



IBM i  
Printing  
Basic printing

*7.1*







IBM i

Printing

Basic printing

*7.1*

**Note**

Before using this information and the product it supports, read the information in “Notices,” on page 401.

This edition applies to IBM i 7.1 (product number 5770-SS1) and to all subsequent releases and modifications until otherwise indicated in new editions. This version does not run on all reduced instruction set computer (RISC) models nor does it run on CISC models.

© **Copyright IBM Corporation 1999, 2010.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

---

# Contents

## Basic printing . . . . . 1

What's new for IBM i 7.1 . . . . .	1
PDF file for Basic printing . . . . .	1
Printing concepts . . . . .	2
Overview . . . . .	2
Printer file . . . . .	5
Printer file overview . . . . .	5
Printer file overrides . . . . .	6
Spooled files and output queues . . . . .	8
Spooling overview . . . . .	8
Spooled file . . . . .	9
Output queue . . . . .	9
Multiple output queues . . . . .	10
Output queue recovery . . . . .	10
Spooled file cleanup . . . . .	11
Default output queues . . . . .	11
Order of spooled files on an output queue . . . . .	12
Data queue support . . . . .	13
Record type 01 data queue entry format . . . . .	15
Record type 02 data queue entry format . . . . .	16
Record type 03 data queue entry format . . . . .	18
Spooled file names . . . . .	20
Spooled file security . . . . .	20
Output queue security . . . . .	21
QPRTJOB job . . . . .	22
Spooled file subsystem . . . . .	23
Spooled file library . . . . .	23
Spooled files in independent ASPs . . . . .	24
Printer writer program . . . . .	25
Printer data streams . . . . .	26
SNA character string . . . . .	26
Advanced Function Presentation data stream . . . . .	27
Intelligent Printer Data Stream . . . . .	31
American National Standard Code for Information Interchange . . . . .	40
Printer device description . . . . .	40
Job description . . . . .	41
Workstation description . . . . .	41
User profile . . . . .	41
System values . . . . .	41
Controlling printing to output queue or printer . . . . .	42
Routing process order . . . . .	42
Routing examples . . . . .	44
Example 1: Determining your output queue . . . . .	45
Example 2: Determining your output queue . . . . .	45
Example 3: Determining your output queue . . . . .	46
Example 4: Determining your output queue . . . . .	47
Example 5: Determining your output queue . . . . .	47
Example 6: Determining your output queue . . . . .	48
Example 7: Determining your output queue . . . . .	49
Example 8: Determining your printer name . . . . .	50
Example 9: Determining your printer name . . . . .	50
Example 10: Determining your printer name when using batch . . . . .	51
Example 11: Determining your printer name when using batch . . . . .	52

Example 12: Determining your printer name when using batch . . . . .	52
Self-test: Determining output queue and printer device . . . . .	52
Self-test answers . . . . .	53
Printer attachment methods . . . . .	54
TCP/IP-network-attached printers . . . . .	55
Intelligent Printer Data Stream with Print Services Facility . . . . .	55
Simple Network Management Protocol . . . . .	55
Printer Job Language . . . . .	55
Internet Printing Protocol . . . . .	55
Line printer requester/line printer daemon . . . . .	56
PC-attached printers . . . . .	56
Twinaxial-workstation-controller-attached printers . . . . .	56
ASCII-workstation-controller-attached printers . . . . .	57
Lexlink-network-attached printers . . . . .	57
Printers attached to IBM InfoWindow 3477, 3486, 3487, and 3488 display stations . . . . .	57
Remote system printing . . . . .	57
Benefits . . . . .	58
How remote system printing works . . . . .	59
User print information . . . . .	59
Send and defer status . . . . .	60
Fonts . . . . .	61
TrueType and OpenType fonts . . . . .	62
AFP compatibility fonts . . . . .	63
Font character sets . . . . .	63
Font global identifiers . . . . .	65
Code pages . . . . .	68
Stand-alone code pages . . . . .	68
Character set and code page combinations . . . . .	70
Coded fonts . . . . .	71
Double-byte character set support . . . . .	72
Special DBCS printer functions . . . . .	72
Character rotation . . . . .	72
Character expansion . . . . .	73
Condensed printing . . . . .	73
Horizontal and vertical lines . . . . .	74
Shift-control character printing . . . . .	74
Double-byte character printing considerations . . . . .	75
Extended character printing . . . . .	75
Condensed printing . . . . .	75
Page width . . . . .	75
Unprintable double-byte characters . . . . .	76
Double-byte data in an alphanumeric field . . . . .	76
Spanned lines . . . . .	77
Spanned pages . . . . .	77
Print key use . . . . .	77
5553 printer end-of-forms . . . . .	78
Double-byte data printed on alphanumeric printers . . . . .	78
DBCS spooled support . . . . .	79
3130 printer resident font support . . . . .	79
Internet Printing Protocol server . . . . .	79

Network print server . . . . .	80	Verifying IPP server prerequisite programs	127
Network print server objects. . . . .	80	Configuring your Web browser . . . . .	128
How the network print server is accessed . . . . .	80	Starting the administrator interface . . . . .	128
Exit points and the network print server. . . . .	82	Configuring an IPP server . . . . .	128
Exit point QIBM_QNPS_ENTRY . . . . .	83	Configuring an IPP printer . . . . .	129
Exit point QIBM_QNPS_SPLF . . . . .	84	Configuring an IPP client . . . . .	129
Use of network print server exit points . . . . .	86	Configuring Lexlink printers . . . . .	129
Communications network printing functions . . . . .	87	Configuring remote system printing . . . . .	131
3270 printer emulation. . . . .	87	i5/OS to i5/OS for remote system printing	131
Remote job entry printing . . . . .	88	i5/OS to VM/MVS for remote system	
3x74-attached printers . . . . .	89	printing . . . . .	137
Distributed data management printing . . . . .	90	Configuring host print transform. . . . .	141
Object distribution printing . . . . .	91	Customizing ASCII printers that use the host	
Host print transform . . . . .	91	print transform function. . . . .	141
Host print transform advantages . . . . .	91	Preparing to customize an ASCII printer	141
How the host print transform function works	92	Retrieving the workstation customization	
AFP-to-ASCII transform function . . . . .	93	source. . . . .	142
Host print transform in raster mode . . . . .	95	Customizing WSCST object of	
Host print transform and emulator products	96	*TRANSFORM transform type . . . . .	143
Image print transform . . . . .	96	Customizing WSCST object of *CTXFORM	
Image print transform overview . . . . .	96	transform type . . . . .	196
Image print transform advantages. . . . .	97	Changing an existing printer device	
Image configuration objects . . . . .	97	description . . . . .	205
Convert Image API. . . . .	98	Displaying the printer device description . . . . .	206
Conversion of PostScript data streams with		IBM IBM i Access for Windows PC5250	
image print transform . . . . .	98	configuration recommendations . . . . .	206
Page size . . . . .	100	3486, 3487, and 3488 InfoWindow display	
Additional printing considerations . . . . .	101	station configuration recommendations. . . . .	206
Direct print considerations . . . . .	101	3477 InfoWindow display station	
Open considerations . . . . .	101	configuration recommendations . . . . .	207
Output considerations . . . . .	101	3197 display station configuration	
Close considerations . . . . .	101	recommendations . . . . .	208
First-character forms-control data		ASCII workstation controller configuration	
considerations . . . . .	102	recommendations . . . . .	209
Printer font considerations . . . . .	103	IBM remote 5250 emulation program	
Alternative character sets and code pages for		configuration recommendations . . . . .	210
printer output considerations . . . . .	106	Configuring image print transform . . . . .	211
Output field considerations. . . . .	108	Printing to an ASCII printer with image print	
Externally described printer-file		transform. . . . .	211
record-format considerations . . . . .	109	Printing to an IPDS printer with image print	
Redirecting output considerations . . . . .	110	transform. . . . .	211
3812 and 3816 SCS printer considerations . . . . .	113	Printing with remote output queues and	
3835 printer considerations . . . . .	113	image print transform . . . . .	211
3912, 3916, and 4028 printer considerations	114	Managing printing. . . . .	211
Special printer file considerations for AFP		Managing spooled files . . . . .	212
data stream . . . . .	114	Displaying a list of spooled files . . . . .	212
Special data description specifications		Displaying the contents of a spooled file . . . . .	212
considerations for AFP data stream . . . . .	115	Displaying messages associated with a	
Performance considerations. . . . .	117	spooled file . . . . .	213
Planning printing . . . . .	117	Holding a spooled file . . . . .	213
Configuring printing . . . . .	118	Releasing a spooled file . . . . .	213
Automatically creating printer device		Moving a spooled file . . . . .	214
descriptions . . . . .	118	Deleting spooled files . . . . .	214
Adding a printer with System i Navigator. . . . .	118	Converting a spooled file to PDF. . . . .	214
Configuring PJI printers. . . . .	119	Exporting a spooled file to your PC . . . . .	215
Configuring SNMP printers . . . . .	120	Copying a spooled file to a physical file . . . . .	215
Configuring LPR/LPD . . . . .	122	Sending a spooled file to another user or	
Configuring IPP printers . . . . .	125	system . . . . .	215
Setting up validation lists for the IPP print		Changing attributes of a spooled file . . . . .	216
driver . . . . .	126	Restarting the printing of a spooled file . . . . .	216
Setting up the IPP server . . . . .	127		

Suspending one spooled file and printing another . . . . .	216	Example: Using an externally described printer file with an application program . . . . .	244
Enabling spooled file notification message	217	Troubleshooting printing . . . . .	248
Controlling the number of spooled files . . . . .	217	Troubleshooting the IPP server . . . . .	248
Deleting expired spooled files . . . . .	218	Troubleshooting the image print transform . . . . .	250
Reclaiming spooled file storage . . . . .	218	Reference information . . . . .	251
Saving and restoring spooled files . . . . .	219	CL commands . . . . .	251
Controlling printing by spooled file size . . . . .	220	Jobs . . . . .	251
Repairing output queues and spooled files	221	Output queues . . . . .	251
Printer-file-related tasks . . . . .	221	Printer devices . . . . .	252
Changing printer files . . . . .	221	Printer files . . . . .	252
Overriding printer files . . . . .	221	Printer writers . . . . .	252
Overriding file attributes . . . . .	222	Spooled files. . . . .	253
Overriding file names or types . . . . .	223	User profiles. . . . .	253
Overriding file names or types and file attributes of the new file . . . . .	224	Printer file parameter considerations . . . . .	254
Generic override for printer files . . . . .	224	Align (ALIGN) parameter . . . . .	254
Deleting printer file overrides . . . . .	225	Coded font (CDEFNT) parameter. . . . .	255
Displaying printer file overrides . . . . .	225	Corner staple (CORNERSTPL) parameter . . . . .	255
Managing printers. . . . .	225	DBCS coded font (IGCCDEFNT) parameter	256
Checking the status of a printer . . . . .	225	Device type (DEVTYPE) parameter . . . . .	256
Making a printer available . . . . .	226	Edge stitch (EDGESTITCH) parameter . . . . .	257
Making a printer unavailable . . . . .	226	Fidelity (FIDELITY) parameter . . . . .	259
Managing printer writers . . . . .	226	Font character set (FNTCHRSET) parameter	259
Starting a printer writer . . . . .	226	Margin (FRONTMGN and BACKMGN) parameters . . . . .	260
Stopping a printer writer . . . . .	226	MULTIUP(1, 2, 3, or 4) and REDUCE(*NONE) parameters . . . . .	263
Checking the status of a printer writer . . . . .	226	MULTIUP(1, 2, 3, or 4) and REDUCE(*TEXT) parameters . . . . .	267
Managing remote writers . . . . .	227	MULTIUP support. . . . .	268
Displaying remote writer status . . . . .	227	MULTIUP support for the 4224, 4230, 4234, 4247 printers. . . . .	270
Starting a remote writer . . . . .	227	Restrictions for MULTIUP with REDUCE(*TEXT) . . . . .	271
Stopping a remote writer . . . . .	227	Output priority (OUTPTY) parameter . . . . .	271
Managing IPP server . . . . .	227	Overlay (FRONTOVL and BACKOVL) parameters . . . . .	272
Starting the IPP server . . . . .	228	Page rotation (PAGRIT) parameter . . . . .	278
Stopping the IPP server . . . . .	228	PAGRIT = *AUTO . . . . .	279
Changing an IPP printer configuration . . . . .	228	PAGRIT = *DEVVD . . . . .	279
Displaying an IPP printer configuration . . . . .	228	PAGRIT = 0, 90, 180, or 270 degrees . . . . .	279
Deleting an IPP printer configuration . . . . .	228	PAGRIT = *COR . . . . .	279
Miscellaneous printing tasks . . . . .	229	Page rotation on the 3831, 3835, 3900, InfoPrint 3000, and InfoPrint 4000 printers	280
Allowing direct printing. . . . .	229	Page rotation and duplex printing . . . . .	281
Specifying separator pages . . . . .	229	Saddle stitch (SADLSTITCH) parameter . . . . .	282
Specifying print text . . . . .	230	Spooled output schedule (SCHEDULE) parameter . . . . .	283
Replacing unprintable characters . . . . .	230	To stream file (TOSTMF) parameter . . . . .	284
Printing a graphic with other output . . . . .	231	Workstation customizing object (WSCST) parameter . . . . .	284
Displaying graphic symbol sets . . . . .	232	Printer file return codes . . . . .	285
Changing remote output queue description	232	Major code 00 (00xx) . . . . .	286
Examples: Using DDS and printer files . . . . .	232	Major code 80 (80xx) . . . . .	288
Examples: Using DDS with RPG or COBOL . . . . .	232	Major code 81 (81xx) . . . . .	291
Example: DDS row and column positioning source code . . . . .	232	Major code 82 (82xx) . . . . .	293
Example: DDS absolute positioning source code . . . . .	233	Major code 83 (83xx) . . . . .	295
Example: COBOL positioning source code	234	Printer file feedback areas . . . . .	298
Example: RPG positioning source code . . . . .	235	Open feedback area for printer . . . . .	298
Example: Print font samples . . . . .	236	Device definition list . . . . .	301
DDS source code . . . . .	238	Open feedback extensions list . . . . .	303
C source code . . . . .	238		
COBOL source code . . . . .	238		
RPG source code . . . . .	239		
Example: Using a program-described printer file with an application program . . . . .	239		

Stream file name open feedback extension	303
I/O feedback area	303
AFP compatibility fonts and font substitution	306
Font-related examples	307
Font information	307
Font attributes and font types	317
Printer font support	317
Font substitution by font ID range	330
Mapping host-resident to printer-resident font character sets	331
Mapping host-resident to printer-resident code pages	356
Mapping printer-resident to host-resident font character sets	358
Mapping printer-resident to host-resident code pages	375
CHRID values supported	379
LPI values supported	385
CPI values supported	386
4019 printer information	388
4234 compressed font substitution by lines-per-inch value	390

QWP4019 program	391
QWP4019 parameter names and functions	391
QWP4019 program use	393
QPQCHGCF program	394
QPQCHGCF parameter names and functions	395
QPQCHGCF program use	396
Restrictions on using the QPQCHGCF program	396
Coded fonts whose font character sets reside in the 3130	397
QPQCHGCF instructions for marking coded fonts	398
Related information for Basic printing	398

<b>Appendix. Notices</b>	<b>401</b>
Programming interface information	402
Trademarks	403
Terms and conditions	403



---

## Basic printing

IBM® i products have powerful printing and display functions. You can present information using overlays, bar codes, graphics, images, and more. IBM i products support a variety of commercial printing and presentation solutions.

This topic illustrates some of the core i printing functions, and helps you plan for and configure these functions.

---

### What's new for IBM i 7.1

Read about new or significantly changed information for the Basic printing topic collection.

#### Spooled file security exit

Spooled file security has been enhanced through the addition of a spooled file security exit point. This exit point can be used with a spooled file security exit program to allow more granular access to individual spooled files based on the operation to be performed.

See “Spooled file security” on page 20 and its subtopics for details.



#### Exporting spooled files as PDF

IBM Navigator for i has been enhanced to export spooled files (printer output) to a PC file system in Adobe Portable Document Format (PDF) format.

See “Exporting a spooled file to your PC” on page 215 for details.

#### How to see what's new or changed

To help you see where technical changes have been made, the information center uses:

- The  image to mark where new or changed information begins.
- The  image to mark where new or changed information ends.

In PDF files, you might see revision bars (|) in the left margin of new and changed information.

To find other information about what's new or changed this release, see the Memo to users.

---

### PDF file for Basic printing

You can view and print a PDF file of this information.


To view or download the PDF version of this document, select Basic printing (about 2547 KB).

#### Saving PDF files

To save a PDF on your workstation for viewing or printing:

1. Right-click the PDF link in your browser.
2. Click the option that saves the PDF locally.
3. Navigate to the directory in which you want to save the PDF.
4. Click **Save**.

## Downloading Adobe Reader

You need Adobe Reader installed on your system to view or print these PDFs. You can download a free copy from the Adobe Web site ([www.adobe.com/products/acrobat/readstep.html](http://www.adobe.com/products/acrobat/readstep.html)) .

---

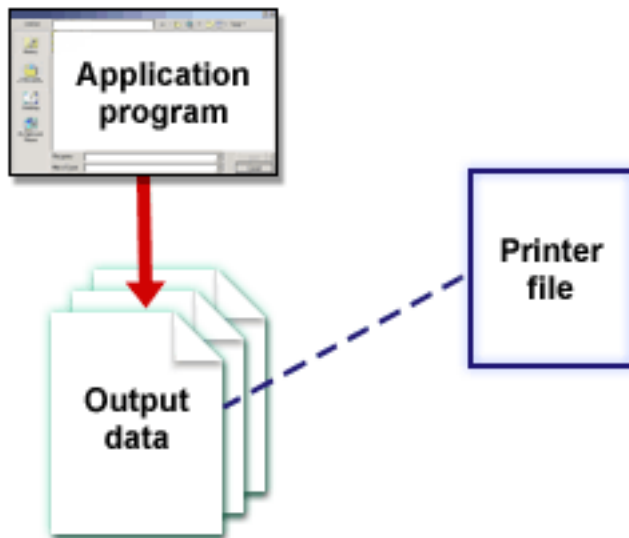
## Printing concepts

Read this section for an overview of the printing function and related concepts.

### Overview

These steps give a high-level overview of the i5/OS printing process.

1. The printing process starts when an application program runs. The application program creates output data. The output data is based on the application program and information contained in the printer file.



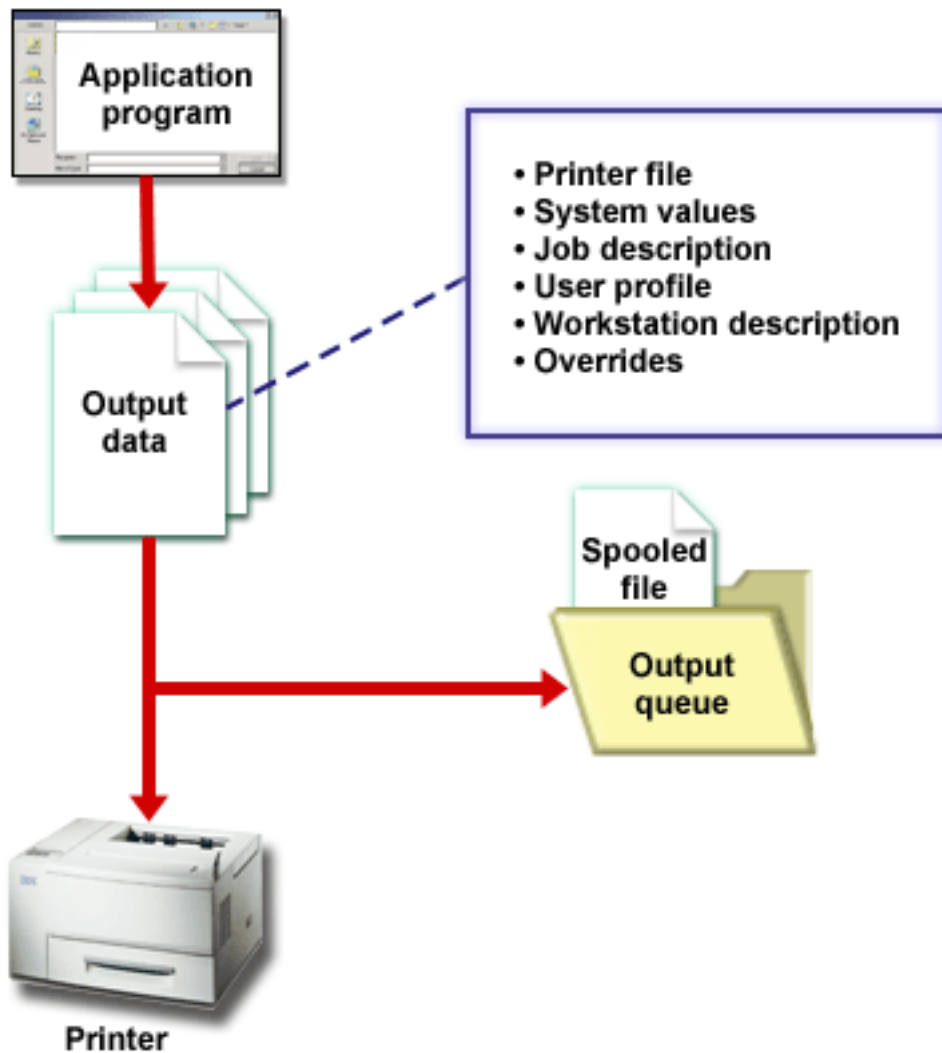
2. If print spooling is selected, the output data is placed in a spooled file and the spooled file is placed in an output queue. If direct printing is selected, the output data is sent directly to the printer. The destination of the output data (see “Controlling printing to output queue or printer” on page 42) is based on values stored in several printing elements:

- Job description
- User profile
- Workstation description
- Printer file
- System values

Output queues are used to manage spooled files. Spooled files in output queues can be:

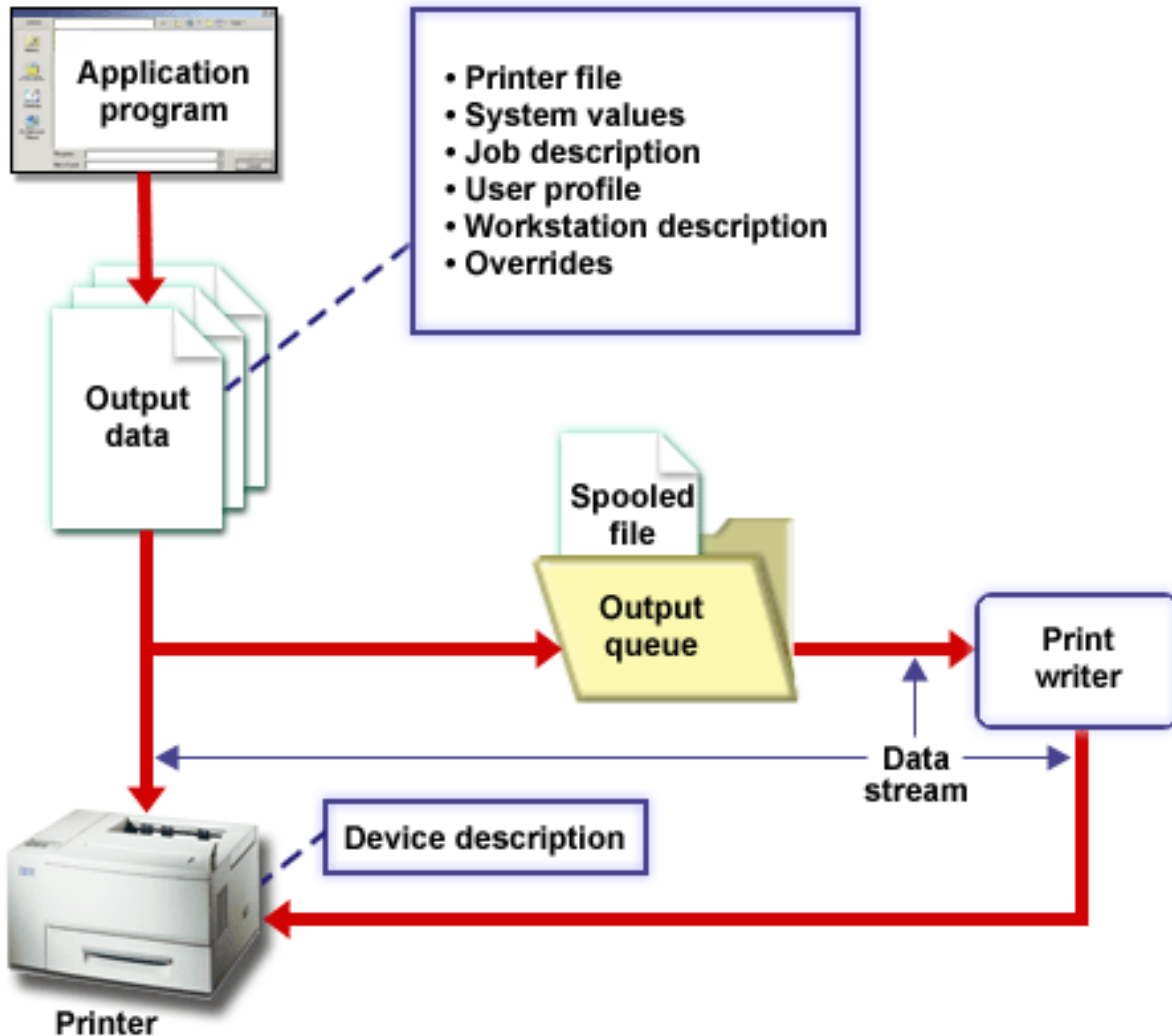
- Printed
- Kept as records
- Used as input to other applications
- Transferred to other output queues
- Sent as e-mail
- Used to create PDF files

Spooled files can also be received from other i5/OS and non-i5/OS systems.



