

Working with Infoprint Manager Transform Programs

Back to Administrator procedures

Infoprint Manager provides transform programs for the following data streams:

- Printer control language (PCL)
- PostScript
- Portable document format (PDF)
- Tag image file format (TIFF)
- Joint photographic experts group (JPEG)
- Graphic interchange format (GIF)
- Line (ASCII, both formatted and unformatted)
- Double Byte text stream

The transforms convert the incoming data stream into the Advanced Function Presentation (AFP) format.

When a job in any of these formats is submitted to print on a PSF printer, Infoprint Manager automatically invokes the appropriate transform. You can also run the transforms from the **Command Prompt** window without printing the generated output. You may find this useful if you intend to print the job later. You can also use these transforms in transform objects and transform-sequences associated with actual destinations. For more information on this function, see *Configuring Data Stream Transform Sequences*. Jobs print more quickly if they have already been transformed.

Most transforms have flags or option keywords that you can use to specify processing information. For a list of the flags or options and values for each transform, refer to the *Infoprint Manager: Reference*. Configuration files, environment variables, and user-exit programs allow you to specify processing information for Infoprint Manager to use when it invokes the transform automatically. You can also use configuration files and environment variables instead of typing flags on the command line whenever you run a transform.

This topic consists of the following procedures:

- “Customizing the PCL, PostScript, and PDF transforms”
- “Customizing the TIFF, JPEG, and GIF transforms” on page 3
- “Working with the transform for line data” on page 5
- “Working with the transform for double byte text streams” on page 9

Note: For information about the fonts used with the double-byte (**db2afp**) transform, refer to the Working with Fonts administration procedure.

Customizing the PCL, PostScript, and PDF transforms

You can customize the transforms for PCL, PostScript, and PDF data by specifying values in the transform configuration files. Table 1 on page 2 shows the default configuration files that Infoprint Manager provides.

In the table, *install_path* is the directory that Infoprint Manager is installed in. If you don't know the install path, you can find it in the **Management Console**. Open the Management Console, click **Edit -> Change Service Configuration** and find the **Install path** field.

You can either modify these files or create your own.

Table 1. Infoprint Manager transforms and configuration files

Data Stream	Transform	Configuration File	Daemon Configuration File
PCL	pcl2afp	<i>install_path</i> \pcl2afp\pcl2afp.cfg	<i>install_path</i> \pcl2afp\pcl2afpd.cfg
PostScript	ps2afp	<i>install_path</i> \ps2afp\ps2afp.cfg	<i>install_path</i> \ps2afp\ps2afpd.cfg
PDF	pdf2afp	<i>install_path</i> \ps2afp\ps2afp.cfg	<i>install_path</i> \ps2afp\ps2afpd.cfg

The PostScript and PDF transforms use the same default configuration files.

Sample configuration file

The following is an example of a configuration file for the PostScript and PDF transforms. The configuration file for the PCL transform is similar.

```
# ps2afp configuration file
# KEYWORD          ps2afp FLAG          PURPOSE
#####            #####
port = 8251         # -P                which TCP/IP port to use
server = 127.0.0.1 # -S                server to connect with
ps_job_timeout = 20 # -j                maximum number of minutes
#                  #                  to process PostScript job
ps_server_timeout = 20 # -s                maximum number of minutes
#                  #                  server waits between jobs
ps_width = 8.5i    # -w                width of generated image
ps_length = 11i    # -l                length of generated image
ps_x_offset = 0i   # -x                left and right margins
ps_y_offset = 0i   # -y                top and bottom margins
ps_resolution = 600 # -r                resolution of printer
ps_output_type = I01_G4 # -a                type of AFP image to
#                  #                  generate
```

The configuration file uses keyword equivalents of the transform flags. Refer to *Infoprint Manager: Reference* for the keywords and values that you can specify in the configuration file.

Daemon configuration files

The PCL, PostScript, and PDF transforms all require a daemon to be running before you can use them. The **pcl2afpd** daemon manages the PCL interpreter portion of the **pcl2afp** transform. The **ps2afpd** daemon manages the PostScript interpreter portion of the **ps2afp** and **pdf2afp** transforms. For more information about these daemons, refer to *Infoprint Manager: Reference*.

Each daemon, like the corresponding transform, uses a configuration file. The daemon configuration file can contain the same keywords as the transform configuration file (except **server**), plus some keywords that control the operation of the daemon. Refer to the *Infoprint Manager: Reference* for the daemon keywords.

