XML or other.
 - Specifying ***2afp** (which transform **to** AFP **from** PS/PDF/PCL/SAP) for modca, line and other.
 - Getting transform resources mixed up.

Common issues

- When you use the print transforms, the amount of available memory for Infoprint Server must be at least 256 megabytes.
- The transform must have read permissions set for each user path directory and file it is accessing.
 - **resource-directories** job attribute
 - **Resource directories** field in the printer definition
 - **USERPATH** parameter of the OUTPUT JCL statement

XML Toolkit

- When transforming XML files, you must install and configure the IBM XML Toolkit for z/OS, C++ edition, V1.10.
 - Add the path to locate the XML DLLs to the LIBPATH environment variable in the **aopstart** EXEC and Infoprint Server libraries before any existing values in LIBPATH.

```
LIBPATH=/usr/lpp/ixm/IBM/xml4c-  
5_7/lib:/usr/lpp/ixm/IBM/xslt4c-1_11/lib:...
```

- After editing the **aopstart** EXEC, stop and restart Infoprint Server.

Tracing for IBM support

- **-T *traceoptions***
Specifies the type of trace you want.
Allowed values are:
 - **all**
All trace options (generates a lot of output).
 - **trans**
File and memory allocations
- To read the trace file in the base directory or a user-supplied directory, you must be a member of the AOPADMIN group.

Tracing for IBM support

- Default location for trace is the transform's base-directory, usually **/var/Printsrv/xfd**. This is deleted when the transform daemon (aopxfd) is stopped and restarted. To write to another location that will persist after the transform daemon has been stopped, specify the trace name with the **-F** option and a fully-qualified path-name.
- How to run a trace on a standalone transform command:
`afpxps -d myprinter -T all -F myfile.trace myfile.afp`
- How to run a trace for an lp command, writing to a non-default location:
`lp -d myprinter -o "filter-options='-T all -F /u/lizs/myfile.trace'" myfile.afp`

Where to go for more info

- IBM Print Transforms from AFP for Infoprint Server for z/OS, Version 1.1 (G325-2634-02)
 - z/OS Internet Library: <http://www-03.ibm.com/systems/z/os/zos/bkserv/>
 - IBM Publications Center: <http://www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss>
 - www.ibm.com/support/docview.wss?uid=pub1g325263402
- Information about OA35704:
<http://www-01.ibm.com/support/docview.wss?uid=isg1OA35704>
- Information about OA38585:
<http://www-01.ibm.com/support/docview.wss?uid=isg1OA38585>

Where to go for more info

- You can search for PMRs using your service request tool or google for APARs, using the component ID for the Print Transforms from AFP products.
 - AFPXPCL: 5655TF2
 - AFPXPDF: 5655TF1
 - AFPXPS: 5655TF3
 - Transforms to AFP: 5655N6001
 - Infoprint Server: 5694-A01

Thank you!