3. The printer writer program interacts between the output queue and the printer and can be used to convert the printer data stream. The printer writer program included in i5/OS supports a variety of printer data streams. Print Services Facility™ for i5/OS provides additional function that provides support for the Advanced Function Presentation (AFP) Intelligent Printer Data Stream (IPDS).

Each printer must have a printer device description. The printer device description contains a configuration description of the printer. Printers can be attached by a variety of attachment methods. A remote writer allows you to route spooled files from an output queue on your system to another system.



#### Related concepts:

“Printer file” on page 5

Printers attached to System i<sup>®</sup> products are supported by the IBM i operating system through printer files.

“Spooled file” on page 9

Spooling is a system function that saves data in a database file for later processing or printing. This data, which is saved and eventually printed, is called a *spooled file* (or printer output file). When spooling is used, spooled files are created from the application program, from a system program, or by pressing the Print key. These files are put in places called output queues.

“Output queue” on page 9

*Output queues* are objects, defined to the system, that provide a place for spooled files to wait until they are printed. Output queues are created by a user or by the system.

“Job description” on page 41

A *job description* is a system object, made up of many parameters, that defines how a job is to be processed.

“User profile” on page 41

The *user profile* is an object with a unique name that contains the user password, the list of special authorities assigned to a user, and the objects the user owns.

“Workstation description” on page 41

The *workstation description*, for a display station, is a collection of information that tells the system how the display station is to be used.

“System values” on page 41

System values are objects supplied by IBM and included in the system. System values control such things as system date, system time, default system printer, and so on.

“Printer writer program” on page 25

The *printer writer program* is a system-supplied program that takes spooled files from an output queue and sends them to a printer. The spooled files on a particular output queue remain stored in the system until the printer writer program assigns a printer to the output queue.

“Printer data streams” on page 26

The IBM i operating system supports different data streams, such as SNA character string (SCS), Advanced Function Presentation data stream, Intelligent Printer Data Stream (IPDS), and American National Standard Code for International Interchange (ASCII) data stream.

“Printer device description” on page 40

Printer device descriptions must be created for each printer attached to the system.

“Remote system printing” on page 57

Remote system printing allows spooled files that are created on a System i platform to be automatically sent to and printed on other systems.

## Printer file

Printers attached to System i products are supported by the IBM i operating system through printer files.

### Printer file overview

Printer files describe how the system operates on the data as it passes between your application program and a printer.

A printer file handles every request for printing with one exception. It applies only to the Print key when the display station has an attached remote workstation controller. In this situation, the printer that is named in the device description for that display station receives the printed version of the display screen. The remote workstation controller, not the system, handles this.

Printer files contain many parameters that tell the system how the output should be formatted, what font to use for the printed output, whether to print on both sides of the page, and more. For example, the parameters that control how your output is handled and where it goes are:

- Spool the data (SPOOL). \*YES is the default value for the SPOOL parameter.

When the SPOOL parameter is set to \*YES, the output from an application program (a spooled file) is sent to an output queue (OUTQ). When SPOOL = \*YES, the system looks at the OUTQ parameter in the printer file to find out which output queue (OUTQ) to send the spooled file to. For example, the OUTQ value in your printer file can be OUTQ1. However, in the default printer file, QSYSPRT, the value specified is \*JOB. This means that the QSYSPRT printer file tells the system to look at your job's OUTQ attribute to determine the name of the output queue (OUTQ).

When the SPOOL parameter is set to \*NO, the output from an application program is sent directly to a printer. When SPOOL = \*NO, the system looks at the DEV parameter in the printer file to find out which printer to send the output to. For example, the DEV value in your printer file can be PRT01. However, in the default printer file, QSYSPRT, the value specified is \*JOB. This means that the QSYSPRT printer file tells the system to look at your PRTDEV job's attribute to determine the name of the printer device (DEV).

- Device (DEV). The device parameter indicates the name of a printer device description. If SPOOL = \*NO is specified, the device parameter identifies the printer device used to produce the printed output. If SPOOL = \*YES is specified, the device (DEV) parameter is ignored unless \*DEV is specified for the output queue parameter. In that case, the default output queue for the specified printer is used for the spooled files.

- Spooled output queue (OUTQ). The output queue parameter indicates which output queue your spooled files are sent to. If you have a program that creates large print jobs you might consider sending them to an output queue that will hold those spooled files until most of your printer work for the day is done. Doing this can help users who have lots of small jobs get their jobs printed in a reasonable amount of time.
- Spooled file owner (SPLFOWN). The SPLFOWN parameter specifies which user profile owns a file. The spooled file can be owned by the current job, a QPRTJOB for the current user, a group profile for the current user, or the group profile of the current job's user. For example, for system job debugging purposes, you might want the current job to own all service dumps, instead of the current user. To accomplish this, you can set the parameter to the special value of \*JOB for the QPSRVDMIP printer file.

There are two different types of printer files:

- **Program-described printer files** rely on the high-level language program to define records and fields to be printed.
- **Externally described printer files** use data description specifications (DDS) rather than the high-level language to define records and fields to be printed. DDS gives the application programmer much more format and print control over the printed output.

You can create your own printer files by using the Create Printer File (CRTPRTF) CL command, or you can use system-provided printer files. See Printer file parameter considerations for more detailed information about certain printer file parameters.

The following list contains the IBM-supplied printer files:

- QSYSPRT: A program-described printer file in library QSYS.
- QPRINT: Default spooled output printer file for normal print.
- QPRINTS: Default spooled output printer file for special forms.
- QPRINT2: Default spooled output printer file for two copies of output.
- QPSPLPRT: Default spooled output printer file for the spooling subsystem.

If another printer file is not specified, the printer uses the default printer file. The default printer file for the system is QSYSPRT.

**Related reference:**

“Example: Using a program-described printer file with an application program” on page 239

This example shows how an application program calls and uses a program-described printer file to control how your printed output looks.

“Example: Using an externally described printer file with an application program” on page 244

This example shows how an application program calls and uses an externally described printer file to control how your printed output looks.

“Printer file parameter considerations” on page 254

These printer file parameters are discussed in more detail to help you better understand how to use them.

**Related information:**

DDS for printer files

Create Printer File (CRTPRTF) command

**Printer file overrides**

Overriding files (printer files, display files, diskette files, database files, and tape files) can be done by commands, from CL programs, or from high-level language programs. Overrides can be called from different levels (programs calling another program). This article discusses overriding printer files.

Overrides are used to temporarily specify a different printer file or temporarily change some of the attributes of a file. An override is only active for the current sign on session. As soon as you end your session or use the Delete Override (DLTOVR) command, the override is no longer active.

Override commands can be entered interactively from a display station or as part of a batch job. They might be included in a control language (CL) program, or they might be issued from other programs through a call to the program QCMDXEC. Regardless of how they are issued, overrides remain in effect only for the job, program, or sign on session in which they are issued. Overrides have no effect on other jobs that might be running at the same time.

Overrides are particularly useful for making minor changes to the way a program functions or for selecting the data on which it operates, without having to recompile the program. Their principal value is in allowing you to use general purpose programs in a wider variety of circumstances. Examples of items where overrides can be used are:

- Changing the name of the file to be processed
- Indicating whether output is to be spooled
- Changing printer characteristics such as lines per inch and number of copies

It is also possible to use overrides to direct data input or data that is sent to a device of a different type. For example, sending data that is intended for a diskette to a printer instead. This use of overrides requires somewhat more foresight than the override applications listed above. The program must be able to accommodate the different characteristics of the two devices involved. For information about the special considerations that are required for overrides that change the file type or redirect files, see the Distributed data management topic.

Files are associated with an application program by the file names specified in the program when it is created. You can override these file names or attributes of a specified file when you compile a program or run a program. The system supplies three override functions: applying overrides, deleting overrides, and displaying overrides. You can process override functions for files using the following CL commands:

- Override with Printer File (OVRPRTF) command
- Delete Override (DLTOVR) command
- Display Override (DSPOVR) command

You can use overrides to change most, but not all, of the file attributes that are specified when the file is created. In some cases, you can specify attributes in overrides that are not part of the original file definition. Refer to the command descriptions for more information.

Overriding a file is different from changing a file in that an override does not permanently change the attributes of a file. For example, if you override the number of copies specified in a printer file by requesting six copies instead of two, the file description for the printer file still specifies two copies, but six copies are printed. The override command tells the system which file to open and what its file attributes are.

### **CL program override considerations**

If a CL program overrides a file and then calls a high-level language program, the override remains in effect for the high-level language program. However, if a high-level language program calls a CL program that overrides a file, the override is deleted automatically when control returns to the high-level language program.

High-level language program:

```
CALL CLPGM1
```

CL program:

```
OVRPRTF FILE(PRTF1) TOFILE(MSTOUT)
.
.
.
ENDPGM
```

High-level language program:

```
OPEN PRTF1
```

The file opened is PRTF1, not MSTOUT. This is because the override in the CL program is deleted when the CL program ends.

### Securing printer files

You might want to prevent the person or program that calls your program from changing the printer file names or parameters you have specified.

You can prevent additional printer file overrides by specifying SECURE(\*YES) on the printer file override command for each printer file you want to protect from overrides.

#### Related information:

Distributed database programming

Override with Printer File (OVRPRTF) command

Delete Override (DLTOVR) command

Display Override (DSPOVR) command

## Spooled files and output queues

The spooling function places spooled files (also known as printer output) in an output queue. This allows you to manage your printing operations more effectively.

### Spooling overview

Spooling functions are performed by the system without requiring any special operations by the program that creates the output. When a program opens a printer file, the operating system determines if the output is to be spooled by looking at the printer file SPOOL parameter.

When a printer file specifying spooling is opened, the spooled file containing the output of the program (data to be printed) is placed on the appropriate output queue in the system. A spooled file can be made available for printing when the printer file is opened, when the printer file is closed, or at the end of the job. This is done by specifying a particular value on the schedule parameter. \*IMMED makes the spooled file available to the writer as soon as the program is opened. \*FILEEND makes the spooled file available to the writer as soon as the file is closed. \*JOBEND makes the spooled file available to the writer as soon as the job is complete.

This process of spooling prevents a potential job limitation imposed by the availability or speed of the printer devices. That is, the system can process application programs that generate printed output much faster than printers can print the output.

By spooling (that is, sending the output to output queues to await printing), the system does not have to wait until the printing for that application program is complete before it can start processing the next application program.

Spooling is especially important in a multiple-user environment where the number of jobs running often exceeds the number of available printer devices. Using spooling, output can be easily redirected from one output queue to another or from one printer to another.



## Spooled file

Spooling is a system function that saves data in a database file for later processing or printing. This data, which is saved and eventually printed, is called a *spooled file* (or printer output file). When spooling is used, spooled files are created from the application program, from a system program, or by pressing the Print key. These files are put in places called output queues.

Almost all application programs that generate printed output make use of the spooling support provided by the IBM i operating system. The values SPOOL(\*YES) and SPOOL(\*NO) on the SPOOL parameter of a printer file determine whether spooling support is requested.

Using the Print key to capture an image of a display screen almost always results in a spooled file being created (SPOOL = \*YES must be specified in the printer file named in the workstation device description). Unless the value has been changed, the default value for the SPOOL attribute in the QSYSPRT printer file is \*YES. When the Print key is pressed, the system looks at the OUTQ parameter in the QSYSPRT printer file to determine which output queue to send the spooled file to.

Spooling (SPOOL = \*YES) has several advantages over direct output (SPOOL = \*NO in the printer file):

- The user's display station remains available for work.
- Other users can request printing work without having to wait for the printer to become available.
- If special forms are required, you can have the spooled files sent to a special output queue and printed at a time when the printer is not busy.
- Because disk operations are much faster than printers, the system is used efficiently.

### Related concepts:

“Managing printing” on page 211

Administering your company's printing solution after the initial setup and configuration involves dealing with changing user and application requirements and handling any upgrades or performance considerations as your system grows.

## Output queue

*Output queues* are objects, defined to the system, that provide a place for spooled files to wait until they are printed. Output queues are created by a user or by the system.

You can create an output queue using the Create Output Queue (CRTOUTQ) command. On the prompt display, specify the name for the output queue to create. The output queue will be in the library identified by the library prompt. You can create as many output queues as you want.

When a printer is configured to the system, either manually or through automatic configuration, the system creates an output queue for that printer in the QUSRSYS library. System-created output queues are commonly called device output queues and have the same name as the printer device. For example, when you configure a printer using the Create Device Description (Printer) (CRTDEVPRT) command, if you assign the printer name PRT01 in the DEVN parameter, the system creates an output queue named PRT01 in the QUSRSYS library.

If none of the IBM-supplied default values for the system have been changed, you can identify your output queue by displaying the system value Default printer (QPRTDEV). Your output queue has the same name as the value shown for the system printer.

Spooled files are created when application programs are run. If you do not want the spooled files to print right away, you can have them sent to an output queue that currently does not have a printer assigned to it. For example, let us assume that you have only one printer available. One of your application programs creates a job that has 600 pages of printed output. Since all users are using the same printer, you do not want to print the 600-page job until everyone has finished working for the day. One solution is to create

two separate output queues. One output queue receives the spooled files from the application program that creates the 600 pages of printed output. The other output queue receives the spooled files from the jobs run by other users.

The program that creates the 600-page job sends the spooled file to a specific output queue. That output queue does not have a printer assigned to it. Therefore, the 600-page spooled file has to wait until a printer is assigned; meanwhile, the spooled files that are in the other output queue can be printed. Multiple output queues can also be used with deferred printing. To print a large spooled file that exceeds the current limit for the printer's output queue, the printer can be assigned to an output queue without any limit. Another solution is to set the maximum spooled file size to print during a specified time. For example, a maximum spooled file size of 100 pages can be set from 08:00:00 to 17:30:00 hours. During this time, only spooled files of 100 or fewer pages are printed. After 5:30 p.m. any spooled file prints. Spooled files that are too large are placed in deferred status (\*DFR) until they can be printed. See Controlling printing by spooled file size for more information about how to configure deferred printing.

#### **Related concepts:**

“Controlling printing by spooled file size” on page 220

You can use the MAXPAGES parameter on the Create Output Queue (CRTOUTQ) or Change Output Queue (CHGOUTQ) command to control the printing of spooled files by size.

### **Multiple output queues**

You might want to create multiple output queues for these reasons.

- Special forms printing
- Output to be printed after normal working hours
- Output that is not printed

An output queue can be created to handle spooled files that need only be displayed or copied to a database file. Care should be taken to remove unneeded spooled files.

- Special uses

For example, each programmer can be given a separate output queue.

- Output of special system files

You might want to consider separate queues for the following system-supplied files:

- **QPJOBLOG:** You might want all job logs sent to a separate queue.
- **QPPGMDMP:** You might want all program dumps sent to a separate queue so you can review and print them if needed or clear them daily.
- **QPSRVDMP:** You might want all service dumps sent to a separate queue so the service representative can review them if needed.

### **Output queue recovery**

If a job that has produced spooled files is running when the job or system stops abnormally, the files remain on the output queue. Some number of records written by active programs might still be in main storage when the job ends and will be lost. You should check these spooled files to ensure that they are complete before you decide to continue using the files.

You can specify if all spooled files (except QPJOBLOG) created by the job are to be kept for normal processing by the printer writer, or if these files are to be deleted.

If an abnormal end occurs, the spooled file QPJOBLOG will be written at the next IPL of the system.

If a printer writer fails while a spooled file is being printed, the spooled file remains on the output queue intact.

### **Recovery of user-created output queues**

If an output queue becomes damaged in such a way that it cannot be used, you are notified by a message sent to the system operator message queue. The message comes from a system function when a printer writer or a job tries to add or remove spooled files from the damaged queue.

You can manually delete a damaged output queue or it will be deleted by the system during the next IPL.

After a damaged output queue is deleted, all spooled files on the damaged output queue are moved to output queue QSPRCLOUTQ in library QRCL. The move operation is performed by the QSPRC00001 system job. The system job issues a completion message to the QSYSOPR message queue when all spooled files have been moved.

If the output queue resides in an independent ASP, then the spooled files are moved to output queue QSPRCLOUTQ in library QRCLxxxx, where xxxx is the independent ASP number of the primary independent ASP (for example, QRCL00033 if the primary independent ASP number is 33). The move operation is performed by the QSPRCxxxx job, where xxxx is the independent ASP number of the primary independent ASP. A message is sent to the QSYSOPR message queue when all spooled files have been moved.

After the damaged output queue is deleted, it can be created again. Then, spooled files on output queue QSPRCLOUTQ can be moved to the newly created output queue.

### **Recovery of system-created output queues**

If the output queue that was damaged was the default output queue associated with a printer, the system automatically re-creates the output queue when it is deleted.

This system-created output queue has the same public authority as specified for the device and default values for the other parameters. After the system re-creates the output queue, you should verify its attributes are correct or change them, if necessary.

When a damaged output queue associated with a printer is deleted and created again, all spooled files on the damaged queue are moved to the re-created output queue. This is done by the QSPRC00001 system job. The system job issues a completion message to the QSYSOPR message queue when all spooled files have been moved.

### **Spooled file cleanup**

System spooled file recovery starts immediately following an initial program load (IPL). Spooled file recovery is done under the system job QSPRC00001. Spooled files on destroyed user-created output queues are moved to output queue QSPRCLOUTQ in library QRCL. Spooled files on destroyed system-created output queues are moved to the re-created output queues.

Spooled file recovery also starts immediately following a vary-on operation of an independent ASP group. Spooled file recovery is done under the system job QSPRCxxxx, where xxxx is the independent ASP number of the primary independent ASP. Spooled files on destroyed user-created output queues are moved to output queue QSPRCLOUTQ in library QRCLxxxx, where xxxx is the independent ASP number of the primary independent ASP.

### **Default output queues**

When a printer is configured to the system, the system automatically creates the printer's default output queue in library QUSRSYS. The output queue is given a text description of 'Default output queue for printer xxxxxxxxxx', where xxxxxxxxxx is the name assigned to the printer during configuration. The printer name is specified in the device description (DEVVD) parameter.

The AUT parameter for the output queue is assigned the same value as that specified by the AUT parameter for the printer device description. All other parameters are assigned their default values. Use

the Change Command Default (CHGCMDDFT) command to change the default values used when creating output queues with the CRTOUTQ command.

The default output queue for a printer is owned by the user who created the printer device description. In the case of automatic configuration, both the printer and the output queue are owned by the system profile QPGMR.

The system is shipped with the defaults set to use the default output queue for the system printer as the default output queue for all spooled output. The system printer is defined by the Default printer (QPRTDEV) system value.

When a spooled file is created by opening a printer file and the output queue specified for the file cannot be found, the system attempts to place the spooled file on output queue QPRINT in library QGPL. If for any reason the spooled file cannot be placed on output queue QPRINT, an error message is sent and the output is not spooled.

The following output queues are supplied with the system:

Output queue	Description
QPRINT	Default printer output queue
QPRINTS	Printer output queue for special forms
QPRINT2	Printer output queue for 2-part paper

## Order of spooled files on an output queue

The order of spooled files on an output queue is mainly determined by the status of the spooled file.

A spooled file that is being processed by a writer might have a status of printing (PRT status), writer (WTR status), pending to be printed (PND status), or being sent (SND status). Spooled files with a status of PRT, WTR, PND, or SND are placed at the top of the output queue. A spooled file being processed by the writer might have a held (HLD) status if a user has held the spooled file but the writer is not yet finished processing the file. All other spooled files with a status of RDY are listed on the output queue after the file being processed by a writer, followed by deferred spooled files (DFR status), and then followed by spooled files with a status other than RDY or DFR.

Each group of spooled files (RDY and non-RDY files) is further sorted by:

1. The output priority of the spooled file.
2. A date and time field (time stamp).
3. The SCHEDULE parameter value of the spooled file. Files with SCHEDULE(\*JOBEND) specified are grouped together and placed after other spooled files of the same job that have SCHEDULE(\*IMMED) or SCHEDULE(\*FILEEND) specified.
4. The spool number of the spooled file.

For output queues with SEQ(\*JOBNBR) specified, the date and time field is the date and time that the job that created the spooled file entered the system. (A sequential job number and time of day value are also assigned to the job when it enters the system.) That is how the spooled files are sorted on the queue.

For first-in-first-out (\*FIFO) output queues, the date and time change to the current system date and time when:

- A spooled file is created by opening a device file.
- The output priority of the job that created the spooled file is changed.
- The status of the spooled file changes from non-RDY to RDY.

**Note:** The date and time do not change when the reason the status changes from RDY to WTR or from WTR to RDY is because the writer was canceled. Also, the date and time do not change when the status changes from RDY to DFR, or from DFR to RDY.

- A spooled file is moved to another output queue that has SEQ(\*FIFO) specified.

Because of the automatic sorting of spooled files, different results occur when SEQ(\*JOBNBR) is specified for an output queue than when SEQ(\*FIFO) is specified. For example, when a spooled file is held and then immediately released on an output queue with SEQ(\*JOBNBR) specified, the spooled file will end up where it started. However, if the same spooled file were held and then immediately released on an output queue with SEQ(\*FIFO) specified, the spooled file would be placed at the end of the spooled files that have the same priority and a status of RDY.

## Data queue support

There are two different types of data queue support for spooled files.

- **Data queue support on output queues**

Support is available to optionally associate a data queue with an output queue using the Create Output Queue (CRTOUTQ) or Change Output Queue (CHGOUTQ) command. Entries are logged in the data queue when spooled files are in ready (RDY) status on the output queue. A user program can determine when a spooled file is available on an output queue using the Receive Data Queue (QRCVDTAQ) API to receive information from a data queue. See the Receive Data Queue (QRCVDTAQ) API in the Programming topic for more information.

Each time a spooled file on the output queue reaches RDY status an entry is sent to the data queue. A spooled file can have several changes in status (for example, ready (RDY) to held (HLD) to release (RLS) to ready (RDY) again) before it is taken off the output queue. These status changes result in entries in the data queue for a spooled file each time the spooled file goes to RDY status.

A spooled file can reach RDY status in these instances:

- When initially spooled on the output queue.
- When the spooled file is opened and the schedule parameter value is \*IMMED.
- When a job completes and the spooled file schedule parameter value is \*JOBEND.
- When the spooled file is released.
- When a spooled file is moved to this output queue from another output queue.
- When a writer is ended immediately while printing a spooled file (the spooled file status is reset from WTR to RDY).

The data queue must be created with a maximum message length (MAXLEN) parameter value of at least 128 bytes. The sequence (SEQ) parameter value should be \*FIFO or \*LIFO. The format of the CRTDTAQ command is:

```
CRTDTAQ DTAQ (<library name>/<data queue name>) MAXLEN(128) SEQ(*LIFO)
```

The Create Output Queue (CRTOUTQ) and Change Output Queue (CHGOUTQ) commands have a data queue (DTAQ) parameter, which is used to specify the data name. An error occurs when using these commands if the specified data queue does not exist or if the user creating or changing the output queue does not have use authority to the data queue.

After a data queue is associated with an output queue, any spooled file that is placed on the output queue in a ready status causes an entry to be placed on the data queue. The data queue entry is added regardless of the authority the user generating the spooled file has to the data queue.

The entry in the data queue has a format of record type 01. See Record type 01 data queue entry format for a description of the information contained in record type 01.

- **Environment variable data queue support**

Using the Add Environment Variable (ADDENVVAR) or Change Environment Variable (CHGENVVAR) command, you can associate a data queue with a job or the system. As spooled files are created, the entries are logged in the data queue. Using the Receive Data Queue (QRCVDTAQ) API to receive information from the data queue, a user program can determine when a spooled file has

been created by the job or by the system. Use the environment variable data queue support if you need to determine the identity of a spooled file that was stored under a QPRTJOB and was created by a job such as a remote command system job.

Using the CL command ADDENVVAR, and specifying a fully qualified data queue name for the environment variable QIBM\_NOTIFY\_CRTSPLF, you can associate a data queue with a job or the system.

The command use is as follows:

```
ADDENVVAR ENVVAR(QIBM_NOTIFY_CRTSPLF)
          VALUE('*DTAQ <library name>/<data queue name>')
          LEVEL(*JOB | *sys)
```

The data queue must be created with a record length of at least 144 bytes. The data queue must also have a public authority of \*USE, or you need to grant the QSPL user profile \*USE private authority to the data queue. You must ensure that the containing library has a public authority of \*EXECUTE, or you need to grant the QSPL user profile \*EXECUTE private authority to the library. The format of the CRTDTAQ command is:

```
CRTDTAQ DTAQ (<library name>/<data queue name>) MAXLEN(144) AUT(*USE)
```

If you want the data queue information to have dates and time in Coordinated Universal Time (UTC), you can use the same environment variable with a different value. Replace the \*DTAQ with \*DTA2 in the ADDENVVAR command above. This value causes a type 03 data queue entry to be placed in the data queue instead of a type 02 entry. The command format is as follows:

```
ADDENVVAR ENVVAR(QIBM_NOTIFY_CRTSPLF)
          VALUE('*DTA2 <library name>/<data queue name>')
          LEVEL(*JOB | *sys)
```

The length of the type 03 data queue entry is 200 bytes. Use the following command format to create a type 03 data queue:

```
CRTDTAQ DTAQ (<library name>/<data queue name>) MAXLEN(200) AUT(*USE)
```

Once a data queue is associated with a job or the system, any spooled file created by the job or system will automatically have an entry placed in the data queue. For this action to occur, the user or user profile QSPL must have authorization to the data queue.

**Note:** An environment variable that is specified at the job level takes precedence over the same environment variable specified at the system level.

## Error conditions

An error occurs if the specified data queue does not exist or if the user creating or changing the output queue does not have use authority to the data queue.

After a data queue is associated with an output queue, any spooled file that is placed on the output queue in ready status causes an entry to be placed on the data queue. The data queue entry is added regardless of the authority the user generating the spooled file has to the data queue.