Hierarchy of transform options

Infoprint Manager assigns a hierarchy to its use of flags and configuration file information when it runs the PCL, PostScript, or PDF transform program. The following lists the order in which Infoprint Manager uses flags and information specified in configuration files:

1. Any values you specify on the command line, including values in the configuration file you specify with the **-C** flag.

Infoprint commands process flags from left to right. If you enter the same flag more than once, Infoprint Manager uses the last occurrence of the flag to determine the value to use. For example, if you issue the following command:

```
ps2afp -Cconfig.file -r240 -r300 myfile.ps
```

the **ps2afp** command transforms the file using 300-pel resolution. Infoprint Manager ignores the resolution value specified in the configuration file `config.file` as well as the first **-r240** flag and value.

2. Values specified in the **pcl2afp.cfg** or **ps2afp.cfg** configuration file in the current directory, if one exists and if you issue the transform command directly. Infoprint Manager ignores values in the current directory if you invoke the transform command indirectly by submitting a job to print.
3. Values specified in the **pcl2afp.cfg** or **ps2afp.cfg** configuration file in your home directory, if one exists.
4. Values specified in the default transform command configuration file named in "Sample configuration file" on page 2.
5. Values specified in a customized configuration file identified with the **-C** flag of the **pcl2afpd** or **ps2afpd** command when the transform daemon started.
6. Values specified in the default transform daemon configuration file named in "Sample configuration file" on page 2.
7. Default values that are built into Infoprint Manager. These are the same as the transform defaults except that the default PostScript or PDF output type is an IM1 uncompressed image.

Customizing the TIFF, JPEG, and GIF transforms

You can customize the transforms for TIFF, JPEG, and GIF data by specifying values in the transform configuration files. Table 2 shows the default configuration files that Infoprint Manager provides. You can either modify these files or create your own.

Table 2. Default TIFF, JPEG, and GIF Transform Configuration Files

Data Stream	Transform	Configuration Files
TIFF	tiff2afp	<i>install_path</i> \tiff2afp\tiff2afp.cfg <i>install_path</i> \tiff2afp\tiff2afp.3160cfg <i>install_path</i> \tiff2afp\tiff2afp.3900cfg
JPEG	jpeg2afp	<i>install_path</i> \jpeg2afp\jpeg2afp.cfg <i>install_path</i> \jpeg2afp\jpeg2afp.3160cfg <i>install_path</i> \jpeg2afp\jpeg2afp.3900cfg

Table 2. Default TIFF, JPEG, and GIF Transform Configuration Files (continued)

Data Stream	Transform	Configuration Files
GIF	gif2afp	<i>install_path</i> \gif2afp\gif2afp.cfg <i>install_path</i> \gif2afp\gif2afp.3160cfg <i>install_path</i> \gif2afp\gif2afp.3900cfg

Infoprint uses the .3160cfg and .3900cfg configuration files to transform documents that have been submitted to print on an IBM Infoprint 60 printer or IBM Infoprint 4000 laser printer respectively.

Sample configuration file

The following is an example of a configuration file for the TIFF transform. The configuration files for the JPEG and GIF transforms are similar.

```
a=iocal0
choice=full
cmp=g4
ms=5120
msf=0.01
nosniff=yes
noterm=yes
v=yes
pagetype=page
r=600
x=0
y=0
```

The configuration file uses options in the same format as the transform options that you specify on the command line, except that command line options begin with a dash (-). Refer to the *Infoprint Manager: Reference* for the options and values that you can specify in the configuration file.

Hierarchy of transform options

Infoprint Manager assigns a hierarchy to its use of options and configuration-file information when it runs the TIFF, JPEG, or GIF transform program. The following lists the order in which Infoprint uses command-line options and information specified in configuration files:

1. Any values you specify on the command line, including values in the configuration files you specify with the **-C** option.

Infoprint Manager commands process options from left to right. If you enter the same option more than once, Infoprint Manager uses the last occurrence of the flag to determine the value to use. For example, if you specify the following command:

```
tiff2afp -Cconfig.file -r240 -r300 myfile.tif
```

the `tiff2afp` command transforms the file using 300-pel resolution. Infoprint Manager ignores the resolution value specified in the configuration file `config.file` and the first `-r240` flag and value.

2. Values specified in the default transform command configuration file named in Table 2 on page 3.
3. Default values that are built into Infoprint Manager. These are the same as the transform defaults.

Working with the transform for line data

Infoprint Manager includes a line-data transform program that transforms line data into the Advanced Function Presentation (AFP) data stream. Converting the line data allows you to print it on high-speed AFP printer devices.

The **line2afp** command and its keywords are a subset of the AFP Conversion and Indexing Facility (ACIF) **acif** command. When you use this transform and if errors occur, you may receive some error messages that make references to ACIF.