If the system tries to add entries to a data queue that does not exist or that has an invalid length, the system continues with its processing but sends an informational message to the QSYSOPR message queue. This message indicates that there is a problem with the data queue and specifies the data queue name. This message is sent the first time a specific problem occurs with the data queue of an output queue. The message is sent once every 24 hours.

For example, if message X is received at 10:00 a.m., it is logged in the QSYSOPR message queue. If message X is received again at 10:30 a.m., 11:00 a.m., 1:00 p.m., or 1:30 p.m., it will not be logged. As you can see, the message will not be logged until after 10:00 a.m. the next day, even if it continues to be received all day.

If after message X is logged at 10:00 a.m., message Y is received at 2:00 p.m., message Y is logged. If message X is received again at 2:30 p.m., message X will be logged again even though it was logged earlier in the day.

The intent is not to log the same recurring message all day, but to inform the user of each change of error messages associated with the data queue of a particular output queue.

### Additional considerations

Changing the data queue of an output queue is allowed regardless of whether there are spooled files on the output queue. For data queue entries of record type 01, only spooled files that reach RDY status after the change will have entries on the data queue. Spooled files already having a status of ready on the output queue will not have entries on the new data queue.

It is the user's responsibility to manage the data queues. These responsibilities include creating, clearing, and deleting data queues.

When clearing all output queues during IPL, any associated data queues are not cleared. If a damaged system output queue is found, it is re-created without any associated data queue name. Damaged data queues are not re-created.

### Related information:

Receive Data Queue (QRCVDTAQ) API

### Record type 01 data queue entry format:

The Record type 01 data queue entry format table shows the format of a 01 data queue entry when a spooled file changes to ready status on an output queue.

Table 1. Record type 01 data queue entry format

Decimal Offset	Hex Offset	Type	Description
0	0	CHAR(10)	Function  Identifies the function that created the data queue entry. The value for a spooled file is *SPOOL.
10	A	CHAR(2)	Record type  Identifies the record type within the function. Valid values are:  <b>01</b> A spooled file that is in READY status has been placed on the output queue.
12	C	CHAR(26)	Qualified job name  Identifies the qualified job name of the job that created the spooled file placed on the output queue.  <b>CHAR(10)</b> Job name  <b>CHAR(10)</b> User name  <b>CHAR(6)</b> Job number

Table 1. Record type 01 data queue entry format (continued)

Decimal Offset	Hex Offset	Type	Description
38	26	CHAR(10)	Spooled file name Identifies the name of the spooled file placed on the output queue.
48	30	BINARY(4)	Spooled file number Identifies the unique number of the spooled file placed on the output queue.
52	34	CHAR(20)	Qualified output queue name Identifies the qualified name of the output queue on which the spooled file was placed. <b>CHAR(10)</b> Output queue name <b>CHAR(10)</b> Library of the output queue
72	48	CHAR(8)	Job system name Identifies the name of the system on which the spooled file was generated.
80	50	CHAR(7)	Spooled file creation date Identifies the date on which the spooled file was created in CYYMMDD format (local system time).
87	57	CHAR(1)	Reserved
88	58	CHAR(6)	Spooled file creation time Identifies the time that the spooled file was created in HHMMSS format (local system time).
94	5E	CHAR(7)	Spooled file creation date in UTC Identifies the date on which the spooled file was created in CYYMMDD format.
101	65	CHAR(1)	Reserved
102	66	CHAR(6)	Spooled file creation time in UTC Identifies the time that the spooled file was created in HHMMSS format.
108	6C	CHAR(20)	Reserved

**Record type 02 data queue entry format:**

The Record type 02 data queue entry format table shows the format of a DTAQ entry for creating a spooled file.



Table 2. Record type 02 data queue entry format

Decimal Offset	Hex Offset	Type	Description
0	0	CHAR(10)	Function  Identifies the function that created the data queue entry. The value for a spooled file is *SPOOL.
10	A	CHAR(2)	Record type  Identifies the record type within the function. Valid values are: <b>02</b> A spooled file has been created and placed on the output queue.
12	C	CHAR(26)	Qualified job name  Identifies the qualified job name of the job that owns the spooled file placed on the output queue. <b>CHAR(10)</b> Job name <b>CHAR(10)</b> User name <b>CHAR(6)</b> Job number
38	26	CHAR(10)	Spooled file name  Identifies the name of the spooled file placed on the output queue.
48	30	BINARY(4)	Spooled file number  Identifies the unique number of the spooled file placed on the output queue.
52	34	CHAR(20)	Qualified output queue name  Identifies the qualified name of the output queue on which the spooled file was placed. <b>CHAR(10)</b> Output queue name <b>CHAR(10)</b> Library of the output queue
72	48	CHAR(26)	Creating qualified job name  Identifies the qualified job name of the job that created the spooled file. <b>CHAR(10)</b> Job name <b>CHAR(10)</b> User name <b>CHAR(6)</b> Job number

Table 2. Record type 02 data queue entry format (continued)

Decimal Offset	Hex Offset	Type	Description
98	62	CHAR(10)	User data Identifies the user specified data for the spooled file that was created.
108	6C	BINARY(4)	Auxiliary Storage Pool Identifies the Auxiliary Storage Pool where the spooled file was created.
112	70	CHAR(8)	Thread ID Identifies the thread of the job that created the spooled file.
120	78	CHAR(10)	System name Identifies the name of the system on which the spooled file was generated.
130	82	CHAR(7)	Creation date Identifies the date on which the spooled file was created in CYYMMDD format (local system time).
137	89	CHAR(6)	Creation time Identifies the time that the spooled file was created in HHMMSS format (local system time).
143	8F	CHAR(1)	Reserved

**Record type 03 data queue entry format:**

The Record type 03 data queue entry format table shows the format of a DTA2 entry for creating a spooled file.

Table 3. Record type 03 data queue entry format

Decimal Offset	Hex Offset	Type	Description
0	0	CHAR(10)	Function Identifies the function that created the data queue entry. The value for a spooled file is *SPOOL.
10	A	CHAR(2)	Record type Identifies the record type within the function. The valid value is: <b>03</b> A spooled file has been created and placed on the output queue.

Table 3. Record type 03 data queue entry format (continued)

Decimal Offset	Hex Offset	Type	Description
12	C	CHAR(26)	<p>Qualified job name</p> <p>Identifies the qualified job name of the job that owns the spooled file placed on the output queue.</p> <p><b>CHAR(10)</b> Job name</p> <p><b>CHAR(10)</b> User name</p> <p><b>CHAR(6)</b> Job number</p>
38	26	CHAR(10)	<p>Spooled file name</p> <p>Identifies the name of the spooled file placed on the output queue.</p>
48	30	BINARY(4)	<p>Spooled file number</p> <p>Identifies the unique number of the spooled file placed on the output queue.</p>
52	34	CHAR(20)	<p>Qualified output queue name</p> <p>Identifies the qualified name of the output queue on which the spooled file was placed.</p> <p><b>CHAR(10)</b> Output queue name</p> <p><b>CHAR(10)</b> Library of the output queue</p>
72	48	CHAR(26)	<p>Creating qualified job name</p> <p>Identifies the qualified job name of the job that created the spooled file.</p> <p><b>CHAR(10)</b> Job name</p> <p><b>CHAR(10)</b> User name</p> <p><b>CHAR(6)</b> Job number</p>
98	62	CHAR(10)	<p>User data</p> <p>Identifies the user-specified data for the spooled file that was created.</p>
108	6C	BINARY(4)	<p>Auxiliary storage pool</p> <p>Identifies the auxiliary storage pool where the spooled file was created.</p>
112	70	CHAR(8)	<p>Thread ID</p> <p>Identifies the thread of the job that created the spooled file.</p>

Table 3. Record type 03 data queue entry format (continued)

Decimal Offset	Hex Offset	Type	Description
120	78	CHAR(10)	System name Identifies the name of the system on which the spooled file was generated.
130	82	CHAR(7)	Creation date in UTC Identifies the date in UTC on which the spooled file was created in CYYMMDD format.
137	89	CHAR(6)	Creation time in UTC Identifies the time that the spooled file was created in HHMMSS format.
143	8F	CHAR(57)	Reserved

## Spooled file names

When spooled files are created, the spooled file name is typically the same as the name of the printer file that was used to create it. For example, if the Print key is pressed, the spooled file is called QSYSPRT, because QSYSPRT is the printer file used by the Print key operation.

There are several ways in which the spooled file can have a different name:

- The Override with Printer File (OVRPRTF) command was used and a name was specified in the SPLFNAME parameter. For example, typing the following command:

```
OVRPRTF QSYSPRT SPLFNAME(REPORT1)
```

causes the name of the spooled file to be REPORT1 instead of QSYSPRT.

- The OVRPRTF command was used and a different printer file is specified in the TOFILE parameter. For example, typing the following command:

```
OVRPRTF QSYSPRT TOFILE(PRTF2)
```

causes the spooled file to be called PRTF2 (the name of the printer file specified in the TOFILE parameter of the OVRPRTF command).

- Some IBM applications might create spooled files that have names different from the printer files used to create them. Users have no control over spooled file names in this situation.

## Spooled file security

Spooled security is primarily controlled through the output queue that contains the spooled files.

In general, there are five ways that a user can become authorized to control a spooled file (for example, hold or release the spooled file):

- User is assigned spool control authority (SPCAUT(\*SPLCTL)) in the user profile.

This authority gives a user control of all spooled files in the output queues of all libraries to which the user has \*EXECUTE authority. Only grant this authority to appropriate users.

- User is assigned job control authority (SPCAUT(\*JOBCTL)) in the user profile, the output queue is operator-controlled (OPRCTL(\*YES)), and the user has \*EXECUTE authority to the library that the output queue is in.

User is granted authority by the use of a spooled file security exit program. A spooled file security exit program can be registered for the QIBM\_QSP\_SECURITY exit point to allow more granular access to individual spooled files and to control access/operations to a spooled file for any user. A user can be granted or denied access to any spooled file for one or several operations. For more information about how to use a spooled file security registered exit program, see the Spooled File Security Exit Program topic.

| • User has the required object authority for the output queue. The required object authority is specified  
| by the AUTCHK parameter on the CRTOUTQ command. A value of \*OWNER indicates that only the  
| owner of the output queue is authorized to control all the spooled files on the output queue. A value  
| of \*DTAAUT indicates that users with \*CHANGE authority to the output queue are authorized to  
| control all the spooled files on the output queue. The use of a spooled file security exit program can  
| override this object authority and stop the user from controlling the spooled file.

| **Note:** The specific authorities required for \*DTAAUT are \*READ, \*ADD, and \*DLT data authorities.

| • A user is always allowed to control the spooled files created by that user unless not allowed by a  
| spooled file security exit program.

| For the Copy Spooled File (CPYSPLF), Display Spooled File (DSPSPLF), and Send Network Spooled File  
| (SNDNETSPLF) commands, in addition to the five ways already listed, there is an additional way a user  
| can be authorized.

| If DSPDTA(\*YES) was specified when the output queue was created, any user with \*USE authority to the  
| output queue is allowed to copy, display, send, or move spooled files assuming that the user is not  
| stopped by the use of a spooled file security exit program. The specific authority required is \*READ data  
| authority.

| If the user is authorized to control the file by one of the five ways already listed previously, using  
| DSPDTA(\*NO) when creating the output queue will not restrict the user from displaying, copying, or  
| sending the file. DSPDTA authority is only checked if the user is not otherwise authorized to the file.

| DSPDTA(\*OWNER) is more restrictive than DSPDTA(\*NO). If the output queue is created with  
| DSPDTA(\*OWNER), only the owner of the spooled file (the person who created it), a user with  
| SPCAUT(\*SPLCTL), or a user granted access by the use of a spooled file security exit program can  
| display, copy, or send a file on that queue. Even users with SPCAUT(\*JOBCTL) on an operator-controlled  
| (OPRCTL(\*YES)) output queue cannot display, copy, move, or send spooled files they do not own.

See the Security topic for details about the authority requirements for individual commands.

To place a spooled file on an output queue, one of the following authorities is required:

- Spool control authority (SPCAUT(\*SPLCTL)) in the user profile. The user must also have \*EXECUTE authority to the library that the output queue is in.

This authority gives a user control of all spooled files on the system and should only be granted to appropriate users. If you have spool control authority you can delete, move, hold, and release any spooled files on the system. You can also change the attributes of any spooled file.

- Job control authority (SPCAUT(\*JOBCTL)) in the user profile and the output queue is operator-controlled (OPRCTL(\*YES)). The user must also have \*EXECUTE authority to the library that the output queue is in.
- \*READ authority to the output queue. This authority can be given to the public by specifying AUT(\*USE) on the CRTOUTQ command.

#### **Related information:**

Security

Copy Spooled File (CPYSPLF) command

Display Spooled File (DSPSPLF) command

Send Network Spooled File (SNDNETSPLF) command

Spooled File Security Exit Program

### **Output queue security**

Output queues are created with a level of security determined by the value of the AUT parameter on the Create Output Queue (CRTOUTQ) command.

To work with the spooled files on that output queue, you must have the appropriate authority for that output queue (as specified in the AUT parameter). For example, holding or releasing a spooled file might require one level of authority while reading the contents of that spooled file might require a higher level of authority.

For more information about spooled file and output queue security, see the Security topic.

**Related information:**

Security

**QPRTJOB job**

A QPRTJOB job is a job that spooled files are associated with when the current job's user name is not the same as the user profile currently running.

System jobs can change to run under a user's profile in order for a user to obtain ownership of the spooled file instead of the system job. For example, if you send a spooled file using the Send Network Spooled File (SNDNETSPLF) command to user TINA on a different system, the file is spooled for job 999999/TINA/QPRTJOB. Spooling the file for this user's job instead of the system job makes sure that user TINA owns the spooled file. Then, when she runs the Work with Spooled Files (WRKSPLF) command, the spooled file sent to her is shown.

**Note:** Use the SPLFOWN parameter to specify who owns the spooled file.

QPRTJOB jobs are created automatically by the system. There can be more than one QPRTJOB per user on a system. A QPRTJOB has a default value of 9999 spooled files. That number can be expanded to the maximum of 999,999 by changing the number in the Maximum printer output files (QMAXSPLF) system value. For more information about the Maximum printer output files (QMAXSPLF) system value, see the Work Management topic. When a user's QPRTJOB gets full, the system automatically creates a new one for the user. A separate QPRTJOB is created for each user that receives spooled files sent by the SNDNETSPLF command. If you use the SNDNETSPLF command to send users TINA and KEVIN spooled files, there would be jobs named 999999/KEVIN/QPRTJOB and 999999/TINA/QPRTJOB on the receiving system.

QPRTJOB jobs are created and used by a variety of system functions. For example:

- Using the Send TCP/IP Spooled File (SNDTCPSPLF) or SNDNETSPLF commands to send a spooled file to another user on a different System i platform.
- Sending a spooled file from VM or MVS™ through a VM/MVS bridge to a System i platform.
- Receiving a spooled file using TCP/IP or the line printer daemon (LPD) process.
- Using the Create Spooled File (QSPCRTSP) Spool API to create a spooled file for another user.
- Using the Set Profile (QWTSETP) Security API to set the user profile to a different user and then create a new spooled file.

Other applications that are running can use the QSPCRTSP and QWTSETP APIs resulting in additional QPRTJOB jobs on the system.

- Using the UNIX SETGID API to create a spooled file for a different, current, or group user profile when SPLFOWN is set to \*CURGRPPRF.
- Using the UNIX SETUID API to set the user profile to a different user and then create a new spooled file for that user.

QPRTJOB jobs continue to be reused until they have been inactive more than 24 hours. Inactive means all spooled files for the job have been deleted and no new ones have been received for that user in more than 24 hours. The recovery is done by the system job QSPLMAINT.

**Related information:**

Work management

## Spooled file subsystem

The spooled file subsystem, QSPL, is used for processing the printer writer programs and must be active when printer writer programs are active. The spooled file subsystem and the individual printer writer programs can be controlled from jobs that run in other subsystems.

The Start Printer Writer (STRPRTWTR) command submits writer jobs to the job queue of the spooled file subsystem.

Requests for writer jobs are placed on the QSPL job queue, and the next entry on the QSPL job queue is selected to run if:

- The number of active jobs is less than the QSPL subsystem attribute of MAXJOBS.
- The number of active jobs from the QSPL job queue is less than the MAXACT attribute for the job queue.

## Spooled file library

The spooled file library (QSPL or QSPLxxxx, where xxxx is the number of the basic user ASP or primary independent ASP) contains database files that are used to store data for inline data files and spooled files. Each file in library QSPL or QSPLxxxx can have several members. Each member contains all the data for an inline data file or spooled file.

When the spooled file is printed or deleted, its associated database member in the spooling library is cleared of records, but not removed, so that it can be used for another inline data file or spooled file. If no database members are available in library QSPL or QSPLxxxx, then a member is automatically created.

Having some empty spooled file members available for creating new spooled files increases system run-time performance. However, a large number of empty spooled files can use large amounts of storage and decrease system abnormal IPL performance. For example, each spooled file member might take 24 KB of storage.

It is best to keep the QSPL or QSPLxxxx library small by periodically deleting old spooled files with the DLTSPFL or CLROUTQ commands. This procedure allows database members to be used again, rather than having to increase the size of the spooling library to accommodate new database members.

Displaying the data in the QSPL or QSPLxxxx library might prevent the data from being cleared, wasting storage space. Any command or program used to look at a database file in the QSPL or QSPLxxxx library must allocate the database file and member; if a writer tries to remove an allocated member after printing is completed, it will not be able to clear the member. Because the member is not cleared, it cannot be used for another inline data file or spooled file, and it will not be removed by setting the Automatically clean up unused printer output storage (QRCLSPLSTG) system value or running the RCLSPLSTG command.

Saving a database file in the QSPL or QSPLxxxx library can cause more problems than displaying the data in one member of the file because all members will be allocated a much longer time when a database file is saved. Because restoring these files destroys present and future spooled file data, there is no reason to save one of these files.

The QSPL or QSPLxxxx library type and authority should not be changed. The authority to the files within QSPL or QSPLxxxx should also not be changed. The QSPL or QSPLxxxx library and the files in it are created in a particular way so that system spooling functions can access them. Changing the library or files could cause some system spooling functions to work incorrectly and destroys the integrity of the spooled file security scheme.

**Related tasks:**

“Reclaiming spooled file storage” on page 218

You can use the Reclaim Spool Storage (RCLSPLSTG) command or the Automatically clean up unused printer output storage (QRCLSPLSTG) system value to reclaim spooled file storage. These are the only allowable ways to remove spooled database members from the QSPL or QSPLxxxx libraries. Any other way can cause severe problems.

## Spooled files in independent ASPs

Spooled files can be stored in output queues that are located in independent disk pools (also known as independent auxiliary storage pools or independent ASPs).

The creator of the spooled file must make sure the output queue to be selected is on the independent ASP that is wanted. This can be managed several ways such as through the printer file, job attribute, job description, or user profile.

The creator of the spooled file should make sure that a change of the name space (a set of libraries to which a job can resolve) does not occur during the creation of the spooled file. The printer writer must be started from a job with the independent ASP as part of its name space (job was started with INLASPGRP set to independent ASP or user has done SETASPGRP independent ASP to get independent ASP in its name space) in order for the printer writer to use that independent ASP and process the spooled files.

If the name space does change and the independent ASP on which the spooled file is being created gets varied off (this occurs if a name space change was made and the reservation on the independent ASP was gone), then put and close errors might occur. This might also cause data inaccuracies in spooled internal information. These inaccuracies can be fixed when the independent ASP is varied back on. Because the recovery of this condition is done in a background job, users might see some inconsistencies for those spooled files until the QSPRCxxxxx system job is able to finish the operation. If the independent ASP is not varied off, creation of the spooled file should be able to continue without any trouble.

The QSPMNxxxxx job has the following responsibilities:

- Clearing unused database members for deleted spooled files
- Automatic removal of database members that have not been reused within the days specified on the Automatically clean up unused printer output storage (QRCLSPLSTG) system value

The QSPRCxxxxx job has the following responsibility:

- Moving stranded spooled files onto output queue QSPRCLOUTQ in the QRCLxxxxx library in the primary ASP when a damaged output queue is deleted by a user

**Note:** There is one QSPMNxxxxx and one QSPRCxxxxx system job for each ASP group that is varied on.

If a printer writer or job ends abnormally, it might cause a spooled file or an output queue to become unusable or to be left in an unstable state where some operations are not permitted. In such cases, you can use the **Start Spool Reclaim (STRSPLRCL)** command to repair spooled files and output queues that are left in unrecoverable states. If the **STRSPLRCL** command fails to repair the spooled files and output queues, you need to vary off the independent ASP and then vary it back on again.

Spooled files placed into an independent ASP are automatically detached from the job when the job ends and no spooled files for the job reside in the system or basic user ASPs. You should ensure that all applications make use of the spooled file identity values JOBSYSNAME and CRTDATE, including specific date and time, to prevent duplicate spooled file or duplicate job error messages. Note that when an independent ASP is moved from system A to system B on a fail-over, the spooled files no longer have the original jobs available to them (the spooled files have been detached from the job). When the spooled files are detached, there is no operating system protection from another job starting with same identity as a job that ran on system A.



An independent ASP cannot be varied on if it contains a spooled file that already exists on the system or basic user disk pools (\*SYSBAS). For more information, see Printing considerations in the Disk management topic collection.

For more information about working with independent ASPs, see the Using independent disk pools topic.

The following CL commands have spooled file in independent ASP limitations related to S/36 support, operational assistant support, and library name space.

- Change Job (CHGJOB) command
- Change Writer (CHGWTR) command
- Copy Spooled File (CPYSPLF) command
- Hold Job (HLDJOB) command
- Reclaim Spool Storage (RCLSPLSTG) command
- Release Job (RLSJOB) command
- Work with Job (WRKJOB) command
- Work with Spooled Files (WRKSPLF) command

**Related tasks:**

“Reclaiming spooled file storage” on page 218

You can use the Reclaim Spool Storage (RCLSPLSTG) command or the Automatically clean up unused printer output storage (QRCLSPLSTG) system value to reclaim spooled file storage. These are the only allowable ways to remove spooled database members from the QSPL or QSPLxxxx libraries. Any other way can cause severe problems.

**Related information:**

Printing considerations

Independent disk pool examples

Change Job (CHGJOB) command

Change Writer (CHGWTR) command

Copy Spooled File (CPYSPLF) command

Hold Job (HLDJOB) command

Reclaim Spool Storage (RCLSPLSTG) command

Release Job (RLSJOB) command

Work with Job (WRKJOB) command

Work with Spooled Files (WRKSPLF) command

Start Spool Reclaim (STRSPLRCL) command

## Printer writer program

The *printer writer program* is a system-supplied program that takes spooled files from an output queue and sends them to a printer. The spooled files on a particular output queue remain stored in the system until the printer writer program assigns a printer to the output queue.

The printer writer program takes spooled files, one at a time, from the output queue, based on their priority. The printer writer program prints a spooled file only if its entry on the output queue indicates that it has a ready (RDY) status. You can display the status of a particular spooled file using the Work with Output Queue (WRKOUTQ) command.

Do not confuse the printer writer program with an actual printer device or a printer file. The printer writer program is the program that allows you to assign an actual printer device to an output queue and select spooled files from the output queue to be printed. The Start Printer Writer (STRPRTWTR) command and the Work with Writers (WRKWTR) command provide you with the ability to assign any configured printer to any output queue. Even though the name of the command (Work with Writers)

indicates you are working with printer writers, you are actually using the printer writer program to make a match between an output queue and a physical printer.

If the spooled file has a ready status, the printer writer program takes the entry from the output queue and prints the specified job or file separators or both, followed by the output data in the spooled file. If the spooled file does not have a ready status, the printer writer program leaves the entry on the output queue and goes on to the next entry. In most cases the printer writer program continues to print spooled files (preceded by job and file separators) until all spooled files with a ready status have been taken from the output queue.

**Notes:**

1. The printer writer program uses the printer file QPSPLPRT. This printer file is included in the system. It is set up for the printer writer program and should not be changed or used for other applications.
2. If you run the CHGPRTF command to make all the IBM-supplied printer files DBCS capable (CHGPRTF FILE(\*all/\*all) IGCDTA(\*YES)), you must change the IGCDTA parameter value for QPSPLPRT printer file back to \*NO.

If your printer stops for any reason (out of paper, for example), the system does not automatically assign a different printer to continue printing the jobs in the output queue that your printer was assigned to. You must manually assign another printer to that output queue.

Multiple printer writers can be started to one output queue. The limit is 10. This support allows many printers (up to 10) to begin printing spooled files from the same output queue.

The multiple printer writer function supports work load balancing between printers. It also provides backup for printed jobs running unattended. For example, if one printer jams or runs out of paper the others continue to print spooled files from the associated output queue.

A remote writer is an i5/OS program that takes spooled files from a remote output queue and sends them to the specified remote system. The remote writer, which is a system job, sends the spooled files using SNADS or TCP/IP. This function is known as remote system printing on the System i platform. The Start Remote Writer (STRRMTWTR) command is used to initiate remote system printing.

After the spooled file is successfully sent to a remote system, it is deleted or saved as determined by the SAVE spooled file attribute value.

More than one remote writer can be started to the same remote output queue (10 is the limit). The actual number is specified in the remote output queue description. However, each writer name must be unique and of the same type (printer, remote, or diskette).

**Related concepts:**

“Remote system printing” on page 57

Remote system printing allows spooled files that are created on a System i platform to be automatically sent to and printed on other systems.

## **Printer data streams**

The IBM i operating system supports different data streams, such as SNA character string (SCS), Advanced Function Presentation data stream, Intelligent Printer Data Stream (IPDS), and American National Standard Code for International Interchange (ASCII) data stream.

### **SNA character string**

The SNA character string (SCS) has a relatively simple structure, consisting of a 1-byte hexadecimal control code followed by the data to be printed. Multiple control codes can be inserted into SCS by printing applications in order to achieve certain types of output. The SNA character string is sent to the printer in physical blocks of 256 bytes.

Here are a few examples of SCS control codes:

Hex	Description	Code
03	ASCII transparent data	(TRNA)
05	Horizontal tab	(HT)
0B	Vertical tab	(VT)
0C	Forms feed	(FF)
0D	Carriage return	(CR)
1A	Unit backspace	(UBS)
15	New line	(NL)
16	Backspace	(BS)
35	Transparent	(TRN)
2843	Set attribute	(SA)
2BC1	Set horizontal format	(SHF)
2BC2	Set vertical format	(SVF)
2BC6	Set line density	(SLD)
2BD2	Set presentation page size	(SPPS)
2BD4	Begin underscore	(BUS)
2BFE	Load alternate character	(LAC)

### How print attributes are accomplished by SCS

The following examples show how common print functions can be accomplished by printing applications using SCS.

#### Line spacing

The equivalent number of NL (new line) control codes is inserted.

#### Underlining

The underlined text is first inserted into SCS, followed by the appropriate number of BS (backspace) control codes, and then the same number of underscores.

#### Overstriking

This is similar to underlining, except that the overstrike characters are used instead of the underscore.

#### Emphasis

The text to be emphasized is put into SCS, followed by the necessary number of BS (backspace) control codes and then the emphasized text is repeated. This sequence might be repeated several times.

#### Page eject

The FF (forms feed) control code is inserted.

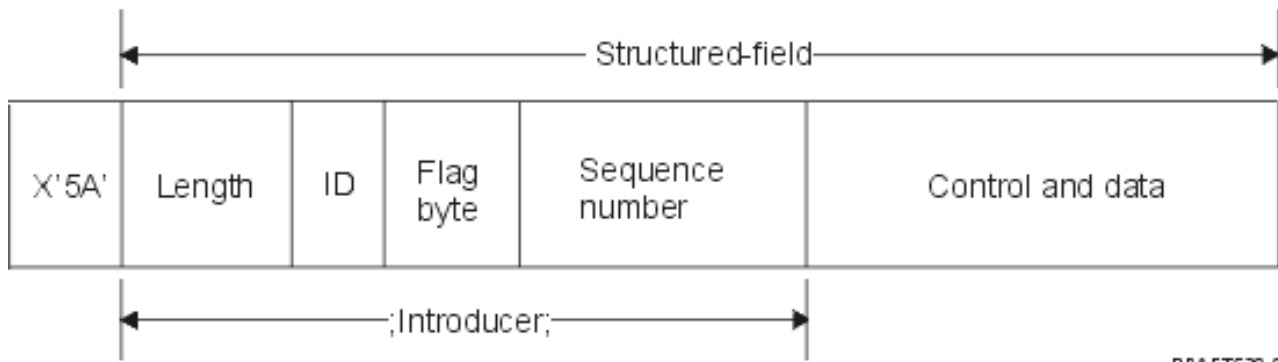
#### Font changes

Font changes are supported by printers such as the 5219 that support final-form text (FFT).

### Advanced Function Presentation data stream

Advanced Function Presentation data stream is the application interface to Advanced Function Presentation (AFP) based on the Mixed Object:Document Content Architecture–Presentation (MO:DCA–P). It is independent of both printers and operating systems.

AFP data stream is a structured data stream. Structured fields are used to define composed text pages, line format data, and a mixture of line and composed text data. The structured field is a self-identifying string of bytes containing data or parameters and must have an introducer, which contains a length field, an identifier, a flag, and a sequence number. This is followed by parameter bytes that contain control information or data to be printed.



RRAF5301-1

### **Length**

A 2-byte field that specifies the length of the record (not including the 5A control character).

### **Identifier (ID)**

A 3-byte field that specifies the type of the structured field.

### **Flag byte**

A 1-byte field that specifies information about the data field.

### **Sequence number**

A 2-byte field that identifies the record.

### **Control and data**

A text control code, the name of an object, or coordinates for positioning an image or page segment. The control information is followed by the data to be printed.

### **Source programs that generate Advanced Function Presentation data stream**

The following IBM licensed programs generate Advanced Function Presentation data streams:

- i5/OS
- Advanced Function Presentation Utilities for i5/OS (AFP Utilities for i5/OS)
- Document Composition Facility (DCF)
- DisplayWrite/390 (DW/390)
- Graphical Data Display Manager (GDDM)
- Advanced Function Presentation Utilities for z/OS®:
  - Page Printer Formatting Aid (PPFA)
  - Overlay Generation Language (OGL)
  - Font Library Service Facility (FLSF)
  - Print Management Facility (PMF)
  - Print Service Access Facility (PSAF)

### **Advanced Function Presentation**

Advanced Function Presentation data stream describes what the page of data looks like and refers to printer resource objects by name, assuming that they are resident on the system where the actual printing is to take place and not on the system where the output was spooled. The printer resource objects that are resident on the system are downloaded to the printer by the processor when needed.

The IBM i operating system uses the following objects when processing the AFP data stream:

**AFP data stream spool buffers**

The AFP data stream is sent to the system in one or more buffers. These buffers contain single or multiple contiguous structured fields. A pointer to the buffer and the length of each buffer are passed to the system.

**AFP data stream resource objects**

Resource objects contain data and control information that can be used in printing a job. These can be shared by different pages in the same job. A resource is composed entirely of structured fields.

Types of resources are:

- Fonts
- Form definitions
- Page segments
- Overlays
- Page definitions
- Resources stored in the integrated file system

These resources can be transmitted from a host System z<sup>®</sup> to a System i product or loaded from tape into space on the system using i5/OS commands. Overlays and page segments can be created by AFP Utilities for i5/OS.

**Messages**

Messages generated during the processing of AFP data stream spooled files are placed in the printer writer job log.

**Mixed Object: Document Content Architecture (MO:DCA)**

Mixed Object Document Content Architecture (MO:DCA) is an object-oriented data stream that is designed to allow consistent printing that is independent of operating system or device. An object is a collection of data that can be treated as a unit, such as text, images, and graphics. This architecture supports:

- Coexistence and migration of existing IBM document architecture and printer data streams
- Device independence
- Separation of functions to simplify transformation of objects into other data streams
- National Language Support
- Office Document Architecture (ODA) support
- Standard Generalized Markup Language (SGML)

The data stream for an MO:DCA document consists of various objects as well as the logical and layout structure of the document. The logical structure defines the logical content of the document—chapters, figures, and lists. The following layout structure defines the way the data should be presented.

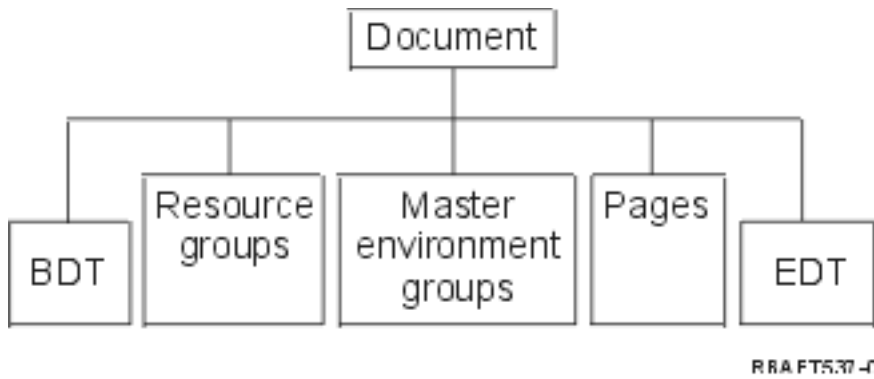


Figure 1. MO:DCA Document Structure

**BDT (Begin document)**

Indicates the beginning of the document

**Resource groups**

Specifies fonts, overlays, and segments so that these objects can be transmitted as part of the data stream. They can be referred to by an MO:DCA Include structured field.

**Master environment groups**

Specifies the processing environment, such as space definitions, suppression of data, number of copies, and internal data stream references.

**Pages** Contains objects that are part of the document. These objects can be text, graphics, and images.

**EDT (End document)**

Indicates the end of the document.

The following different types of objects make up MO:DCA. All of these objects are supported by IPDS:

- Bar Code Object Content Architecture (BCOCA)
- Image Object Content Architecture (IOCA)
- Graphics Object Content Architecture (GOCA)
- Presentation Text Object Content Architecture (PTOCA)
- Font Object Content Architecture (FOCA).

**Bar Code Object Content Architecture (BCOCA)**

A bar-code object can contain “draw rule” commands or raster data, depending on whether the bar code is to be drawn as a graphics object or has been scanned into the data stream as an image. A bar code object containing draw rule commands is built up using only lines of a specified length and width. A graphics object is constructed from a number of primitives, such as lines, arcs, symbols, shaded areas, and point arrays.

**Image Object Content Architecture (IOCA)**

IOCA represents images in device-independent format. A standard set of constructs has been defined to describe the image data, the characteristics of that data, and manipulation functions that can be performed on the data. The image content is inserted in an image segment.

**Graphics Object Content Architecture (GOCA)**

GOCA describes complex pictures. These pictures are formed from a collection of primitives, such as lines, arcs, characters, symbols, and shaded areas or point arrays. Each of these primitives has its own set of attributes, such as line width, orientation, and resolution. In addition to these attributes, there is a set of general drawing attributes like color, which apply to all primitives.

## **Presentation Text Object Content Architecture (PTOCA)**

PTOCA describes the text part of a document. The presentation text object, in common with the other objects, is designed not only to be carried by, but to be an integral part of, the data stream, providing the following:

- Structured field introducer and syntax for the structured field
- Begin/end object structure
- Control of alternate action selection for error recovery
- Passing of exception conditions back to the originating process
- Initial state of the object
- Relationship of presentation text objects to other objects contained in the data stream.

Two structured fields provide the necessary presentation information to the printer:

### **P T descriptor structured field**

Defines several positional parameters for the object

### **P T data structured field**

Contains the presentation text and the control sequences for positioning graphic characters. These graphic characters are defined within the coded fonts.

## **Font Object Content Architecture (FOCA)**

In order to achieve uniform document presentation output, it is essential that font resources be consistently defined and put into effect. These resources must be identified by means of a constant, unvarying set of parameters.

FOCA makes it possible to achieve the required degree of consistency by defining:

- A common font and character definition model that can be used by all products and architectures as the basis for font applications
- A composite set of parameters specific to a font resource and references to that resource
- A device-and-technology-independent method of defining font measurements
- The specification of formats for conveying font information to suit the application

FOCA defines the parameter content of:

- IBM font resources
- References to the font resources
- Information accessed by the font resources

## **Intelligent Printer Data Stream**

Intelligent Printer Data Stream (IPDS) is IBM's host-to-printer data stream for Advanced Function Presentation subsystems. It provides an interface to all-points-addressable (APA) printers that makes possible the presentation of pages containing an architecturally unlimited mixture of different data types: high-quality text, raster image, vector graphics, and bar code.

IPDS incorporates the following functions:

- Different applications can create source data (graphics, image, bar code, and text), independently of one another. IPDS makes it possible for the output of these independent applications to be merged at print time resulting in an integrated mixed data page.

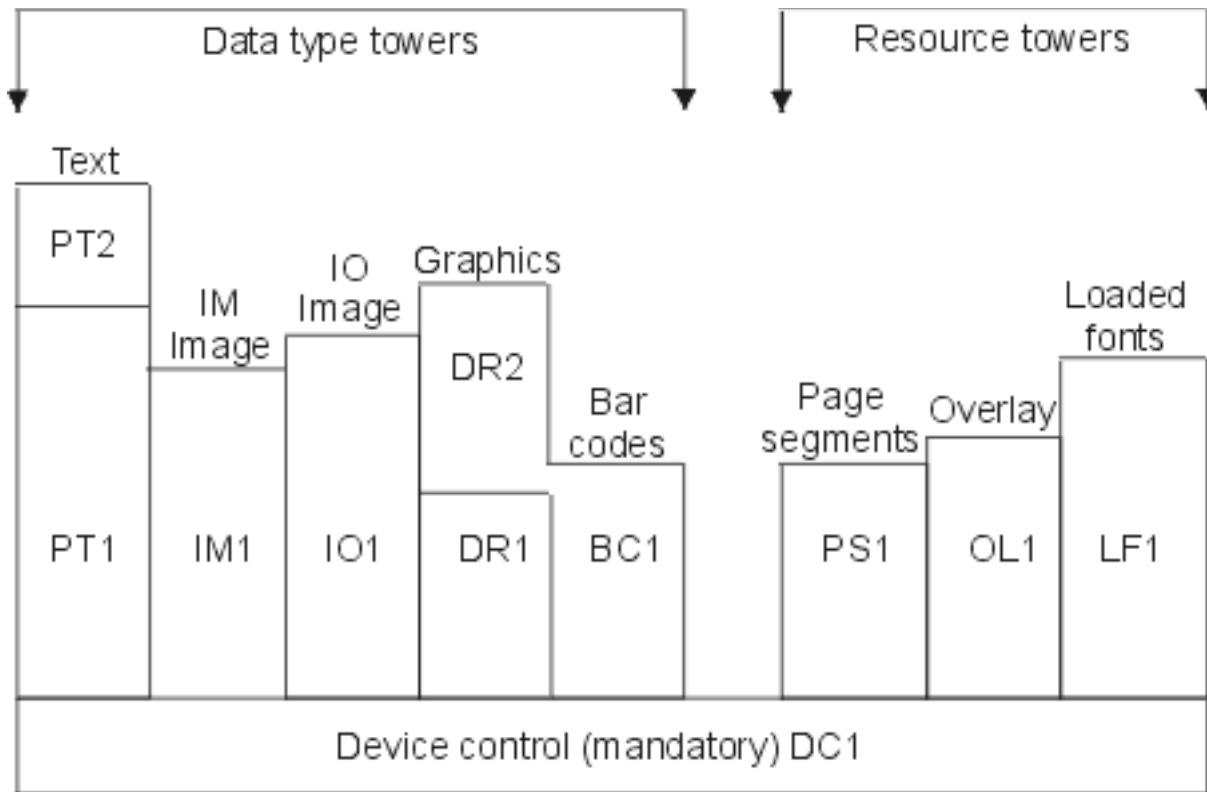
IPDS makes this possible by carrying independently defined blocks of data (*objects*). The IBM 3270 display data stream also carries similarly defined independent objects, thus making it possible to use the same objects in both environments.

- IPDS is independent of the carrying communications protocol. This allows the transmission of the same data stream to channel-attached printers, controllers, local area networks, and any other networking link supporting transparent transmission of data.

- IPDS transfers all data and commands through self-identifying structured fields which describe the presentation of the page and provide for these:
  - Dynamic management of downloaded resources (overlays, page segments, and loaded fonts) and resident fonts
  - Control of device functions such as duplexing, media bin selection, and output finishing
  - Comprehensive handling of exception functions, enabling users to control the level of exception handling
- IPDS provides an extensive acknowledgement protocol at the data stream level. This acknowledgement protocol helps synchronize host and printer processes, exchange query/reply information, and return detailed exception information.

**IPDS functional divisions**

IPDS architecture is divided into several functional areas, each of which contains a set of IPDS commands representing a major printer capability. This function set design allows IPDS to support a wide range of printer products. Product developers can match function set implementations to the specific needs of their product.



RRA FT531-0

**Device control**

Function set is composed of the IPDS commands that set up a page, communicate device controls, and manage acknowledgement protocol. The device-control function set is the only mandatory function set for IPDS printers, although not all DC1 orders are required to be supported.

**Text**

Function set composed of commands and orders (text controls) required to present text information on a page, page segment (stored command sequence), or an overlay (electronic form). The text function set contains two presentation text (PT) subsets: PT1 and PT2. Text printers support either subset. PT2 is a superset of PT1 and, therefore, all orders contained in PT1 are also in PT2.



**IM image**

Function set containing the IPDS commands required to present raster image data on a page, page segment, or an overlay.

**IO image**

Function set containing commands that present raster data (similar to IM image) but with additional functions.

**Graphics**

Function set composed of the IPDS commands and drawing orders required to present vector graphics on a page, page segment, or overlay. The graphics function set contains two drawing (DR) subsets: DR1 and DR2. Vector graphics printers support DR2, which is a superset of DR1.

**Bar code**

Function set composed of the IPDS commands required to present machine-readable bar code information on a page, page segment, or overlay.

**Page segments and overlays**

Function sets composed of the IPDS commands required to store and present IPDS constructs containing text, graphics, image, and bar code information. These stored constructs can be either page segments or overlays.

**Loaded font**

Function set composed of the IPDS commands necessary to load and delete font information.

**Function-set requirements for IPDS**

In order to claim support of the IPDS architecture, a product must meet the following requirements:

- Accomplish all required commands in the device-control function set
- Accomplish at least one subset of one other data function set
- Accomplish all required commands, orders, and controls for each supported function set or subset

**Return of function-set information**

A host presentation services program determines the functional capabilities of an IPDS printer by issuing certain IPDS query commands to the printer and requesting an acknowledgement. The data that the printer returns in the acknowledge reply shows the printer type and model, details of the function sets supported, and a variety of printer characteristics.

**The IPDS page environment**

IPDS creates mixed-data pages within a hierarchy of presentation spaces. These presentation spaces are: physical page, logical page, and data blocks.

**Physical page**

The medium (typically paper) on which information is placed. The physical page has boundaries of width and depth that define the limits of the medium.

**Logical page**

The electronic representation of the page that is sent to the printer. The logical page is a rectangular area that might not be the same size as the physical page on which it is placed. Printing can only occur where the current logical page intersects the physical page (valid printable area).

**Data blocks**

Rectangular areas positioned on the logical page. Data blocks can be one of three types:

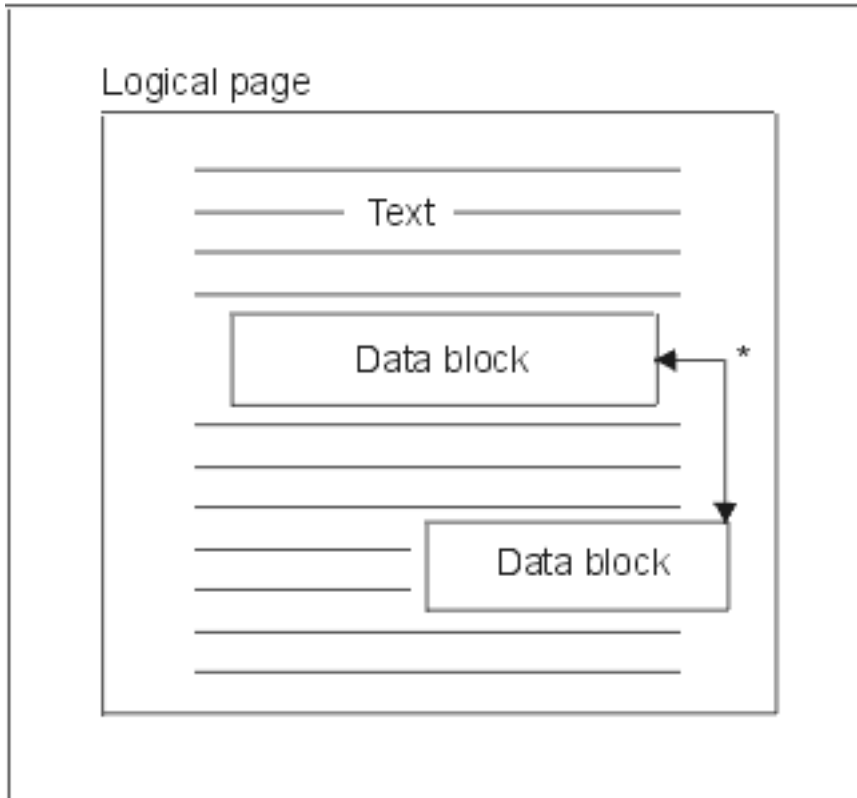
- Image Data Blocks: Raster information
- Graphics Data Blocks: Lines, curves, areas, and other drawing elements

- Bar Code Data Blocks: Bar-coded, machine-readable characters or human-readable characters.

**Note:** There is no text data block. Text can be positioned anywhere in the valid printable area. Data blocks can be positioned in relation to the text.

IPDS has a hierarchical relationship between a physical page, a logical page, and data blocks. In the following figure, the logical page boundaries do not correspond to the physical page boundaries.

### Physical page



\* Independent blocks of image, graphics, or bar code data

RR AFT5.32-C

One of the strengths of IPDS is that independent applications can create source data for each data block. The output of these independent applications is merged at the printer to create an integrated mixed data page. For example, text data can be produced on an editor, image data can be the output of a scanner stored in a folder, and graphics data can be produced by Business Graphics Utility. IPDS makes it possible to integrate application output rather than requiring the use of integrated applications.

### Overlays and page segments

IPDS stores resources in the printer for later use. Overlays and page segments can be merged with the logical page before the logical page is printed on the physical page.

#### Overlay

Macro-like construct loaded by the host processor and sent to the printer's storage. An overlay can consist of any combination of text data, image block data, graphics block data, or bar code block data. An overlay contains the same type of presentation commands used in the logical page, but overlays are independent of the logical page environment. The major difference between overlays and logical pages is that overlays are stored until deleted but logical pages, if stored, are only stored until printed. Overlays are often used as electronic forms.

### Page segment

This is like the overlay in makeup. The difference between a page segment and an overlay is that page segments are not independent of the page environment; they are merged with the logical page and assume the currently active environment.

### Loaded fonts

A font is a set of characters in a particular type style and size. Fonts can be downloaded from the host or can be resident in printer storage. Downloaded fonts are called *loaded fonts* and come in one of the following configurations:

#### Coded font

A complete code page of graphic characters in a particular style. (A code page maps each character in a font to a numeric value or code point.)

#### Symbol set

A set of characters simpler in structure than a coded font. Symbol sets are used where typographic quality is not required. Many dot matrix printers and displays use symbol sets.

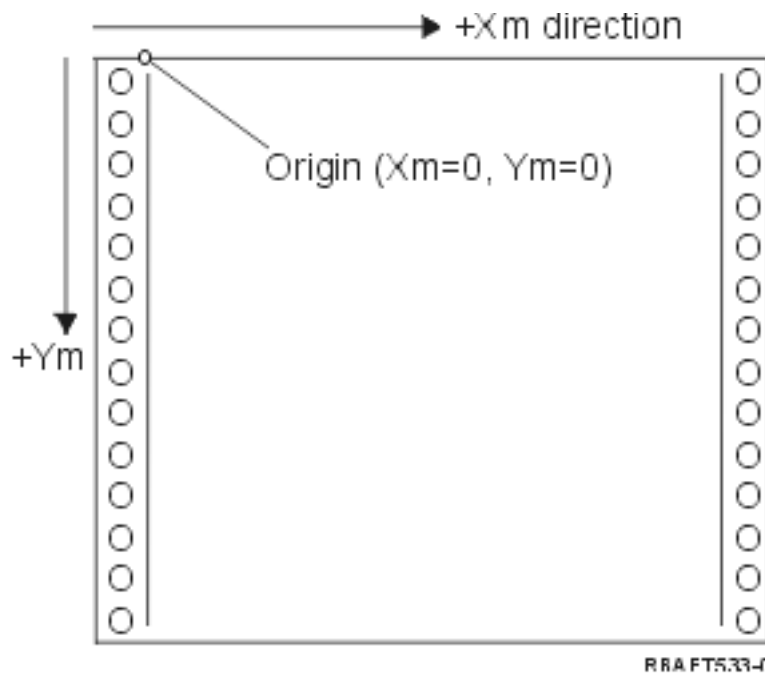
### Coordinate systems

#### X<sub>m</sub>, Y<sub>m</sub> coordinate system (physical page)

IPDS uses orthogonal coordinate systems to define any point on a page. Distances between these coordinate systems are measured in logical units or *L-units* rather than physical pels.

The X<sub>m</sub>, Y<sub>m</sub> coordinate system is the physical-page coordinate system. The top-left corner of the physical page is always (0,0).

The printer defines the top of a physical page.



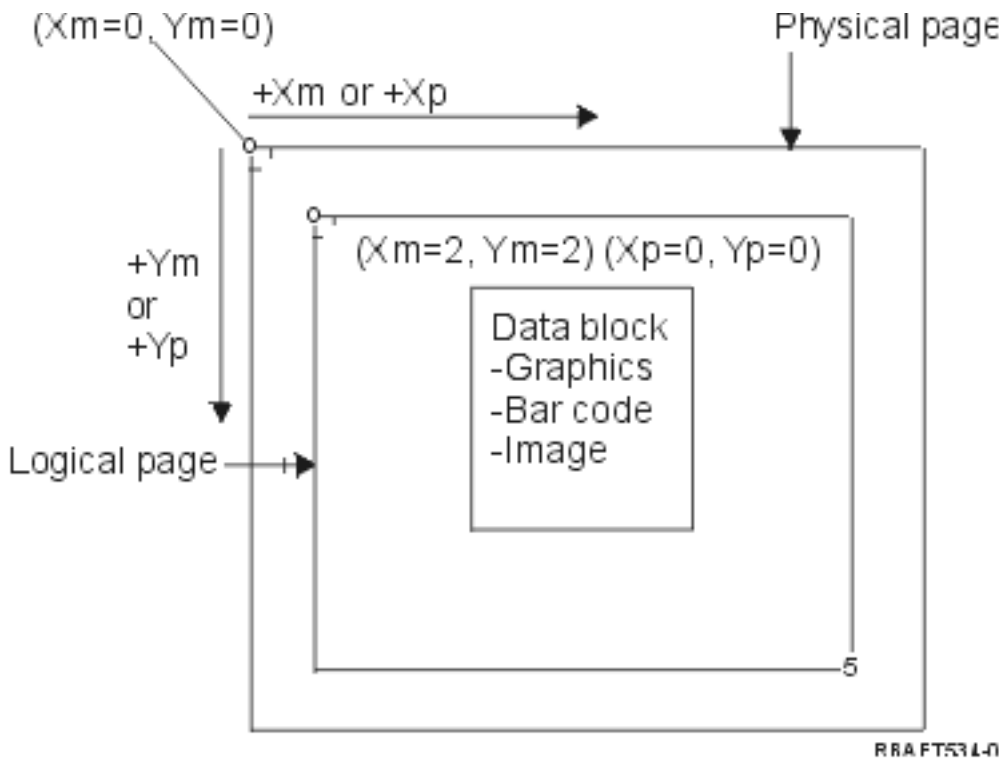
The X<sub>m</sub>, Y<sub>m</sub> coordinate system is fixed for each media size. IPDS commands cannot change the orientation of these coordinates.