This section describes the following:

- “What is line data?”
- “What are ANSI and machine carriage controls?” on page 6
- “What are variable-length and fixed-length files?” on page 7
- “How does the imageout keyword affect processing?” on page 8
- “Customizing the line data transform” on page 8

What is line data?

Line data is simply unformatted data; data with few or no embedded control characters to create font changes, page breaks, and so on. The term line data can cover a variety of data streams, such as:

- Traditional line printer format, also called *1403 format*
- Unformatted ASCII files without escape sequences
- DBCS (double-byte character set) ASCII files generated for an IBM 5577 or 5587.

Notes:

1. The **line2afp** transform does not handle DBCS ASCII files. For this type of input, use the **db2afp** transform.
2. ACIF accepts EBCDIC DBCS line data files that use SOSI (shift-out/shift-in) characters. ACIF must be coded with the **PRMODE** control statement set to SOSI1, SOSI2 or SOSI3, as documented in the *ACIF User's Guide*.

Line data is typically created by an application program. For example, an application might produce a bank statement that has line data indicating an account's daily balance.

The Infoprint line-data transform lets you format the following kinds of data with a page definition and form definition for printing on Infoprint-managed printer devices:

- S\370 line-mode data

This type of line data was originally designed for a 1403 printer, is typically generated on a S\370 host processor, and is usually EBCDIC characters with embedded control characters (carriage control, table reference characters, and so on). This type of line data can have double-byte code points and shift in\shift out controls.
- Mixed-mode data

S\370 line-mode data embedded with certain AFP structured fields (for example, **Include Page Segment**)
- Single-byte ASCII data with no embedded control characters except for newlines.
- Single-byte ASCII data that contains ANSI carriage-control characters, table-reference characters, or both.

- Single-byte ASCII data that contains carriage returns and form feed controls, if you specify the `asciinp` or `asciinp` user-exit program.

When you transform the line-data job, you must specify a page definition and a form definition. For more information about the page definitions and form definitions available to you, refer to the *Infoprint Manager: Reference*. For more information about associating page definitions and form definitions with jobs, refer to the online help in the Infoprint Manager Administration GUI.

Notes:

1. You cannot use the line-data transform when printing files containing Proprinter ASCII.
2. When the **line2afp** command processes any unrecognized or unsupported keyword, **line2afp** issues a message, ignores the keyword, and continues processing any remaining keywords. The **line2afp** command then stops processing.
3. Infoprint Manager also lets you transform and print DBCS ASCII data and DBCS EUC data without a page definition. For more information, see the **db2afp** command in the *Infoprint Manager: Reference*.

What are ANSI and machine carriage controls?

In many environments (including IBM mainframes and most minicomputers), printable data normally contains a carriage control character. The carriage control character acts as a vertical tab command to position the paper at the start of a new page, at a specified line on the page, or to control skipping to the next line. The characters can be one of two types: ANSI carriage control or machine carriage control.

ANSI carriage control characters

The most universal carriage control is ANSI, which consists of a single character that is a prefix for the print line. The standard ANSI characters are listed in Table 3.

Table 3. ANSI carriage controls

ANSI	Action
space	Single space the line and print.
0	Double space the line and print.
-	Triple space the line and print.
+	Do not space line and print.
1-9	Skip to a vertical position on the page as defined by a vertical tab record (which is either printer- or system-dependent) or by a Page Definition.
A,B,C	Skip to a vertical position on the page as defined by a vertical tab record (which is either printer- or system-dependent) or by a Page Definition.

All ANSI control characters perform the required spacing before the line prints. Using **line2afp** command keywords and values, ANSI controls may be encoded in EBCDIC (**cctype=a**) or in ASCII (**cctype=z**).

Machine carriage control characters

Machine carriage controls were originally the actual hardware control commands for IBM printers, and are often used on non-IBM systems. Machine controls are binary values, not symbols. They are not represented as characters in any encoding and, therefore, machine controls cannot be translated. Some typical machine

controls are listed in Table 4.

Table 4. Machine carriage control characters

Machine	Action
X'09'	Print the line and single space
X'11'	Print the line and double space
X'19'	Print the line and triple space
X'01'	Print the line (do not space)
X'0B'	Space one line immediately (do not print)
X'89'	Print the line, then skip to channel 1 (top of form)
X'8B'	Skip to channel 1 immediately (do not print)

Machine controls print before performing any required spacing. There are many more machine control commands than ANSI.