#### X<sub>p</sub>, Y<sub>p</sub> coordinate system (logical page)

The  $X_p$ ,  $Y_p$  coordinate system is the logical-page coordinate system. The origin of this system ( $X_p=0$ ,  $Y_p=0$ ) is specified as an offset from the physical page origin ( $X_m=0$ ,  $Y_m=0$ ) through the Load Page Position command. IPDS commands cannot change the orientation of the  $X_p$ ,  $Y_p$  coordinate system; it is always parallel to, but offset from, the  $X_m$ ,  $Y_m$  coordinate system.

The size of the logical page in the  $X_p$  dimension is called the  $X_p$  extent. The size of the logical page in the  $Y_p$  dimension is called the  $Y_p$  extent. The  $X_p$ ,  $Y_p$  coordinate system is used to locate data blocks on the logical page.

The coordinate system for overlays is the same as the  $X_p$ ,  $Y_p$  coordinate system for logical pages.



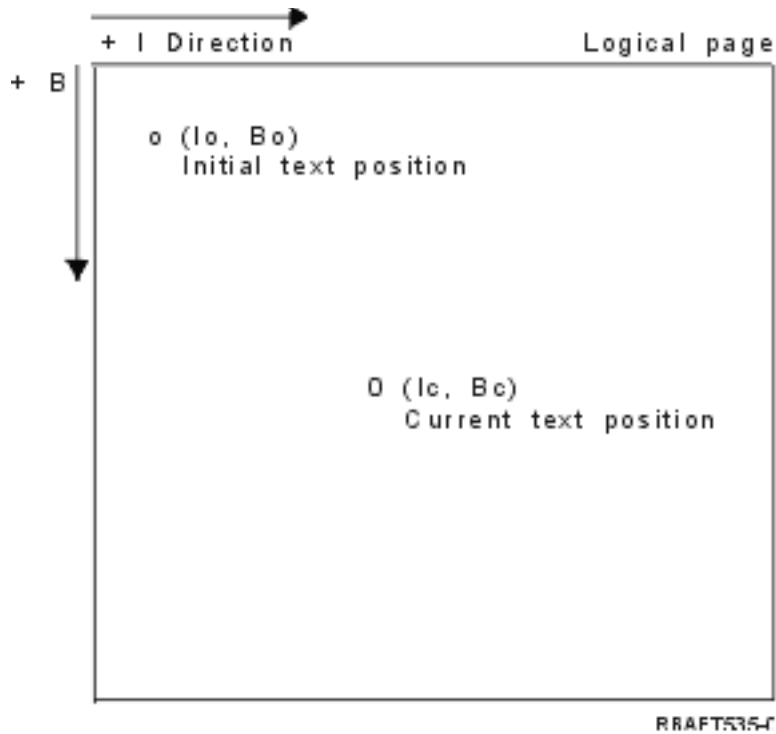
### I, B Coordinate System (Text)

The Inline, Baseline (I, B) coordinate system describes the placement of data blocks on the logical page. The printer places characters along the I-axis to form a line of text and places lines of text along the B-axis on the logical page. IPDS commands can change both the origin and the orientation of the inline and baseline axes.

As characters are developed on the page, the inline coordinate is incremented in the **positive inline** (or +I) direction. As lines are developed on the page, the baseline coordinate is incremented in the **positive baseline** (or +B) direction.

**Note:** Characters are developed on a page in the direction in which they will be read (left to right, for example). The printer can actually place characters or lines on a page in various directions (as in bidirectional printing).

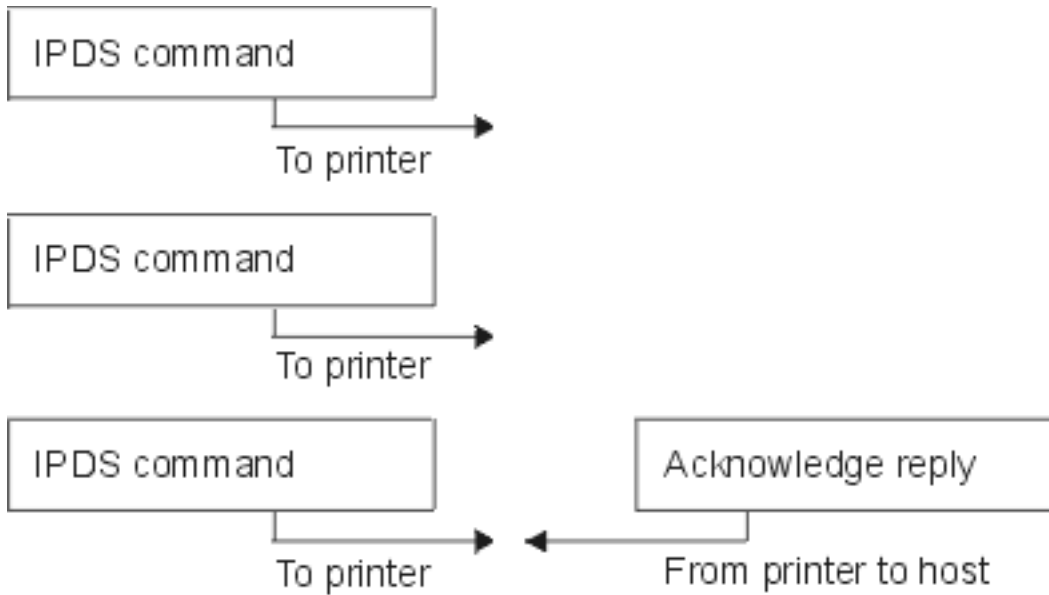
The coordinates of the first text position on the logical page are called the initial inline text coordinate ( $I_0$ ) and the initial baseline text coordinate ( $B_0$ ). The coordinates of the current position on the logical page are called the current inline text coordinate ( $I_c$ ) and the current baseline text coordinate ( $B_c$ ).



### Processing IPDS commands

The structured field format of IPDS allows commands to be sent to the printer in a continuous stream. Each command is self-describing. The command length, identifier, flag byte, and data (not always present) are all part of each command. The printer-host conversation is carried on as if IPDS commands were processed in sequential order by the printer.

Every IPDS command contains a flag byte. The setting on the acknowledgement-required bit on this flag byte indicates the end of a command sequence to the printer. The printer then sends an acknowledge reply to the host, as illustrated in the following diagram:



RAFT536-1

### The IPDS command format

All IPDS commands are encoded in the following format:

Length	Command	Flag	CID	Data
--------	---------	------	-----	------

#### Length

A 2-byte field that specifies the length of the command. This count includes itself, the command field, the flag byte and the optional correlation ID (CID), and data fields. The length field can range from X'0005' to X'7FFF'.

#### Command

A 2-byte field that specifies the IPDS command.

**Flag** A 1-byte field that contains the IPDS command stream flags.

- Bit 0 is the acknowledgement required (ARQ) flag. If this bit is on, the host requests the printer to send an acknowledge reply.
- Bit 1 is the correlation ID (CID) flag. If it is on, a 2-byte correlation ID follows. If it is off, the CID is not present and the following bytes (if any) contain the data field.

#### CID (correlation ID)

A 2-byte field that specifies an identifier for the command. A presentation services program can use any value between X'0000' and X'FFFF' for the correlation ID.

**Data** Not present for all commands. If present, it contains specific orders, parameters, and data appropriate for the given command.

### IPDS operating states

IPDS commands are defined within the context of printer operating states. The printer moves between these operating states during command processing. IPDS printers are *state machines* with the following operating states:

- Home state
- Block state

- IO image block state
- IM image block state
- Graphics block state
- Bar code block state.
- Page state
- Overlay state
- Page segment state
- Font state
- Any-state

### **Home state**

The initial IPDS operating state. The printer returns to home state at the end of each downloaded page, page segment, coded font, or overlay.

While in home state, the printer receives control and initialization commands to prepare for the print operation. In home state, the printer can also receive commands that delete resources or request the return of printer information to the host presentation services program.

### **Block states**

State for establishing the initial processing conditions for a block of data and placing the block of data on the logical page, page segment or overlay. The printer can only enter a block state from page, page segment, or overlay states.

### **Page state**

The operating state for printing a logical page. The printer enters page state from home state on receiving a Begin Page command and exits on receiving an End Page command.

In page state, the printer can receive commands that merge previously defined and loaded overlays and page segments with the current page information. The printer can also receive Write Text commands that position text on the logical page and can enter a block state to write image, bar code, and graphics blocks.

### **Overlay state**

State that allows overlay data to be stored in the printer. The printer enters overlay state from home state on receiving a Begin Overlay command and exits on receiving an End Page command.

In overlay state, the printer can receive commands that merge previously defined and loaded overlays and page segments with the current page information. The printer can also receive Write Text commands that position text on the logical page and can enter a block state to write image, bar code, and graphics blocks.

### **Page segment state**

State that allows page segment data to be stored in the printer. The printer enters page segment state from home state on receiving a Begin Page Segment command and exits on an End Page command.

In page segment state, the printer can receive Write Text commands that position text on the logical page and can enter a block state to write image, bar code, and graphics blocks.

### **Font state**

State that allows the printer to receive downloaded coded-font data. The printer enters font state from home state on receiving a Load Font Control command.

While the printer is in font state, the Load Font command can send coded-font, character-raster pattern data to the printer. Receipt of an End command returns the printer to home state.

### **Any-state**

Some IPDS commands can be received in any IPDS operating state. These commands do not change the IPDS operating state, with the exception of XOA Discard Buffered Data.

## Default handling

Defaults are values used as control parameters when no other values are specified in the current command. IPDS defaults are called through omission or through values transmitted in the data field portion of commands. The IPDS default structure is normally hierarchical. General IPDS default rules are:

- If power has been interrupted or if the printer has been initialized, printer-established page default values are used until specific IPDS default values are received.
- Initial page values are established when the printer receives a Load Page Descriptor command. If no such command is received, printer-established default values remain in effect.
- Initial data block values are established when the printer receives either a Write Image Control, Write Image Control 2, Write Bar Code Control, or Write Graphics Control command. These values remain in effect until data controls override them or until the printer receives an End command that ends the block.

## American National Standard Code for Information Interchange

There is no formal structure controlling the use of the American National Standard Code for Information Interchange (ASCII) data stream to control printers attached to systems providing ASCII support. Control of page printers, like the IBM 3812, is done by using page map primitives (PMPs), which are a set of commands or basic instruction set of these printers when attached in ASCII mode.

ASCII data sent to a page printer is translated into PMPs. The page printer composes the page of data in its internal memory or page map. Two page orientations (portrait and landscape) as well as four print directions are supported. Complexity of the printed data is determined by the application print program, which can set the pels on explicitly in the page set, or implicitly, by instructing the printer to generate characters or vectors (lines). Fonts available for printing are stored on the printer's microcode or font diskette. Most page printers support macros, which are a saved list of PMP commands, avoiding the necessity for the application program to send a string of individual commands each time a particular printed function is required.

There are five basic categories of PMP commands:

### Page commands

Set overall page parameters, such as size and orientation

### Cursor commands

Move the cursor on the page map

### Font commands

Manage fonts within the page printer

### Generation commands

Create pels on the page map

### Macro commands

Allow strings of other commands to be saved for later processing.

Printing capabilities and functions in ASCII attach mode are governed by individual application programs that are written to suit the capabilities of specific printers (or printers that provide an emulation of that printer). There is no architectural data stream standard to which ASCII printers can conform in the interests of uniformity. ASCII printing applications are therefore totally printer dependent.

On the i5/OS operating system, ASCII printing support is provided by translating EBCDIC characters to the ASCII equivalents.

## Printer device description

Printer device descriptions must be created for each printer attached to the system.



Use the Create Device Description (Printer) (CRTDEVPRT) command to assign a name to each printer. If you have a twinaxial-attached printer, automatic printer device configuration is done for you by the system.

## Job description

A *job description* is a system object, made up of many parameters, that defines how a job is to be processed.

Once a job begins, the parameters in the job description become the attributes of the job. For more information, see Job description in the Work management topic.

The following job description parameters provide information to determine where the printed output will go:

- Output queue (OUTQ)
- Printer device (PRTDEV)

### Related information:

Job description

## Workstation description

The *workstation description*, for a display station, is a collection of information that tells the system how the display station is to be used.

The following workstation description parameters provide information to determine where the printed output will go:

- Output queue (QUTQ)
- Printer device (PRTDEV)

When you sign on to the system, if no defaults or system values have been changed, your output will be sent to the output queue and the printer device specified in the workstation description of the display station where you are signed on. Any batch job submitted from the interactive job uses the same printer device or output queue that is currently specified in the workstation description.

## User profile

The *user profile* is an object with a unique name that contains the user password, the list of special authorities assigned to a user, and the objects the user owns.

To become a user of the system, you must have a user profile. In most cases, someone having security officer authority adds new users to the system. A user profile is created for each new user added to the system.

The following user profile parameters provide information to determine where the printed output will go:

- Job description (JOBID)
- Output queue (QUTQ)
- Printer device (PRTDEV)

## System values

System values are objects supplied by IBM and included in the system. System values control such things as system date, system time, default system printer, and so on.

System values associated with printing include:

**Default printer (QPRTDEV)**

Sets the default printer for the system. The value supplied by IBM for the system value QPRTDEV is PRT01.

**Automatically clean up unused printer output storage (QRCLSPLSTG)**

Sets automatic clean up for unused printer output storage and specifies the retention period.

**Format when using Print key (QPRTKEYFMT)**

Specifies whether border or header information is included when using the Print key.

**Printed page footer (QPRTTXT)**

Specifies the page footer for the system.

**Related information:**

Printing system values: Default printer

Storage system values: Automatically clean up unused printer output storage

Printing system values: Format when using Print key

Printing system values: Printed page footer

## Controlling printing to output queue or printer

The routing of the spooled file to an output queue or to a printer is based on values stored in several printing elements. These values are looked at by the system in a defined order. This process gives you a great amount of flexibility to control the routing of spooled files.

### Routing process order

This topic gives an overview of the routing process.

Assuming that the method used to start the job does not override the output queue or printer device values, the following order is followed:

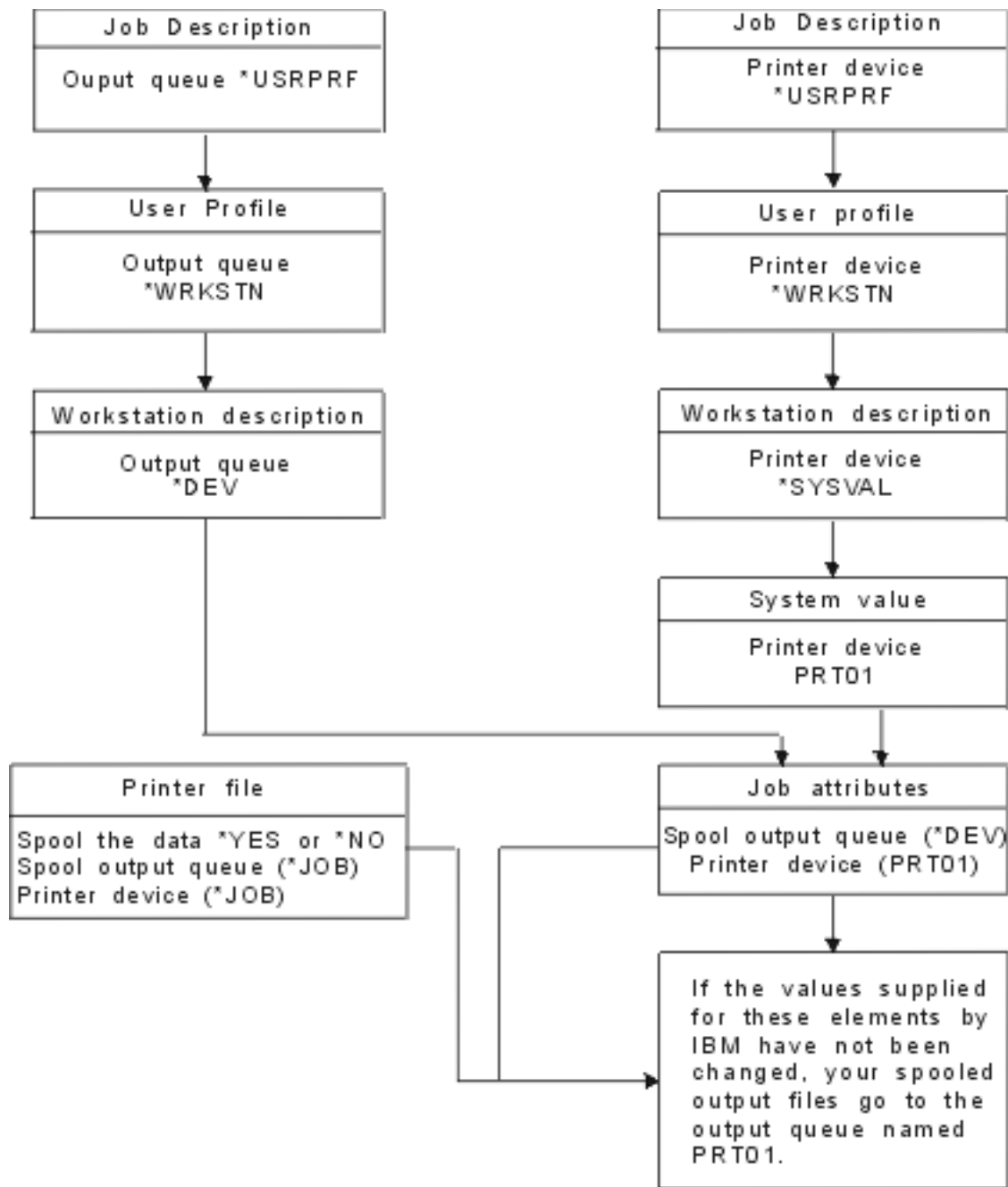
- The job initiation phase begins. The system looks at the following values:
  1. The output queue value in the job description.
  2. The output queue value in the user profile.
  3. The output queue value in the workstation description.
  4. The printer device value in the job description.
  5. The printer device value in the user profile.
  6. The printer device value in the workstation description.
  7. The system value Default printer (QPRTDEV). If the printer identified in the system value does not exist, the output goes to the QPRINT output queue.

The results of this phase determine the job attributes.

- The job run time phase begins. The system looks at the following values:
  1. The output queue value in the printer file.
  2. The output queue value in the job attributes.
  3. The printer device value in the printer file.
  4. The printer device value in the job attributes.
  5. The SPOOL value in the printer file.

The results of this phase determine the output queue name or printer device name to where the spooled file is routed.

The following figure shows the printing elements and how they relate to the printing process.



RR APT50-1

The Change Job (CHGJOB) CL command allows you to override the output queue and printer device values. The Override with Printer File (OVRPRTF) CL command allows you to override many printer file values including the output queue and printer device values. See Printer file overrides for more information on overriding printer file values.

### Printing in a batch environment

When a user signs on to the system, an output queue and a printer device are established for that job.

The names for the output queue and printer device are resolved when the system searches through the user profile, job description, workstation description, and system values.

A similar process occurs when batch jobs are started on the system, with two differences:

- The output queue and printer device values are passed to the batch job from the job that started it. This means that no resolution of the output queue and printer device needs to be done.
- The value \*WRKSTN has no real meaning because batch jobs do not have an associated workstation. If \*WRKSTN is encountered in the resolution of output queue and printer device, \*WRKSTN is replaced with \*DEV if output queue is being resolved and \*SYSVAL if printer device is being resolved.

**Related concepts:**

“Printer file overrides” on page 6

Overriding files (printer files, display files, diskette files, database files, and tape files) can be done by commands, from CL programs, or from high-level language programs. Overrides can be called from different levels (programs calling another program). This article discusses overriding printer files.

**Related information:**

Change Job (CHGJOB) command

Override with Printer File (OVRPRTF) command

**Routing examples**

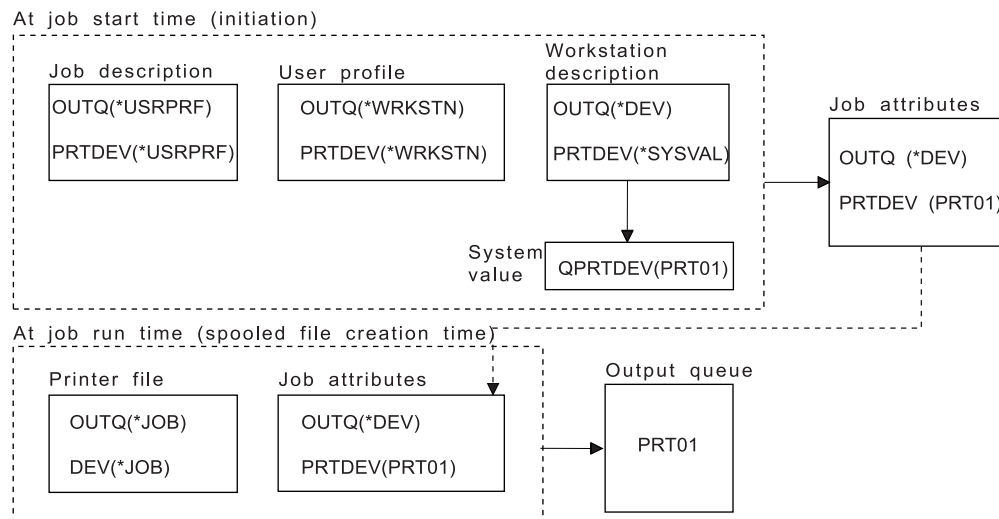
These examples demonstrate different types of routing based on changed values in the printing elements.

Example name	Assumptions include
Example 1: Determine your output queue	Default values
Example 2: Determine your output queue	OUTQ values set
Example 3: Determine your output queue	Current user has a group profile
Example 4: Determine your output queue	<ul style="list-style-type: none"> <li>• Switch to alternate user profile</li> <li>• Printer file SPLFOWN set</li> <li>• Job description OUTQ set</li> </ul>
Example 5: Determine your output queue	<ul style="list-style-type: none"> <li>• Switch to alternate user profile</li> <li>• Data area</li> <li>• Job description OUTQ set</li> </ul>
Example 6: Determine your output queue	<ul style="list-style-type: none"> <li>• Switch to alternate user profile</li> <li>• Job description OUTQ set</li> </ul>
Example 7: Determine your output queue	<ul style="list-style-type: none"> <li>• Switch to alternate user profile</li> <li>• Group profile</li> <li>• Data area</li> <li>• Second data area</li> </ul>
Example 8: Determine your printer name	<ul style="list-style-type: none"> <li>• User profile PRTDEV set</li> <li>• Workstation description PRTDEV set</li> </ul>
Example 9: Determine your printer name	Workstation description PRTDEV set
Example 10: Determine your printer name when using batch	<ul style="list-style-type: none"> <li>• Job runs in batch</li> <li>• Default values</li> </ul>
Example 11: Determine your printer name when using batch	<ul style="list-style-type: none"> <li>• Job runs in batch</li> <li>• Submit Job PRTDEV set</li> <li>• Submit Job OUTQ set</li> </ul>
Example 12: Determine your printer name when using batch	<ul style="list-style-type: none"> <li>• Job runs in batch</li> <li>• Submit Job PRTDEV set</li> <li>• Submit Job OUTQ set</li> </ul>

After you have reviewed the examples, you can take the self-test.

**Example 1: Determining your output queue:**

In the printer file, assume that: the SPLFOWN value is \*CURUSRPRF; the OUTQ value is \*JOB; the DEV value is \*JOB; the SPOOL value is \*YES. Because the SPOOL value is \*YES, the output must go to an output queue. Also assume that there has not been a switch to an alternate user profile.



RBAFT510-2

At job initiation, the following takes place:

The system looks at the OUTQ parameter in the job description; that value is \*USRPRF. This tells the system to look at the OUTQ parameter in the user profile. In this example, that value is \*WRKSTN. This tells the system to look at the OUTQ parameter in the workstation description. In the workstation description, the OUTQ parameter value is \*DEV. \*DEV is stored in the job attribute OUTQ.

The system looks at the PRTDEV parameter in the job description; that value is \*USRPRF. This tells the system to look at the PRTDEV parameter in the user profile. In this example, that value is \*WRKSTN. This tells the system to look at the PRTDEV parameter in the workstation description. This tells the system to look at the system value Default printer (QPRTDEV) and to use the output queue on the system that has the same name as the system printer named in the system value Default printer (QPRTDEV). In this example, that is PRT01. PRT01 is stored in the job attribute PRTDEV.

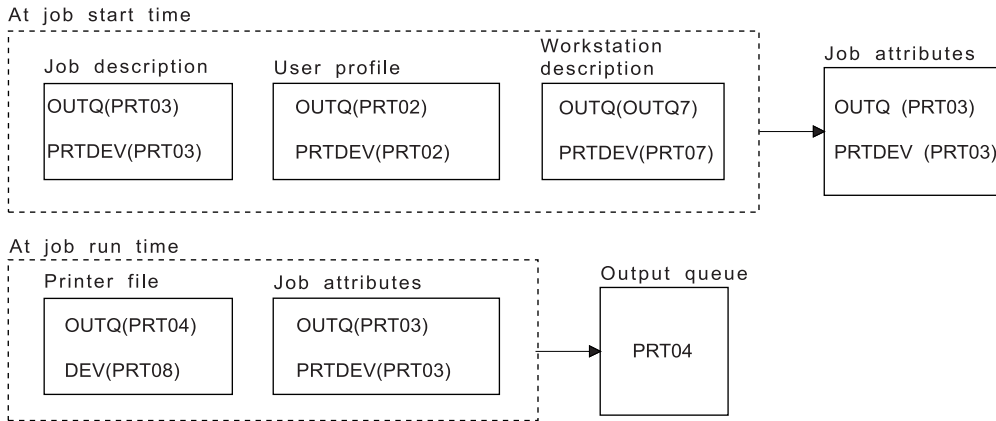
At job run time, the following takes place:

When a spooled file is to be created, the system looks at the OUTQ parameter in the printer file for the output queue name. In this example, that value is \*JOB. This tells the system to look at the OUTQ attribute for the job. The OUTQ job attribute was set to \*DEV during the job initiation stage. The OUTQ job attribute, \*DEV, tells the system to look at the DEV parameter in the printer file. The value in the DEV parameter of the printer file is \*JOB. That value tells the system to look at the PRTDEV attribute of the job.

If the IBM-supplied value for QPRTDEV has not been changed, the printer device name is PRT01 and the output queue name is PRT01.

**Example 2: Determining your output queue:**

In the printer file, assume that: the SPLFOWN value is \*CURUSRPRF; the OUTQ value is PRT04; the DEV value is PRT08; the SPOOL value is \*YES.

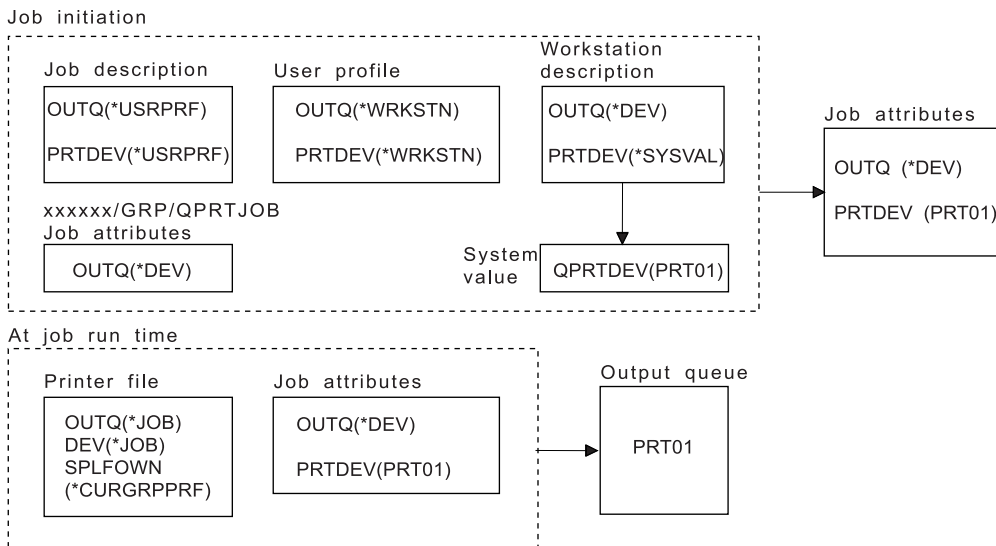


RBAFT504-2

The output queue is PRT04. The system found the output queue parameter value of PRT04 in the printer file rather than a value that points it to your OUTQ job attribute.

### Example 3: Determining your output queue:

In the printer file, assume: the SPLFOWN value is \*CURGRPPRF; the OUTQ value is \*JOB; the DEV value is \*JOB; the SPOOL value is \*YES. Also assume: the job did not switch to an alternate user profile; the current user has a group profile GRP.



RBAFT515-2

**Note:** Because the SPLFOWN parameter of the printer file is \*CURGRPPRF, the spooled file will be created under job xxxxxx/GRP/QPRTJOB (where xxxxxx is 000000-999999).

At job initiation, the following events take place:

The system looks at the OUTQ value in the current job description. The value \*USRPRF, in the job description, tells the system to look at the OUTQ parameter in the user profile. The value in the OUTQ parameter of the user profile is \*WRKSTN. This tells the system to look at the OUTQ parameter in the workstation description. In the workstation description, the OUTQ parameter is \*DEV. In the job attributes, the OUTQ job attribute gets set to \*DEV.

The system looks at PRTDEV parameter in the job description. The value \*USRPRF, in the job description, tells the system to look at the PRTDEV parameter in the user profile. The value \*WRKSTN, in the user

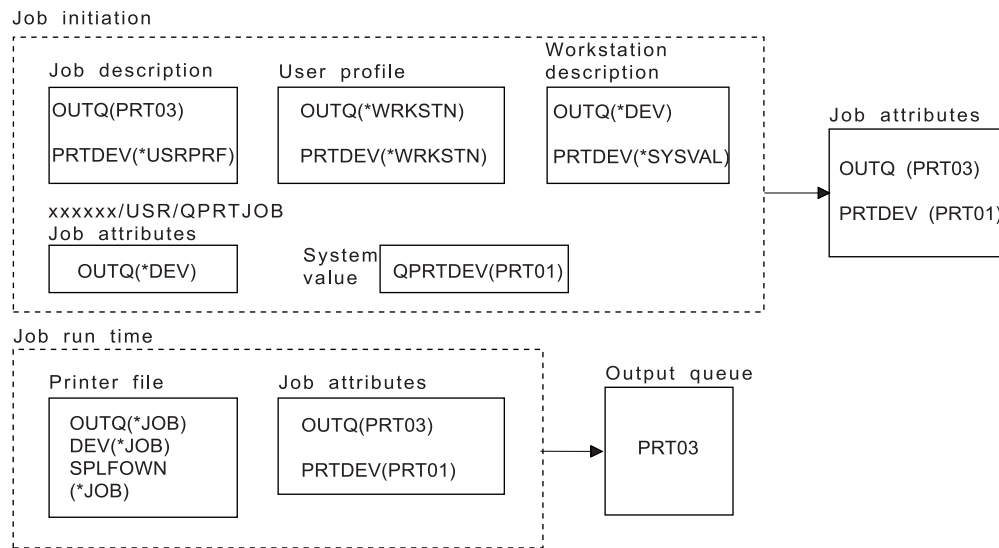
profile, tells the system to look at the PRTDEV parameter in the workstation description. The value \*SYSVAL, in the workstation description, tells the system to look at the system value and use the value set for Default printer (QPRTDEV). The value in Default printer (QPRTDEV) is PRT01. PRT01 becomes the value for the PRTDEV job attribute.

At job run time, the following events take place:

The system looks at the OUTQ value in the printer file. That value \*JOB tells the system to use the OUTQ job attribute for job xxxxxx/GRP/QPRTJOB which is \*DEV. This tells the system to look at the DEV attribute in the printer file, which is \*JOB. The value, \*JOB tells the system to look at the PRTDEV value in the job attributes. The value for the PRTDEV job attribute is PRT01.

**Example 4: Determining your output queue:**

In the printer file, assume the following: the SPLFOWN value is \*JOB; the OUTQ value is \*JOB; the DEV value is \*JOB; the SPOOL value is \*YES. Also assume that there has been a switch to an alternate user profile USR.



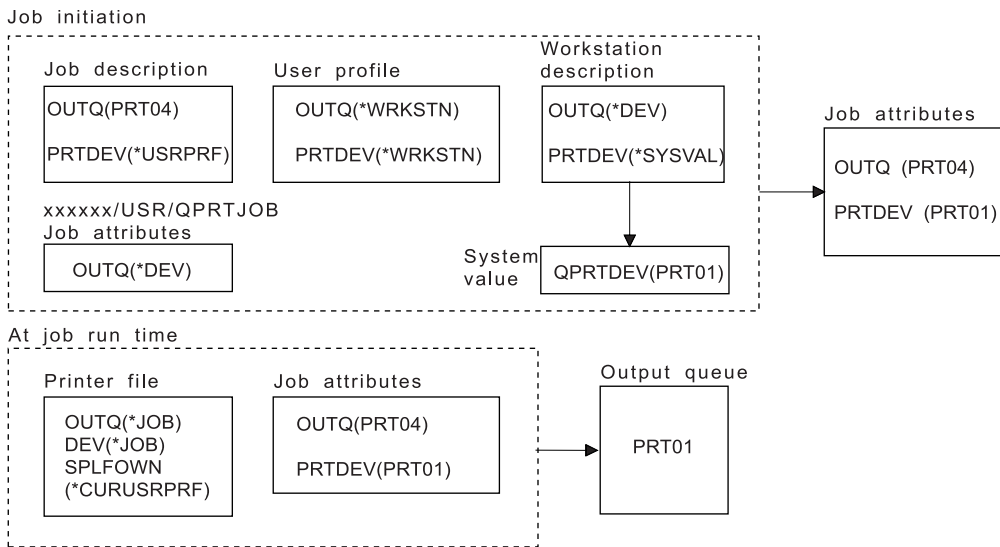
RBAFT512-1

**Note:** The SPLFOWN parameter of the printer file is \*JOB, and the job has switched to user profile USR. The current job will create the spooled file.

The system looks at the OUTQ parameter in the printer file for the output queue name. The value, in this example, \*JOB, tells the system to look at the OUTQ job attribute. Because the SPFLOWN parameter is set to \*JOB, the job attribute OUTQ of the current job is used. The value is PRT03. In this example, the spooled file goes to output queue PRT03.

**Example 5: Determining your output queue:**

In the printer file, assume: the SPLFOWN value is \*CURUSRPRF; the OUTQ value is \*JOB; the DEV value is \*JOB; the SPOOL value is \*YES. Also assume: there has been a switch to an alternate user profile USR; the data area QPRTJOB, of type \*LGL, with a value of false (0), exists in library QUSRSYS and is owned by QSPL user profile.



RBAFT513-1

**Note:** The SPLFOWN parameter of the printer file is \*CURUSRPRF, and the job has switched to user profile USR. The spooled file will be created under job xxxxxx/USR/QPRTJOB (where xxxxxx is 000000-999999).

At job initiation, the following takes place:

The system looks at the OUTQ parameter in the job description. That value, PRT04 tells the system that it does not need to look any further and sets the OUTQ job attribute to PRT04.

The value \*USRPRF in the PRTDEV parameter of the job description, tells the system to look at the PRTDEV attribute of the user profile. In the user profile, the value \*WRKSTN tells the system to look at the PRTDEV parameter in the workstation description. That value, \*SYSVAL tells the system to look at the system value Default printer (QPRTDEV) and to use the output queue that is named in that value. In this example, the value is PRT01 and it is stored in the job attribute PRTDEV.

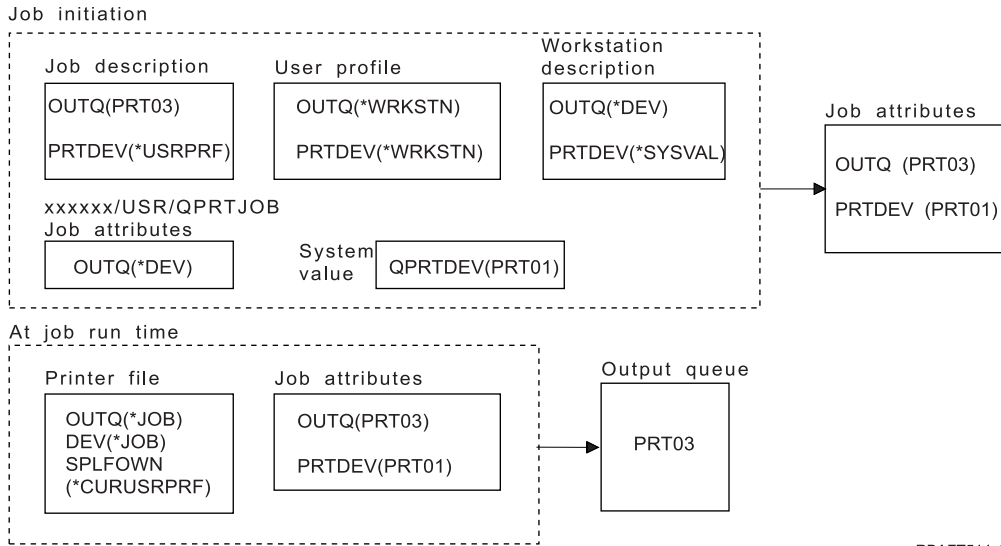
At job run time, the following takes place:

The system looks at the OUTQ parameter in the printer file for the output queue name. That value \*JOB, tells the system to look at the OUTQ attribute of the job. There is a data area, QPRTJOB in QUSRSYS, that is owned by the user profile QSPL that has a logical value of false. Because of this data area, the system will look at the OUTQ attribute for job xxxxxx/USR/QPRTJOB. In the xxxxxx/USR/QPRTJOB, the OUTQ attribute value \*DEV tells the system to look at the DEV parameter in the printer file. The value \*JOB in the DEV parameter of the printer file, tells the system to look at the PRTDEV attribute of the current job. That value is PRT01.

#### Example 6: Determining your output queue:

In the printer file, assume: the SPLFOWN value is \*CURUSRPRF; the OUTQ value is \*JOB; the DEV value is \*JOB; the SPOOL value is \*YES. Also assume that there has been a switch to an alternate user profile USR.





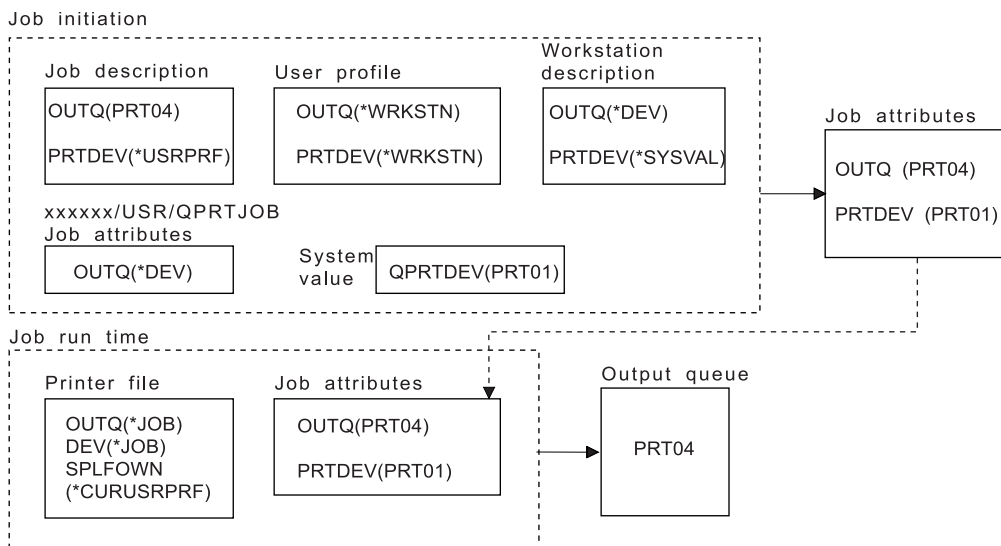
RBAFT514-1

**Note:** The SPLFOWN parameter of the printer file is \*CURUSRPRF, and the job has made a switch to user profile USR. The spooled file will be created under job xxxxxx/USR/QPRTJOB (where xxxxxx is 000000-999999).

The system looks at the OUTQ parameter in the printer file for the output queue name. That value, \*JOB, tells the system to look at the OUTQ job attribute. The system will look at the OUTQ job attribute of the current job, which is PRT03.

### Example 7: Determining your output queue:

In the printer file, assume: the SPLFOWN value is \*CURUSRPRF; the OUTQ value is \*JOB; the DEV value is \*JOB; the SPOOL value is \*YES. Also assume: there has been a switch to alternate user profile USR; the current user has a group profile X; the data area QPRTJOB, of type \*LGL, with a value of false (0), exists in library QUSRSYS and is owned by QSPL user profile; another data area QPRTJOB, of type \*LGL, with a value of true (1), exists in the first product library of the current job's library list. The QSPL user profile owns the data area.



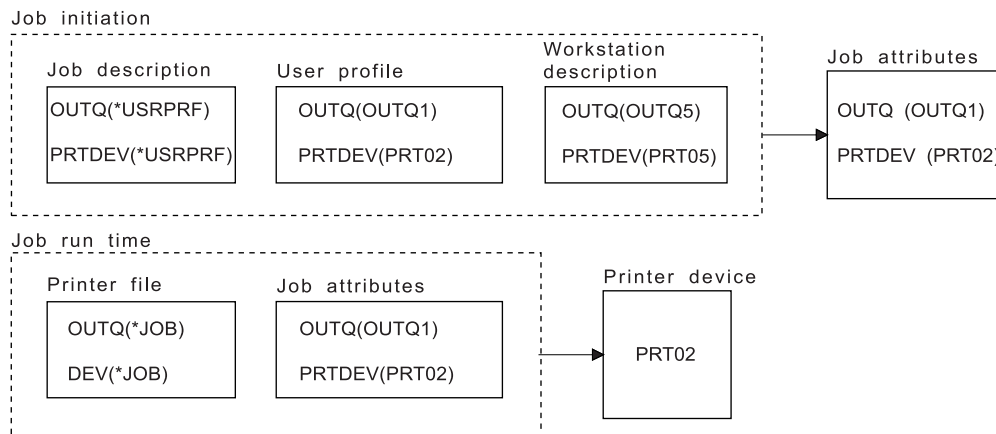
RBAFT511-1

**Note:** The SPLFOWN parameter of the printer file is \*CURUSRPRF, and the job has switched to alternate user profile USR. The spooled file will be created under job xxxxxx/USR/QPRTJOB (where xxxxxx is 000000-999999).

The system looks at the OUTQ parameter in the printer file for the output queue name. That value, \*JOB, tells the system to look at the OUTQ job attribute. Because data area QPRTJOB with a logical value of true exists, the system will look at the OUTQ job attribute of in the current job, which is PRT04.

**Example 8: Determining your printer name:**

In the printer file, assume that: the SPLFOWN value is \*CURUSRPRF; the OUTQ value is \*JOB; the DEV value is \*JOB; the SPOOL value is \*NO. Also assume that there has not been a switch to an alternate user profile.



RBAFT505-2

At job initiation, the following takes place:

The system looks at the OUTQ parameter in the job description. That value \*USRPRF tells the system to look at the OUTQ parameter in the user profile. The value of the OUTQ parameter in the user profile is OUTQ1. Because this is the name of a specific output queue, this value is stored as the OUTQ value in the job attributes.

The system looks at the PRTDEV parameter in the job description. That value \*USRPRF tells the system to look at the PRTDEV parameter in the user profile. The PRTDEV value in the user profile is PRT02. Because this is the name of a specific printer device, the system stops looking and stores this value as the PRTDEV value in the job attributes.

At job run time, the following events take place:

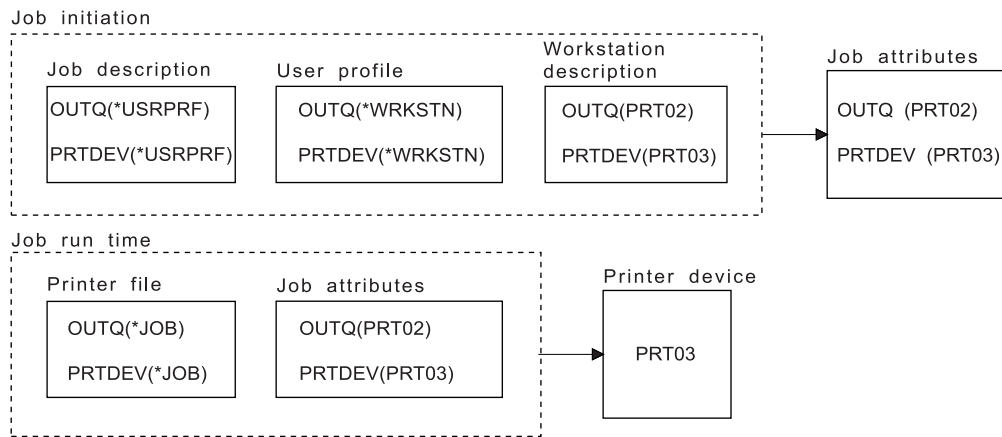
Your printer device is PRT02. This is because the system looked first into the printer file and found the PRTDEV parameter value to be \*JOB, which sent it to the job attribute PRTDEV.

The job attribute PRTDEV value is PRT02.

In this example, the printer file specified SPOOL = \*NO. Your output goes directly to PRT02 for printing and no output queue is used.

**Example 9: Determining your printer name:**

In the printer file, assume that: the SPLFOWN value is \*CURUSRPRF; the OUTQ value is \*JOB; the DEV value is \*JOB; the SPOOL value is \*NO. Also assume that there has not been a switch to an alternate user profile.



RBAFT503-2

At job initiation, the following takes place:

The system looks at the OUTQ parameter in the job description. That value, \*USRPRF tells the system to look at the OUTQ parameter in the user profile. In the user profile, the OUTQ parameter value is \*WRKSTN. This value tells the system to look at the OUTQ parameter in the workstation description. The OUTQ value in the workstation description is PRT02. This value is stored as the OUTQ value in the job attributes.

The system looks at the PRTDEV value in the job description. That PRTDEV value in the job description is \*USRPRF. That value tells the system to look at the PRTDEV value in the user profile. The PRTDEV value in the user profile is \*WRKSTN. That value tells the system to look at the PRTDEV value in the workstation description. The PRTDEV value in the workstation description is PRT03. This value is stored as the PRTDEV value in the job attributes.

At job run time, the following events take place:

The system looked at the printer file and found the PRTDEV parameter value to be \*JOB, which tells the system to look next in the job printer attribute PRTDEV.

In this example, that value is PRT03.

In this example, the printer file specified SPOOL = \*NO. Your output goes directly to PRT03 for printing and no output queue is used.

**Remember:**

You must know the value (\*YES or \*NO) of the SPOOL parameter in the printer file to determine if your output goes to an output queue or to a printer. If SPOOL = \*YES, a spooled file goes to an output queue. If SPOOL = \*NO, the output goes directly to a printer.

**Example 10: Determining your printer name when using batch:**

Assume that: there has not been a switch to an alternate user profile; the spooled file owner is \*CURUSRPRF; the user's output queue (OUTQ) is OUTQ1 and the printer device (PRTDEV) is PRT1; the Submit Job (SBMJOB) command is used to submit the job to batch; the output queue parameter on the SBJOB command is specified as \*CURRENT; the printer device parameter on the SBJOB command is specified as \*CURRENT.

When the job runs in batch, the resulting spooled file is sent to OUTQ1 and that spooled file prints on the printer assigned to OUTQ1.

If no spooling is used when printing, the output goes to printer device PRT1.

The reason OUTQ1 and PRT1 are used is that \*CURRENT is the value passed to the batch job by the user.

**Example 11: Determining your printer name when using batch:**

Assume that: there has not been a switch to an alternate user profile; the spooled file owner \*CURUSRPRF value on the printer file has not been overridden; the user's output queue (OUTQ) is OUTQ1 and the printer device (PRTDEV) is PRT1; the Submit Job (SBMJOB) command is used to submit the job to batch; the output queue parameter on the SBJOB command is specified as \*USRPRF; the user profile has \*WRKSTN as the value for the output queue parameter; the printer device parameter on the SBJOB command is specified as PRT99.

When the job runs in batch, the resulting spooled file is sent to the output queue named PRT99 and the spooled file prints on PRT99. The \*WRKSTN value for output queue is interpreted as \*DEV and an output queue having the same name as the printer device is selected.

If no spooling is used when printing, the output goes to printer device PRT99.

**Example 12: Determining your printer name when using batch:**

Assume that: there has not been a switch to an alternate user profile; the spooled file owner \*CURUSRPRF value on the printer file has not been overridden; the user's output queue (OUTQ) is OUTQ1 and the printer device (PRTDEV) is PRT1; the Submit Job (SBMJOB) command is used to submit the job to batch; the output queue parameter on the SBJOB command is specified as \*USRPRF; the user profile has \*WRKSTN as the value for the output queue parameter; the printer device parameter on the SBJOB command is specified as \*WRKSTN.

When the job runs in batch, the resulting spooled file is sent to the system printer. This is because the \*WRKSTN value for output queue is interpreted as \*DEV and the printer device value of \*WRKSTN is interpreted as \*SYSVAL.

If no spooling is used when printing, the output goes to the printer defined as the system printer. This is the printer name assigned to the system value Default printer (QPRTDEV).

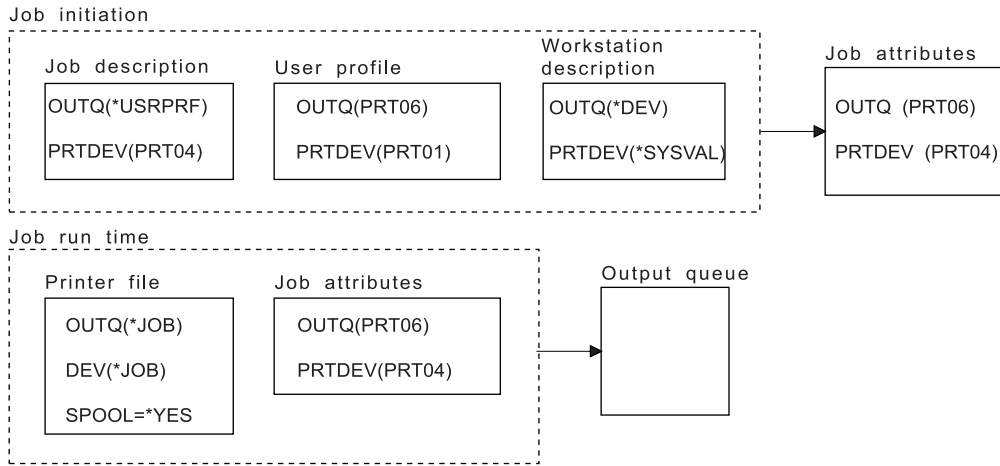
**Self-test: Determining output queue and printer device:**

The diagrams in this test are similar to the ones used in the examples. Read the information in the diagrams. Using the information you have acquired about the hierarchy of printing elements, determine what the output queue and printer device names are.