Carriage controls may be present in a file or not, but if you want to use carriage controls at all, every record in the file must contain a carriage control. If the file contains carriage controls but the **line2afp** command keyword and value **cc=no** is specified, the carriage controls are treated as printing characters. If no carriage controls are specified, the file prints as though it were single spaced.

What are variable-length and fixed-length files?

The line-data transform program needs to know two things about a file in order to transform it:

- The length of each print record
- The kind of carriage control used

Some files contain information in each record that describes the record length; these are called variable-length files. Other files require an external definition of length; these are called fixed-length files.

MO:DCA structured fields are treated as a special case. All such structured fields are self-identifying and contain their own length. They do not need to contain a length prefix to be correctly interpreted, but are also processed correctly if the length prefix is present.

Variable-length and fixed-length files

Variable-length files may use a length prefix, which means they contain a prefix that identifies the length of the record in the file. Each record contains a field that gives the length of the record. If the record contains a length, that length must be a prefix for each record and it must be a 16-bit binary number that includes the length of the 2-byte length prefix. Use the **fileformat=record** keyword and value to identify files with length prefixes.

Variable-length files may use either a separator or a delimiter to indicate the end of a record, instead of using a length prefix. All of the bytes up to, but not including, the delimiter are considered part of the record. For compatibility with other systems, the default record separator for the **line2afp** command is X'0A'. Files created under Windows NT use carriage return/linefeed (X'0D0A') for a delimiter. You must specify **fileformat=stream,(newline=X'0D0A')** to process files created using Windows NT.

Infoprint Manager reads the first six bytes and tests for all ASCII characters (code points from X'00' to X'7F') to determine if a file is encoded in ASCII or EBCDIC. If no non-ASCII characters are found, the line-data transform program assumes that the file uses the ASCII **newline** character, X'0A'. Remember that for files created by Windows NT, you should code `newline=X'0D0A'`, as shown above, instead of accepting the default. Otherwise, the transform assumes the file uses the EBCDIC **newline** character, X'25'. Because an input file can misguide the line-data transform, a set of rules has been established to determine how the transform processes the file.

Table 5.

Data Type	Newline Character
All EBCDIC	EBCDIC X'25'
All EBCDIC	ASCII X'0D0A' or X'0A' ¹
All ASCII	EBCDIC X'25' ¹
All ASCII	ASCII X'0D0A' or X'0A'
<p>1: These combinations are possible only if a file contains a prefix with a string that indicates a different code set than actually exists.</p> <ul style="list-style-type: none"> • For EBCDIC data with ASCII newlines, use X'0320202020200A' as the first seven bytes of the file. • For ASCII data with EBCDIC newlines, use X'03404040404025' as the first seven bytes of the file. 	

Fixed-length files

Fixed-length files contain records that are all the same length. No other separators or prefixes or self-identifying information exists that indicates the record length. You must know the record length and use the **fileformat=record**, *nnn* keyword and value, where *nnn* represents the length of each record.

How does the **imageout** keyword affect processing?

The line-data transform converts image formats (IM1) in the input file, in overlays, and in page segments to uncompressed IOCA format, if **line2afp imageout=ioca** keyword and value is specified (the default). An uncompressed IOCA image may use a significantly higher number of bytes than an IM1 image and may take more processing time to convert, especially for shaded or patterned areas. Although IOCA is the MO:DCA-P standard for image data, and some data stream receivers may require it, all products may not accept IOCA data. All software products from the IBM Printing Systems do, however, accept IOCA data as well as IM1 image data.

If you are using non-IBM Printing Systems products, you may want to specify **imageout=asis**.

Customizing the line data transform

Infoprint Manager provides several sample programming exits to assist you in customizing the line-data transform. Use of the programming exits is optional. You specify the names of the exit programs with the **inpxit**, **outexit**, and **resexit** keywords. Each of these keywords is described in the *Infoprint Manager: Reference*.

Infoprint Manager provides the following sample exits:

```
install_path\exits\acif\apkinp.c
    Input record exit
```


install_path\exits\acif\apkout.c
Output record exit

install_path\exits\acif\apkres.c
Resource exit

In addition, Infoprint Manager provides the following user input record exits to translate line-data data streams:

install_path\exits\acif\apka2e.c
Converts the ASCII stream data into the EBCDIC stream data.

install_path\exits\acif\asciinp.c
Converts unformatted ASCII data that contains carriage returns and form feeds into a record format that contains ANSI carriage control characters. This exit encodes the ANSI carriage control character in byte 0 of every record.

install_path\exits\acif\asciinpe.c
Converts unformatted ASCII data into a record format as does *asciinp.c*, and then converts the ASCII stream data into EBCDIC stream data.