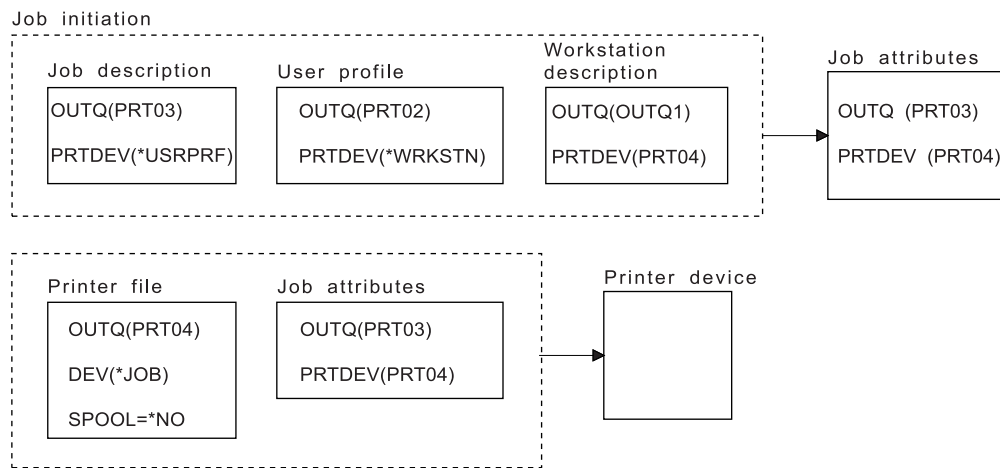
**Note:** Keep in mind the SPOOL parameter value when deciding on your answer.

You should also assume the following for both self-tests:

- There has not been a switch to an alternate user profile.
- The spooled file owner attribute is \*CURUSRPRF.



RBAFT506-1



RBAFT507-1

When you are done, check your answers with Self-test answers.

**Related reference:**

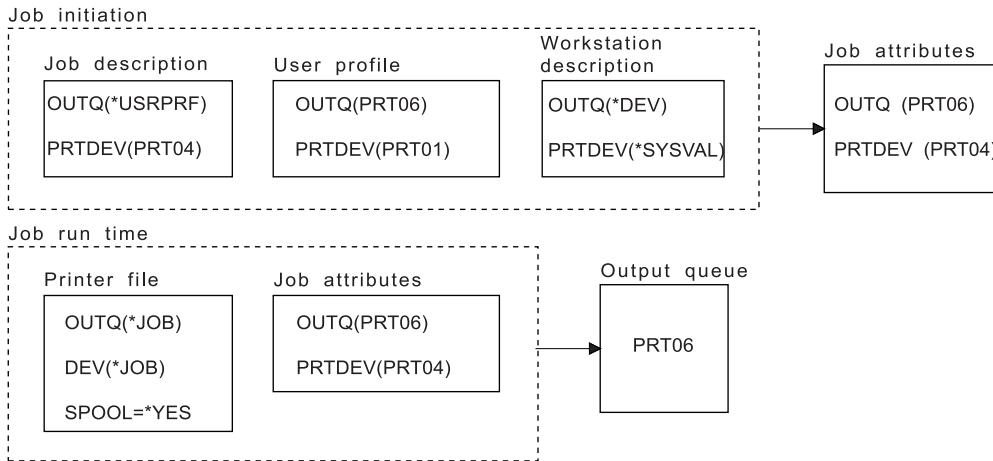
“Self-test answers”

Here are the diagrams from the self-test, with the correct output queue and printer device parameter values filled in.

**Self-test answers:**

Here are the diagrams from the self-test, with the correct output queue and printer device parameter values filled in.

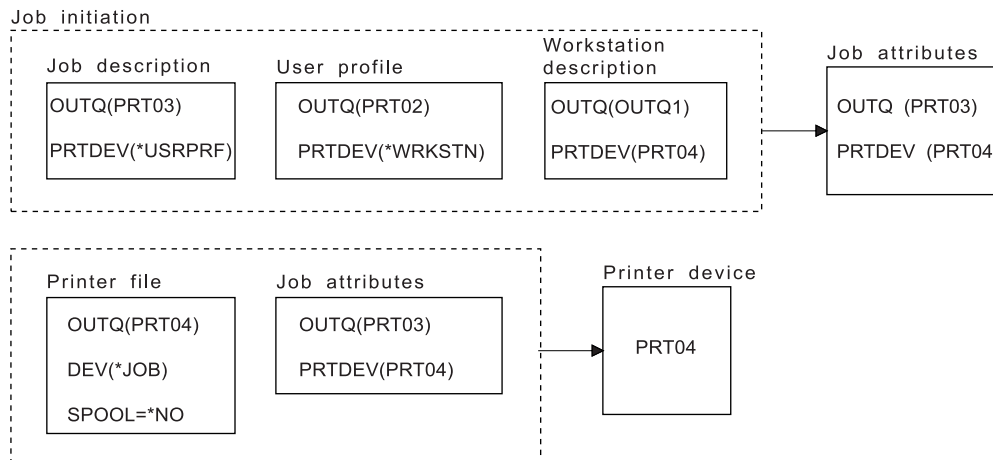
For the first diagram, the output queue name is PRT06.



RBAFT508-1

The system looked first at the printer file and found SPOOL = \*YES. Next, it looked at the output queue value in the printer file, which is \*JOB. It then looked at the output queue value in the job attribute OUTQ which is PRT06.

For the second diagram, the printer device value is PRT04.



RBAFT509-0

Again, the system looked first at the printer file, but this time it found SPOOL = \*NO. Next, it looked at the device value in the printer file, which is \*JOB. It then looked at the device value in the job attribute PRTDEV.

The job attribute PRTDEV is PRT04.

**Related reference:**

“Self-test: Determining output queue and printer device” on page 52

The diagrams in this test are similar to the ones used in the examples. Read the information in the diagrams. Using the information you have acquired about the hierarchy of printing elements, determine what the output queue and printer device names are.

**Printer attachment methods**

You can use many methods to attach a printer to a System i product, such as a TCP/IP network, a personal computer, a twinaxial workstation controller, and an ASCII workstation controller.

## **TCP/IP-network-attached printers**

Attaching a printer with a TCP/IP network requires one of several TCP/IP network print protocols.

### **Intelligent Printer Data Stream with Print Services Facility:**

Intelligent Printer Data Stream (IPDS) with Print Services Facility (PSF) offers industry-leading printing performance and functions, and can take advantage of the Advanced Function Presentation print resources.

#### **Related information:**

Advanced Function Presentation

### **Simple Network Management Protocol:**

Simple Network Management Protocol (SNMP) printing provides excellent support for resource sharing and problem handling by using two separate TCP/IP ports for communication, one to send print data and another to track job status.

Using two separate TCP/IP ports for communication allows SNMP printing solutions to display error messages or status while a job is printing. SNMP also uses IBM Shared Connections to ensure that sockets are released after every copy of a printed document. This allows the system to efficiently share a printer with other users. Additionally, since SNMP supports both PostScript and Print Job Language (PJM) print data, it offers excellent hardware and application compatibility.

SNMP printing requires the printer and print server or network adapter to support the Host Resource Management Information Base (Host Resource MIB), and for full functionality, the Printer Management Information Base (Printer MIB). Not all printing hardware will support SNMP, so you should carefully check for compatibility before implementing this solution.

#### **Related tasks:**

“Configuring SNMP printers” on page 120

To set up Simple Network Management Protocol (SNMP) printers, follow these steps.

### **Printer Job Language:**

Printer Job Language (PJM) printing solutions provide problem handling and status information during the print process by using bidirectional communication between the printer and print server over a single TCP/IP port.

With PJM printing, you can share a printer between the System i product and other network users, but since the IBM i operating system keeps communicating with the printer until the IBM i output queue is empty, resource sharing is more limited than with either Simple Network Management Protocol (SNMP) or line printer requester/line printer daemon (LPR/LPD).

PJM printing through a System i product requires that the printer and printer adapter or network adapter support Printer Control Language level 5e. Also, the cabling, printer adapter, and network adapter (or print server) must be capable and configured for bidirectional communication.

#### **Related tasks:**

“Configuring PJM printers” on page 119

To set up Hewlett Packard print job language (PJM) printers, follow these steps.

### **Internet Printing Protocol:**

Internet Printing Protocol (IPP) printing allows you to send and manage print information from a variety of remote sites by sending print information over the Internet or intranet. IPP is a versatile printing method, and is supported by a variety of printers and network adapters.

You can also send print information using IPP to any IBM i printer (even if the printer does not support IPP) by using the IBM IPP server for i5/OS.

This protocol provides the advantages of printing with line printer requester/line printer daemon (LPR/LPD), but is significantly easier to manage and troubleshoot because print status information is available during the printing process. IPP also provides excellent security by allowing Secure Sockets Layer (SSL) encryption.

IPP printing solutions communicate using TCP/IP ports, and require an HTTP server, Java™, and Digital Certificate Manager (if SSL is used). Not all devices support IPP, so verify hardware compatibility before implementing an IPP-based printing solution.

IPP is transported over HTTP 1.1 using a message body whose content-type is application/ipp. IPP uses well-known port 631.

**Related concepts:**

“Internet Printing Protocol server” on page 79

The IBM Internet Printing Protocol (IPP) server for i5/OS allows IPP clients (such as Windows PCs) to submit and manage print jobs to any System i printer.

“Setting up the IPP server” on page 127

You can administer and configure the Internet Printing Protocol (IPP) server using the IBM IPP Server Administrator for iSeries, a browser-driven graphical user interface.

**Related tasks:**

“Configuring IPP printers” on page 125

To set up Internet Printing Protocol (IPP) printers, follow these steps.

**Line printer requester/line printer daemon:**

Line printer requester/line printer daemon (LPR/LPD) printing sends print information from a remote output queue to remote servers or printers. This printing method is supported by most hardware, but provides less error handling support than other options. It also provides the least amount of print function, and does not support page range selection or job accounting.

This printing method requires that you configure remote output queues for the remote printers. Most printers and adapters will support this protocol. See Configure LPR/LPD for information about how to configure LPR/LPD printing.

**Related tasks:**

“Configuring LPR/LPD” on page 122

To configure line printer requester/line printer daemon (LPR/LPD) printing, follow these steps.

**PC-attached printers**

If you have a printer attached to your personal computer (PC), an emulation program needs to be run on the personal computer to allow i5/OS to print to it. This type of emulation program is included with the IBM i Access for Windows licensed program.

For further information about attachment of printers to personal computers with IBM i Access for Windows, see the System i Access topic.

**Related information:**

System i Access

**Twinaxial-workstation-controller-attached printers**

The twinaxial workstation controller provides the ability to attach twinaxial (5250) displays and printers to the System i products.



For further information about attachment of printers to the twinaxial workstation controller, see Local Device Configuration PDF.

**Related information:**

 Local Device Configuration PDF

### **ASCII-workstation-controller-attached printers**

The ASCII workstation controller provides the ability to attach ASCII displays, ASCII printers, and personal computers to System i products through the RS232 or RS422 interface.

The workstation controller also provides the ability to attach a personal computer that is running the IBM i Access for Windows workstation function. A printer attached to the personal computer can be used as a System i printer.

For further information about attachment of printers to the workstation controller, see ASCII Work Station Reference PDF.

**Related information:**

 ASCII Work Station Reference PDF

### **Lexlink-network-attached printers**

ASCII LAN-attached printers that use the Lexlink protocol must be attached to an IBM 4033 LAN adapter device or a MarkNet XLe device, or the printer must have a MarkNet or MarkNet XL Internal Network Adapter (INA) card in the printer. (The IBM 4039 Printer is an example of a printer with an INA card.)

**Related tasks:**

“Configuring Lexlink printers” on page 129  
To set up Lexlink printers, follow these steps.

### **Printers attached to IBM InfoWindow 3477, 3486, 3487, and 3488 display stations**

The IBM InfoWindow display station can be locally attached to the System i product or remotely attached to the IBM 5294 or 5394 Remote Control Unit using twinaxial cable. The InfoWindow has a printer port that can support attachment of most personal printers.

The attached printer can be used either as a local screen printer or as a system printer to print i5/OS spooled jobs (for example, printing an i5/OS document or a job generated on a personal computer using the network printer function).

There are several advantages to using personal printers that attach to the InfoWindow. The lower cost and smaller size of personal printers makes it convenient to put personal printers at your display stations that are attached to your System i product.

**Note:** If you specify local for the type style option for printer setup on a 3477, 3486, 3487, or 3488 InfoWindow display, font selection or substitution by the printer might give you unpredictable results.

## **Remote system printing**

Remote system printing allows spooled files that are created on a System i platform to be automatically sent to and printed on other systems.

The spooled files are sent, from an output queue, using the Start Remote Writer (STRRMTWTR) command. The STRRMTWTR CL command allows spooled files to be automatically sent to other systems using SNA distribution services (SNADS) or Transmission Control Protocol/Internet Protocol (TCP/IP).

## Benefits

Using remote system printing has many benefits.

- Output queue placement.

Spooled files can be placed automatically on a specific output queue of a target system. This support is provided through the Create Output Queue (CRTOUTQ) and Start Remote Writer (STRRMTWTR) commands.

- Multiple remote writers increase throughput.

Output queues can have multiple remote writers started to them. This allows multiple jobs to send spooled files simultaneously from one output queue.

**Note:** 10 remote writers can be started to one output queue.

- One-command interface

Once the environment (hardware and software) has been established, the Start Remote Writer (STRRMTWTR) command initiates all activity necessary to send spooled files to a remote system. An auto-start job entry exists in the QSPL subsystem that starts a job automatically when the QSPL subsystem is started. This job runs the STRRMTWTR command with the OUTQ parameter value set to \*ALL. Therefore, a remote writer(s) is started to all output queues that have a remote system and a number of writers to autostart specified. Remote writers are also started to a remote output queue when the remote output queue is changed or a new one created.

- Distributed print routing with spooled file attributes

For distributed print routing, spooled file attributes are available. They are:

- User who created the file

This attribute identifies the user who created the spooled file.

- System where file was created

This attribute identifies the system on which the spooled file was created.

- User print information

This attribute is composed of the characters retrieved from the user-defined text.

Once a spooled file is created with user defined text, the text cannot be changed. When the spooled file is sent with the data format parameter value of \*ALLDATA, the user print information becomes an attribute of the spooled file.

See User print information for more detail on how to work with the display, retrieve, and change user print information commands.

- Send (SND) and Defer (DFR) statuses for spooled files

These statuses enable you to monitor the activity of spooled files.

- SND

The spooled output file is being sent or has been sent to a remote system

- DFR

The spooled output file has been deferred from being sent

After spooled output files are successfully sent to a remote system (as best as can be determined), they are deleted or saved, as specified by the save spooled file attribute.

### Related concepts:

“User print information” on page 59

User print information is made up of user-defined text associated with a user. The user-defined text is saved with the spooled file when the spooled file is created. It can be displayed using the Display Spooled File Attributes (DSPSPLFA) command or retrieved using the Retrieve User Print Information (RTVUSRPTI) command.

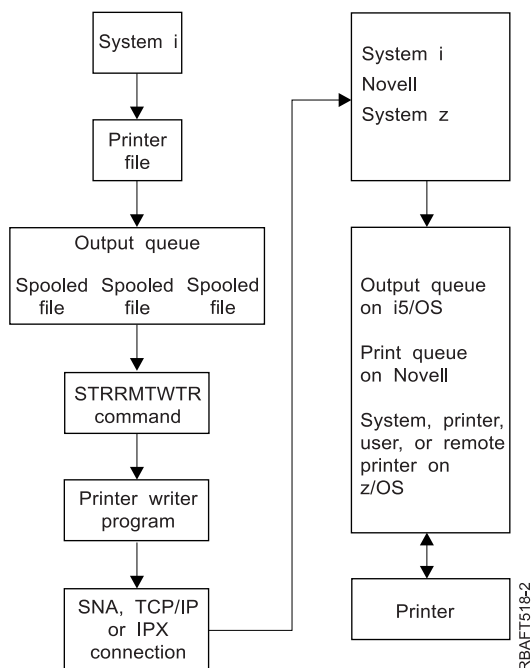
“Send and defer status” on page 60

When a spooled output file is on an output queue, its status can vary depending on the mode of the output queue and the activity taking place with a particular spooled file. Of particular interest to remote

system printing are the statuses of SND and DFR.

## How remote system printing works

This topic explains the process of the remote system printing function.



An output queue is created to hold spooled output files. A remote output queue is an output queue created for use in sending spooled output files to a remote system. Several parameters on the CRTOUTQ CL command are required to do this. When these parameters are given values, you then have a remote output queue. The spooled output files on a remote output queue are sent by the remote writer or remote writers started to the output queue. Remote writers are automatically started based on the value specified on number of writers to autostart (AUTOSTRWTR) parameter. Or they can be started using the STRRMTWTR CL command.

The STRRMTWTR CL command starts the writer that sends the spooled output files on the remote output queue, to the remote system. The writer, which is a system job, takes spooled output files from a remote output queue and sends them to a remote system using SNADS or TCP/IP. The spooled output file can be sent to the same user who owns the spooled output file being sent, a specific output queue, or the output queue for the system printer on the target system. If the sending user profile does not exist on the target system, the QNETSPLF user profile is used when using SNADS.

**Note:** When spooled output files are sent to a system with the destination type of \*OTHER and using SNADS, the user profile to whom the spooled output files are being sent must exist or be created on that target system.

### Related information:

Create Output Queue (CRTOUTQ) command

Start Remote Writer (STRRMTWTR) command

### User print information

User print information is made up of user-defined text associated with a user. The user-defined text is saved with the spooled file when the spooled file is created. It can be displayed using the Display Spooled File Attributes (DSPSPLFA) command or retrieved using the Retrieve User Print Information (RTVUSRPTI) command.

User print information is not used when sending spooled output files to another System i platform or an S/3X system. It is used only as information to pass to the VM/MVS bridge exit program to aid in setting the Network Job Entry (NJE) header fields.

It is possible for a system administrator to limit users access by revoking public authority to the particular commands.

User print information can be used any way you want. For example, it can consist of printout distribution information or be used for accounting information (department to charge for printing).

You can work with user print information by using the Change User Print Information (CHGUSRPRTI), Display User Print Information (DSPUSRPRTI), and Retrieve User Print Information (RTVUSRPRTI) commands.

### Using the CHGUSRPRTI command

There is no command that allows creation of user print information. If no user print information exists, it can be created using the CHGUSRPRTI command.

For example, running the following command modifies (or creates if it does not exist) the user print information for user LAWSON.

```
CHGUSRPRTI USER(LAWSON) TEXT('DEPT. ABC P.O. BOX 123')
```

The command acts on the user print information for the user LAWSON. The user information is changed (or created) to DEPT. ABC P.O.Box 123.

### Using the DSPUSRPRTI command

The Display User Print Information (DSPUSRPRTI) command displays the user print information for the specified user.

```
DSPUSRPRTI USER(LAWSON)
```

**Note:** DSPUSRPRTI uses the QPDSPUSRPI printer file when OUTPUT is specified as \*PRINT.

### Using the RTVUSRPRTI command

The Retrieve User Print Information (RTVUSRPRTI) command can be used in a CL program to retrieve the user print information value associated with a user. The values are returned in the specified CL variables for that user.

```
RTVUSRPRTI USER(LAWSON) RTNTEXT(&TEXT);
```

When the above command is run, the following is returned:

```
&TEXT    'DEPT ABC P.O. BOX 123 ____'
```

The coded character set identifier (CCSID) is used when the text description prints on the output.

### Send and defer status

When a spooled output file is on an output queue, its status can vary depending on the mode of the output queue and the activity taking place with a particular spooled file. Of particular interest to remote system printing are the statuses of SND and DFR.

**Note:** DFR status is not unique to spooled output files on a remote output queue. Spooled output files on nonremote output queues can also have a status of DFR.

- SND

When a spooled output file is being sent to a remote system, it has a status of SND. If the connection type is \*SNA, the spooled output file might remain in SND status until a confirmation message from the remote system is received by the remote writer. At this time the spooled output file is deleted or saved, depending on the save spooled file attribute. If the writer is ended while spooled output files are in SND status, the spooled files are changed back to RDY status.

- DFR

When a writer (printer or remote) is started to an output queue, it determines the maximum spooled file size for the current time. Any RDY spooled files which exceed the limit are changed to DFR status. If the spooled file exceeds the current limit and is added to the output queue (created or moved) after a writer is started to the output queue, the spooled output file status will be DFR.

When the system time-of-day changes such that a new maximum spooled output file size is to take effect, the writer goes through the output queue again and updates RDY spooled files to DFR, or DFR to RDY, depending on the new limit and the size of the particular spooled output file. When the writer is ended, all DFR spooled output files return to RDY.

When time ranges for the maximum spooled output file size overlap, the smaller of the number of pages values is used. For example, assume that there were two time ranges of 8:00:00 to 16:00:00 and 12:00:00 to 12:30:00, with number of pages 40 and 10 respectively. The largest spooled output file that prints from 8:00 a.m. to 12:00 p.m. is 40 pages. The largest spooled output file that prints from 12:00 p.m. to 12:30 p.m. is 10 pages. The largest spooled output file that prints from 12:30 p.m. to 4:00 p.m. is 40 pages.

The following screen capture illustrates an output queue (RMTOUTQ) with a status of released and the first spooled output file (DMB18R1) being written (RLS/WTR). Because DMB18R1 is being sent to a remote system, its status is SND. The next spooled output file, DMB18R2, has a status of DFR. It can be deferred from activity because of its size and the time of day that certain sizes of spooled output files are allowed to be printed or sent.

```

Work with Queue (WRKOUTQ *RMTOUTQ)
Queue:  RMTOUTQ      Library:  Lawson      Status:  RLS/WTR
Type options, press Enter.
1=Send  2=Change  3=Hold  4=Delete  5=Display  6=Release  7=Messages
8=Attributes          9=Work with printing status

Opt  File      User      User Data  Sts  Pages  Copies  Form Type  Pty
--  -
_   DMB18R1    LAWSON
_   STUMPF     LAWSON
_   DMB18R2    LAWSON    TEST      DFR    1      1      *STD      5

Parameters for options 1, 2, 3 or command
====>
F3=Exit  F11=View 2  F12=Cancel  F20=Writers  F22=Printers
F24=More keys
Bottom

```

## Fonts

The fonts mentioned in this topic are included in the i5/OS operating system. Other font products, such as Infoprint Fonts, are available and can be purchased separately.

- TrueType and OpenType fonts (Option 43 - Additional fonts)
- AFP compatibility fonts (Option 8 - AFP compatibility fonts)

These fonts can be supplemented by installing IBM licensed programs that provide additional fonts, creating your own fonts on the System i platform, or purchasing them from other companies.

Fonts are a family or assortment of characters. Three elements typically provide a font identity:

- Type family  
Courier is an example of a type family.
- Typeface  
Style, weight (for example, italic or bold), and width (normal or expanded) define typeface.  
Normal means the typical size of characters, while expanded means that the character is wider than normal.
- Type size  
Fonts can range from small (4 point) to large (72 point).

Some printers have fonts built into them and some do not. If the printer does not have fonts built into it, the server can send (download) character sets and code pages to the printer with the document or independent of the document, to be stored for future use.

### **TrueType and OpenType fonts**

OpenType is an enhanced form of TrueType technology that is designed with Unicode in mind. OpenType is the font technology that IBM is using to support presentation of Unicode. TrueType fonts contain characters to support language and scripts from around the world. Presently, over 52 000 glyphs are contained in a single type style. Subsets of this data are also available to allow a smaller file size to support certain geographic areas.

TrueType fonts are supplied in option 43 ("Additional Fonts") of i5/OS. They are provided as stream files in the TrueType (OpenType) format.

TrueType and OpenType fonts reside in one of two integrated file system directories:

- /QIBM/ProdData/OS400/Fonts/TTFonts for IBM supplied fonts.
- /QIBM/UserData/OS400/Fonts/TTFonts for user fonts.

When searching for fonts, the UserData path will be searched first followed by the ProdData path.

You must use the FONTNAME DDS keyword to select the TrueType fonts. Unlike the other supported font resources, the TrueType fonts are referred to by their full font name, not by the file or object name.

The following fonts are supplied in option 43 ("Additional Fonts"):

- Monotype Sans WT
- Monotype Sans WT J
- Monotype Sans WT K
- Monotype Sans WT ME
- Monotype Sans WT SC
- Monotype Sans WT TC
- Monotype Sans Duospace WT
- Monotype Sans Duospace WT J
- Monotype Sans Duospace WT K
- Monotype Sans Duospace WT ME
- Monotype Sans Duospace WT SC
- Monotype Sans Duospace Ext B<sup>1</sup>

---


1. The Monotype Sans Duospace Ext B font is an extension of the Monotype Sans Duospace WT SC font. The Monotype Sans Duospace Ext B font is linked to the Monotype Sans Duospace WT SC font by the resource access table that is supplied with i5/OS Option 43 - Additional Fonts. This linking makes all the characters from both fonts available to documents that specify the Monotype Sans Duospace WT SC font name.

- Monotype Sans Duospace WT TC
- Times New Roman WT
- Times New Roman WT J
- Times New Roman WT K
- Times New Roman WT ME
- Times New Roman WT SC
- Times New Roman WT TC
- Thorndale Duospace WT
- Thorndale Duospace WT J
- Thorndale Duospace WT K
- Thorndale Duospace WT ME
- Thorndale Duospace WT SC
- Thorndale Duospace WT TC

TrueType fonts can only be used with printer files that have a device type of \*AFPDS.

If you need the additional function provided by linked fonts or you want to capture fonts in the printer, then you need the Infoprint Fonts for Multiplatforms V1.1 (5648-E77) licensed program (or equivalent). This product contains fonts as well as the font installer utility.