The C language header file for all line-data transform exit programs is provided in the *install_path\exits\acif\apkexits.h* directory, and the build rules for the line-data transform user exits reside in *install_path\exits\acif\exit_program*.

For more information about compiling user exit programs, refer to *Configuring Data Stream Transform Sequences*.

Working with the transform for double byte text streams

Infoprint Manager includes the **db2afp** transform, a program that transforms both double-byte character set (DBCS) ASCII and Extended UNIX Code (EUC) files into the Advanced Function Presentation (AFP) data stream.

This section describes the following:

- “What are DBCS ASCII and EUC?”
- “What font resources are needed to print DBCS ASCII and EUC?” on page 10
- “Determining what code page your print jobs use and setting the correct environment variable” on page 10

What are DBCS ASCII and EUC?

In ideographic languages like Japanese, Chinese, and Korean, there are thousands of characters. All of these characters cannot be represented with a single byte of information because a single byte only accommodates 256 characters. Therefore, at least one more byte of information is required.

In DBCS ASCII files, each ideographic character is exactly two bytes. In EUC files, each ideographic character may be two, three, or four bytes, depending on the EUC implementation and language. Single-byte ASCII characters may be mixed in with DBCS and EUC characters.

Infoprint Manager for Windows NT and Windows 2000 supports printing DBCS ASCII files in the Japanese and Traditional Chinese languages, which can contain formatting controls for the IBM 5577 or 5587 printers. Infoprint Manager for Windows NT and Windows 2000 also supports printing EUC files in the Japanese,

Traditional Chinese, and Korean languages, but you must submit those jobs from a UNIX or AIX workstation. EUC files cannot contain any formatting control characters.

What font resources are needed to print DBCS ASCII and EUC?

You can use the **db2afp** command to transform DBCS ASCII and EUC files that use the following code pages:

- 932 (Japanese PC)
- 950 (Traditional Chinese PC)
- 33722 or eucJP (Japanese EUC)
- 964 or eucTW (Traditional Chinese EUC)
- 970 or eucKR (Traditional Korean EUC)

Note: Code page 938 (Traditional Chinese PC) is not supported on Windows NT or Windows 2000. If your files use code page 938, you cannot use Infoprint Manager for Windows NT and Windows 2000 to print them.

Your Infoprint Manager server must be able to access the appropriate DBCS fonts when you print the transformed files. DBCS fonts are available for the MVS, VM, Windows NT, Windows 2000, and AIX operating systems as separately orderable licensed programs. To make the fonts available on your Infoprint Windows NT or Windows 2000 system, you must ensure that you have installed the fonts on your system and then make the double-byte fonts known to Infoprint Manager for Windows NT and Windows 2000.

For more information on setting up font resources to print DBCS fonts, refer to the **Fonts for Printing DBCS ASCII and EUC** section of the Working with Fonts topic.

Determining what code page your print jobs use and setting the correct environment variable

Before you transform a print job, you need to determine what code page it uses, then set the PSFDBLANG environment variable to that code page. If the job was created on an AIX or a Windows workstation, you can determine the code page that the job uses by finding out what code page is active on that system. On the system that the job was created on, open a **Command Prompt** window and type:

```
chcp
```

then press **Enter**. The system responds with

```
Active code page: xxx
```

where xxx represents a three- to five-digit number. Use this number to set the PSFDBLANG environment variable by following the steps below.

1. On your Infoprint Manager server system, click the Windows **Start** button and select **Settings ->Control Panel**.

Note: If you have more than one Infoprint Manager server system, you need to set this environment variable on all of the servers that contain actual destinations that will use the **db2afp** transform.

2. In the Control Panel window, double-click the **System** icon.
3. In the **System Properties** dialog, click the **Environment** tab.
4. In the **Variable** field, type PSFDBLANG.

5. In Table 6, locate the **Code Page Value** that you got from your system above.

Table 6. Code pages and their associated PSFDBLANG values

Code Page Value	PSFDBLANG Value	Language
932	J	Japanese PC
950	T	Traditional Chinese PC
33722 or eucJP	JE	Japanese EUC
964 or eucTW	CE	Traditional Chinese EUC
970 or eucKR	KE	Traditional Korean EUC

6. Find the **PSFDBLANG Value** that corresponds to that code page and type it in the **Value** field of the **System Properties** dialog.

Note: The PSFDBLANG Value is case sensitive. Be sure to type it in all caps.

7. Click **OK**.

The **db2afp** command assumes that all jobs use this code page as long as you do not specify either the **-i** flag or the **-e** flag when running the command from a **Command Prompt** window.

Back to Administrator procedures