When the Infoprint Fonts for Multiplatforms V1.1 (5648-E77) licensed program is installed, it changes the way that fonts are searched for and located. The Infoprint Fonts for Multiplatforms V1.1 (5648-E77) licensed program builds a resource access table. The resource access table contains a mapping of the full font name to a system specific file name. Fonts are now searched for based on the information contained in the resource access table.

For more information about the Infoprint Fonts for Multiplatforms V1.1 (5648-E77) licensed program, see *i5/OS and OS/400<sup>®</sup> software: Printing and output software*  ([www.printers.ibm.com/internet/wwsites.nsf/vwwebpublished/iseriessoftware\\_ww](http://www.printers.ibm.com/internet/wwsites.nsf/vwwebpublished/iseriessoftware_ww)).

**Related information:**

FONTNAME (Font name) keyword in printer files

**AFP compatibility fonts**

The i5/OS operating system comes with a certain variety of fonts called the IBM-supplied compatibility fonts or the compatibility set. These fonts provide a range of font styles that support different types of printers that can be attached to the system.

**Related reference:**

“Font information” on page 307

This table contains the Advanced Function Presentation (AFP) compatibility fonts supported by the i5/OS operating system.

**Font character sets:**

Fonts are named in a number of ways. One way is with a character set name. These character sets are downloaded to the printer. Multiple code pages can be used with a single character set.

For valid code pages that can be used with a character set, see the manual *About Type: IBM’s Technical Reference for 240-Pel Digitized Type*, GS544-3516.

Some font character sets come with the i5/OS operating system, some can be downloaded from a System z to a System i platform, some can be received from another System i platform, and some are available as licensed programs.

The following printers accept downloaded font character sets:

- 3112 (has resident fonts also)
- 3116 (has resident fonts also)
- 3130 (has resident fonts also)
- 3160 (has resident fonts also)
- 3812 (has resident fonts also)
- 3816 (has resident fonts also)
- 3820
- 3825
- 3827
- 3828 (MICR printer)
- 3829
- 3831
- 3835
- 3900–001
- 3900–AFCCU (has resident fonts also)
- 3912 (has resident fonts also)
- 3916 (has resident fonts also)
- 3930 (has resident fonts also)
- 3935 (has resident fonts also)
- 4028 (has resident fonts also)
- 4312 (has resident fonts also)
- 4317 (has resident fonts also)
- 4324 (has resident fonts also)
- Infoprint 20 (has resident fonts also)
- Infoprint 32 (has resident fonts also)
- Infoprint 3000 (has resident fonts also)
- Infoprint 4000 (has resident fonts also)

The use of font character sets provides consistent or similar fonts across printers. For example, a document created at one location using a specific font character set can be sent to a different location, printed on a different model printer, and still look the same.

With some exceptions, the above printers support font character sets that are 240 pels. The Infoprint 3000, Infoprint 4000, Infoprint 20, Infoprint 32, 4028, 3130, 3935, 4312, 4317, and 4324 printers support 300-pel fonts. A pel is a picture element, representing the number of dots in a square inch (for example, 240 across and 240 down).

The 3130, Infoprint 3000, and Infoprint 4000 printers support both 240 and 300 pel fonts. The operator can select which mode the printer is in through the printer operator panel.

### **Naming convention for font character sets**

Font character set names on i5/OS can be up to 8 characters long. Each character or group of characters tells something about the font character set.

For example, in the font character set name C0D0GT10:

**C0** The **C0** means that this object is a font character set.



- D** The **D** indicates the origin of the font. In this example, C0D0GT10 is a font character set designed for Document Control Facility (DCF) for a 3800 Model 1 printer or a 3825 printer.
- 0** This **0** indicates that this font is for uniformly spaced and mixed-pitch font character sets.
- GT10** The **GT10** indicates the type family, typeface, and pitch for uniformly spaced and mixed-pitch fonts. In this example, the GT10 means that this font character set is a Gothic Text style and the characters are 10 pitch or 10 characters per inch.

For more information about font character sets, see the manual *About Type: IBM's Technical Reference for 240-Pel Digitized Type*, GS544-3516.

### Selecting font character sets

Select a font character set to use with an application program by specifying the 8-character font character set name as the value on the FNTCHRSET parameter of the printer file.

If you choose to use font character sets with your applications, you must also specify a code page (by providing a value for the CDEPAG parameter of the printer file being used).

### Substituting font character sets

Substitution is determined by i5/OS based on which font character sets are specified in the application, the type of printer to be used, and the value assigned to the fidelity parameter of the printer file being used (\*CONTENT or \*ABSOLUTE).

#### Example 1

Assume:

- The application calls for font character set C0D0GB10 (Gothic Bold, 10 pitch).
- The printer supports only resident fonts.
- The fidelity parameter value is \*CONTENT.

In this example, the spooled file will print with substituted font ID 39 (Gothic Bold 10 pitch) because the fidelity parameter value is \*CONTENT. If the fidelity parameter value is \*ABSOLUTE, the spooled file is held on the output queue and it does not print.

#### Example 2

Assume:

- The application calls for FGID 51 (Matrix Gothic).
- The printer supports only downloaded font character sets.
- The fidelity parameter value is \*CONTENT.

In this example, the spooled file will print. i5/OS substitutes a font character set (C0S0CR10, Courier Roman 10 pitch) for FGID 51. This is not an exact match. The system matched (as closely as possible) the font character to the FGID specified in the application.

**Note:** In this example, if the fidelity parameter is \*ABSOLUTE, the spooled file is HELD.

### Font global identifiers:

Another method of naming a font is by a font global identifier (FGID). An FGID names a type family and a typeface.

FGIDs are identified by a number such as 3, 8, or 11.

There is a different FGID assigned for the same type family but different typeface. For example, a Courier Roman Medium 10 pitch (characters-per-inch) is FGID 11 and Courier Roman Bold 10 pitch (characters per inch) is FGID 46.

The following figure shows an example of FGID 11. The text in the box represents how data prints if your application uses FGID 11.

```
FGID 11 is a monospaced courier font that
will print 10 characters per inch.
```

RV2H331-1

Printers with resident fonts use FGIDs to name the resident fonts. Depending on the technology used with the printer, resident fonts can be stored on font cards, diskettes, in the memory of the printer, or mechanically on a font element or daisy wheel.

The following printers have resident fonts:

- 3112 (can also accept downloaded fonts)
- 3116 (can also accept downloaded fonts)
- 3130 (can also accept downloaded fonts)
- 3160 (can also accept downloaded fonts)
- 3812 (can also accept downloaded fonts)
- 3816 (can also accept downloaded fonts)
- 3930
- 3912, 3916, or 4028 (can also accept downloaded fonts)
- 3935 (can also accept downloaded fonts)
- 4214
- 4224
- 4230
- 4234 Models 8 and 12
- 4247
- 4312 (can also accept downloaded fonts)
- 4317 (can also accept downloaded fonts)
- 4324 (can also accept downloaded fonts)
- 5219
- 5224
- 5225
- 6400
- 6408
- 6412
- 3900–AFCCU (can also accept downloaded fonts)
- Infoprint 20 (can also accept downloaded fonts)
- Infoprint 32 (can also accept downloaded fonts)
- Infoprint 3000 (can also accept downloaded fonts)

- Infoprint 4000 (can also accept downloaded fonts)

To find out which fonts are supported by a printer, check the reference manual for that printer.

### Selecting resident fonts

Select a resident font to use with an application program by specifying an FGID value on the FONT parameter of the printer file.

### Font substitution

Substitution can be one FGID for another, an FGID for a font character set, or a font character set for an FGID.

#### Example 1

Assume:

- Your application calls for a font character set (FNTCHRSET specified on the printer file), for example, C0S0CR10 for Courier Roman medium 10 pitch.
- The printer is a 4224 and has resident fonts identified by FGIDs.
- FGID 11 will be substituted for C0S0CR10 and sent to the printer.

In this example, the system substitutes a font that is resident on that printer.

#### Example 2

Assume:

- Your application calls for a font (specified on the FONT parameter of the printer file). The font specified is font 26 (Gothic Matrix, Roman medium 10 pitch) and the printer is a 3812.
- You decide to print the document on a 4019 printer. Font 26 is not supported on the 4019.

In this example, the system substitutes font 11 (Courier, Roman medium 10 pitch).

For more information about such substitutions, see Printer font support.

#### Example 3

Assume:

- Your application uses a font (specified on the FONT parameter of the printer file). The font specified is font 40 (Gothic, Roman medium 10 pitch).
- The printer you are going to print on supports only font character sets (for example, a 3827).

In this example, the system substitutes font character set C0D0GT10 (Gothic Text, Roman medium 10 pitch).

For more information about such substitutions, see Mapping printer-resident to host-resident code pages.

### Related reference:

“Printer font support” on page 317

This table contains font ID substitution information which identifies which fonts are supported by certain printers. For example, if your application specified a certain font ID that your printer did not support, you can find out which printers support that font and route your printed output to a printer that supports that font ID.

“Mapping printer-resident to host-resident code pages” on page 375

The table in this topic can help you determine what host resident code page will be downloaded to a 3820, 3825, 3827, 3829, 3831, 3835, or 3900 printer when your spooled file refers to a registered code page identifier (ID) instead of a host-resident code page.

**Code pages:**

Code pages come in two types.

- Code page (stand-alone)
- Character set and code page combination (referred to as a CHRID).

Code pages are groups of characters. Within a code page, there are unique hexadecimal identifiers assigned to each of the characters.

As you enter your text at a computer keyboard, each keyboard character is translated into a code point. When the text is printed, each code point is matched to a character ID on the code page you specified. The character ID is then matched to the image (raster pattern) of the character in the character set you specified.

Some of these characters can be repeated in different code pages and have a different hexadecimal identifier assigned to them. Conversely, the hexadecimal identifier can be the same, but the characters will be different. Therefore, if you have applications that use certain characters contained in only one particular code page, it is important to know which code page you are using.

Below is a diagram of two code pages: code page 37 and code page 285. They are printed in a 10-characters-per-inch (courier 10) font. Notice the different character that occurs at code point hex X'5B'. One is a US dollar (\$) sign and the other an English pound or currency sign. This example shows that different characters will print depending on the code page you specify even though you are using an identical font style.

Code page 37 with courier 10

		Code point 5B																
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
4			â	ä	á	ã	ä	ç	ñ	é	<	(	+					
5		&	é	ê	ë	è	í	î	ï	í	ß	!	£	*	)	;	~	
6		-	/	Â	Ã	Ä	Å	Ä	Ç	Ñ	]	,	%	_	>	?		
7		ø	É	Ê	Ë	È	Í	Î	Ï	Ì	`	:	#	•	'	=	"	
8		Ø	a	b	c	d	e	f	g	h	i	«	»	đ	ý	þ	±	
9		°	j	k	l	m	n	o	p	q	r	ä	ö	æ	,	Æ	°	
A		µ	~	s	t	u	v	w	x	y	z	ı	ł	Đ	Ÿ	Ɔ	®	
E	\	S	T	U	V	W	X	Y	Z	:	Ô	Õ	Ö	Ó	Ô			
F		0	1	2	3	4	5	6	7	8	9	'	Û	Ü	Ù	Ú		

Code page 285 with courier 10

		Code point 5B																
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
4			â	ä	á	ã	ä	ç	ñ	£	<	(	+					
5		&	é	ê	ë	è	í	î	ï	í	ß	!	£	*	)	;	~	
6		-	/	Â	Ã	Ä	Å	Ä	Ç	Ñ	]	,	%	_	>	?		
7		ø	É	Ê	Ë	È	Í	Î	Ï	Ì	`	:	#	•	'	=	"	
8		Ø	a	b	c	d	e	f	g	h	i	«	»	đ	ý	þ	±	
9		°	j	k	l	m	n	o	p	q	r	ä	ö	æ	,	Æ	°	
A		µ	~	s	t	u	v	w	x	y	z	ı	ł	Đ	Ÿ	Ɔ	®	
E	\	S	T	U	V	W	X	Y	Z	:	Ô	Õ	Ö	Ó	Ô			
F		0	1	2	3	4	5	6	7	8	9	'	Û	Ü	Ù	Ú		

RV2H330-1

**Stand-alone code pages:**

Code pages supply consistent or similar characters across systems. For example, a document created at one location using a specific code page could be sent to a different location, printed on a different model printer, and still look the same.

Code pages must be downloaded to the printer for use.

The following printers accept downloaded code pages:

- 3112 (has resident fonts also)
- 3116 (has resident fonts also)
- 3130 (has resident fonts also)
- 3160 (has resident fonts also)
- 3812 (has resident fonts also)
- 3816 (has resident fonts also)
- 3820
- 3825
- 3827
- 3828 (MICR printer)
- 3829
- 3831
- 3835
- 3900–001
- 3900–AFCCU (has resident fonts also)
- 3900
- 3912 (has resident fonts also)
- 3916 (has resident fonts also)
- 3930 (has resident fonts also)
- 3935 (has resident fonts also)
- 4028 (has resident fonts also)
- 4312 (has resident fonts also)
- 4317 (has resident fonts also)
- 4324 (has resident fonts also)
- Infoprint 20 (has resident fonts also)
- Infoprint 32 (has resident fonts also)
- Infoprint 3000 (has resident fonts also)
- Infoprint 4000 (has resident fonts also)

### **Naming convention for code pages**

Like character sets, code pages are named in a number of ways. One way is with a code page name. These code pages are downloaded to the printer. The code page name can be up to 8 characters long. Code page names are used with character set names for printing on printers such as the 3820, 3825, 3827, or 3835.

Another way is with a code page global identifier (CPGID). CPGIDs are printer-resident code pages and have numbers for names (for example, 259 or 500). Generally, printers with resident fonts use CPGIDs to name the printer-resident code pages. CPGIDs are also used within CHRIDs.

For example, in the code page name T1V10500:

- T** The **T** means that this object is a code page.
- 1** This is always a 1.
- V1** The **V1** means that this is version 1 of this code page.

**0500** The **0500** is the code page name, number, or category. In this example 500 is the code page name.

### **Selecting code pages**

Code pages are selected by specifying a certain value for the code page (CDEPAG) parameter of the printer file.

If you choose to use code pages with your applications, you must also specify a font character set (by providing a value for the FNTCHRSET parameter of the printer file being used).

### **Substituting code pages**

Substitution of code pages occurs for the following reasons:

- The application specifies a code page that is resident on a printer and the printer being used does not have resident code pages.
- The application specifies a code page that is resident on the host system (System i platform) and the printer being used has resident code pages (not capable of accepting downloaded code pages).
- The job requesting the code page is not authorized to it.
- The code page cannot be found.
- The job is not authorized to the library where the code page is stored.

### **Character set and code page combinations:**

This type of code page is made up of a specific graphic character set and a specific code page and is referred to by a character identifier (CHRID).

These graphic CHRIDs are used for fonts that are resident on the printer. They are used in conjunction with a font ID to obtain a resident font.

The following printers support CHRIDs:

- 3112
- 3116
- 3130
- 3160
- 3812
- 3816
- 3900–AFCCU
- 3912
- 3916
- 3930
- 3935
- 4028
- 4214
- 4224
- 4230
- 4234
- 4247
- 4312
- 4317
- 4324

- 5219
- 5224
- 5225
- 6400
- 6408
- 6412
- Infoprint 20
- Infoprint 32
- Infoprint 3000
- Infoprint 4000

### **Naming convention for CHRIDs**

The names of CHRIDs are made up of two elements: graphic character set and code page. These two elements define a collection of characters. Below is an example of the multinational CHRID 697-500.

**697** This is the name of the graphic character set.

Some graphic character sets identify a character set that is a subset of the code page. Others identify a character set that is equivalent to the code page.

**500** This is the name of the code page.

### **Selecting CHRIDs**

CHRIDs are selected by specifying a certain value for the CHRID parameter of the printer file. Additionally, a font ID value must be specified for the FONT parameter on the printer file.

### **Substituting CHRIDs**

If the CHRID is not available on the printer your application is using, the system will substitute the CHRID that most closely matches the one requested by the application.

### **Coded fonts:**

A coded font is the pairing of a font character set and a code page. Coded fonts allow users to specify a font character set and a code page with one value specified on the printer file.

Coded fonts available on the System i platform can be viewed by using the Work with Font Resources (WRKFNTRSC) command.

Coded font names are read by the system and then translated to a font character set and a code page. These two elements are then sent to the printer.

### **Naming convention for coded fonts**

Unlike other uniformly spaced and mixed-pitch font components, coded font names are generally shortened by excluding the origin and reserved characters (the first two characters of their name). This is necessary because some Advanced Function Presentation (AFP) licensed programs accept only 6 characters for coded font names. However, some applications can use coded fonts named with 6 or 8 characters.

Coded font names on the System i platform are 6 or 8 characters long. Each character or group of characters tells something about the coded font.

For example, in the coded font name X0GT10:

**X0** The X0 means that this object is a coded font.

**XZ** The XZ means that this object is an outline coded font.

**GT10** The **GT10** indicates the type family, typeface, and pitch for uniformly spaced and mixed-pitch fonts. In this example the GT10 means that this font character set is a Gothic Text style and the characters are 10 pitch or 10 characters per inch.

To find out which font character set and code page make up a coded font name, use the Work with Font Resources (WRKFNTRSC) command. This command allows you to specify the font resource to be worked with, the library it is in, and the attribute (coded font).

Additional naming conventions have been adopted to more explicitly name the code page used with a character set.

For more information about coded fonts, see the manual *About Type: IBM's Technical Reference for 240-Pel Digitized Type*, GS544-3516.

### **Selecting coded fonts**

A coded font is selected by specifying the coded font name as the value on the coded font (CDEFNT) parameter of the printer file.

You can use the Work with Font Resources (WRKFNTRSC) command to view the coded fonts that are available on the system.

### **Substituting coded fonts**

No substitution of coded fonts takes place on the System i platform. If the coded font is not available, the document will not print.

You can use the MAPIGCFNT parameter on the CRTPSFCFG and CHGPSFCFG commands to specify that coded font names of the form X0nnnnnn should be mapped to XZnnnnnn. If the XZnnnnnn coded font is found it is used; if it is not found, the X0nnnnnn coded font will be used.

## **Double-byte character set support**

The i5/OS operating system supports printing of double-byte characters.

Before reading this section you should be familiar with double-byte character set (DBCS) support in general. For more information about DBCS support, see Double-byte character set support in the Database file management topic collection and Work with DBCS data in the Globalization topic collection.

### **Related information:**

Double-byte character set support

Work with DBCS data

### **Special DBCS printer functions**

DBCS printers offer functions such as character rotation, character expansion, and condensed printing.

### **Related information:**

Processing double-byte characters

### **Character rotation:**

The DBCS printers can rotate double-byte characters 90 degrees counterclockwise before printing so that the printed output can be read vertically.



For example, the character rotation function takes characters as shown:

文字を縦向きにする  
HRSL5302-2

and rotates them so that you can read the printed characters vertically:

文字を縦向きにする  
HRSL5303-2

Specify character rotation with the IGCCHRRTT parameter on the Create Printer File (CRTPRTF), Change Printer File (CHGPRTF), and Override with Printer File (OVRPRTF) commands, or with the DDS keyword IGCCHRRTT for the file you are printing. This function rotates only double-byte characters. It does not rotate alphanumeric characters.

#### Character expansion:

DBCS printers can expand characters to twice their normal width or their normal height. Specify the character expansion with the DDS character size (CHRSIZ) keyword.

For example, if you specify the value CHRSIZ(2 1), the following characters:

文字を横倍角にする  
HRSL5304-2

文字を横倍角にする  
HRSL5305-2

To print twice as wide and twice as high, you would specify CHRSIZE (2 2).

#### Condensed printing:

The DBCS printers can print 20 double-byte characters per 3 inches so that more double-byte characters fit on a printed line.

For example, the following characters shown:

文字の密度を変更する  
HRSL5306-2

when condensed, are printed as:

文字の密度を変更する  
HRSL5307-2

Specify condensed character printing with the IGCCPI parameter on the Create Printer File (CRTPRTF), Change Printer File (CHGPRTF), and Override with Printer File (OVRPRTF) commands.

### Horizontal and vertical lines:

The record-level define line (DFNLIN) keyword in DDS can be used to draw a horizontal or vertical line (also known as a grid line). A horizontal line is drawn at the bottom of the character spaces. A vertical line is drawn on the left edge of the character spaces. You can draw horizontal lines and vertical lines to form boxes on the printed output.

The DFNLIN keyword is valid for SNA character string (SCS) printers.

The maximum number of lines that can be printed at one time is 200. The maximum number of active vertical lines (vertical lines currently being printed on the page) is 150. More than 200 DFNLIN keywords can be used per page if all the define lines from the previous records have been printed.

Output considerations at run time:

- Spacing and skipping are processed before the DFNLIN keyword. If you space or skip past the start of a line, that line will be truncated (or not printed if the end of the line is passed also).
- A horizontal line cannot extend over a page boundary. A horizontal or vertical line cannot be started over a page boundary.
- The start line value specified on the DFNLIN keyword cannot be larger than the page length value specified on the PAGESIZE parameter on the printer.
- The start position value specified on the DFNLIN keyword cannot be larger than the page width value specified on the PAGESIZE parameter.
- The sum of the length and the start line value for a vertical line (specified on the DFNLIN keyword) cannot be larger than the page length specified on the PAGESIZE parameter.
- The sum of the length and the start position value for a horizontal line (specified on the DFNLIN keyword) cannot be larger than the page width specified on the PAGESIZE parameter.

A diagnostic message is sent whenever the PAGESIZE and DFNLIN values together cannot correctly process a request.

The following is an example of using DFNLIN to produce lines in a table:

社員番号	氏名
010001	田中一郎
010002	鈴木一雄

HRSL5308-1

### Shift-control character printing:

The DBCS printers can print shift-control characters in one of these ways.

- Suppress the shift-control characters so that these characters do not occupy any space on printed output.
- Print one blank in the space occupied by each shift-control character.
- Print two blanks in the space occupied by the shift-in character and suppress the shift-out character.

Specify how to print shift-control characters on the DBCS printers with the IGCSOSI parameter on the CRTPRTF, CHGPRTF, and OVRPRTF commands.

For data printed using the DBCS-graphic data type with an externally described printer file, shift-out/shift-in processing is not used. Instead, the shift control characters added to the DBCS data do not occupy any space on the printed output.

## Double-byte character printing considerations

When you print double-byte data, take these into consideration.

### Extended character printing:

To make sure that extended characters are processed, specify extended character processing. Otherwise, the system prints only basic double-byte characters.

See Processing double-byte characters in the Database file management topic collection for instructions on specifying extended character processing and for information about the effects of such processing.

### Condensed printing:

When specifying condensed printing on DBCS printers (by specifying IGCCPI(\*CONDENSED) on the CRTPRTE, CHGPRTE, or OVRPRTE command), consider these factors.

- Specify the page width in alphanumeric print positions with the CPI parameter. Although the record to be printed can contain 88 double-byte characters (which use 176 print positions in normal printing) and the page width is 132 print positions, the double-byte data should print correctly in condensed mode.
- For program-described printer files, data might not be printed in the correct position on the page. The system does not perform boundary alignment for alphanumeric data in printed records. When double-byte and alphanumeric data are printed on the same line, the printer begins printing alphanumeric data in the first space following the double-byte data. As a result, characters might not be printed on the correct position on the page.
- For DDS files, the printer begins printing alphanumeric data in the first position following the double-byte data, when double-byte and alphanumeric characters are mixed in a field defined with data type O (double-byte-capable). As a result, data might not be printed on the correct position on the page. This situation does not arise when the field contains only double-byte data or when alphanumeric data is printed in a field defined with an alphanumeric data type.

### Page width:

Page width is specified as the second value of the PAGESIZE parameter on the CRTPRTE, CHGPRTE, or OVRPRTE command. The correct page width depends on the printer being used and the characters per inch (CPI) specified for the printer file.

When describing printer files used with printers configured as a 5553 Printer, select a page size in the range based on characters per inch:

CPI	Page-width range
10	1 through 136
12	1 through 163
13.3	1 through 181
15	1 through 204
18	1 through 244
20	1 through 272

Choose one of the following (depending on the CPI selected) when describing printer files used with printers configured as a 5583 Printer:

**CPI Page-width range**

10	1 through 132
12	1 through 158
13.3	1 through 176
15	1 through 198
18	1 through 236
20	1 through 264

**Unprintable double-byte characters:**

A double-byte character is considered unprintable if its double-byte code is not in the valid range or if its double-byte code is valid but does not have a character image defined.

You can specify that the system replace unprintable double-byte characters by specifying the replace unprintable character parameter (RPLUNPRT(\*YES)) on the CRTPRTE, CHGPRTE, or OVRPRTE command, but you cannot choose the replacement character.

Although you cannot choose the replacement character for unprintable double-byte characters, you can choose the replacement character for unprintable alphanumeric characters. To improve system performance, select a blank ( ) as the replacement character for unprintable alphanumeric characters.

When the system finds an unprintable double-byte character during printing, the following happens:

- If you specify RPLUNPRT(\*YES), the system does not send a message when it finds unprintable characters. Instead, the system prints unprintable extended characters as either the double-byte underline ( ) when you specify extended character processing, or as an undefined character when you do not specify extended character processing.

For Japanese printers, the default symbol used is:



For Chinese and Korean printers, the default symbol used is the underscore.

The system prints unprintable basic double-byte characters as double-byte blanks.

- If you specify RPLUNPRT(\*NO), the system sends an inquiry message when it finds unprintable characters. You have the following options:
  - Hold the spooled file.
  - Continue printing where the unprintable character was encountered. If you continue printing, the system sends the inquiry message that you just received. It is sent each time the system finds an unprintable character, regardless of your response to the first message.
  - Continue printing by specifying a page number where printing should continue. When the system finds subsequent unprintable characters, it processes the characters as if the file were specified with RPLUNPRT(\*YES). See the item in this list about RPLUNPRT(\*YES) for a description of how the system processes these characters.

If the system finds invalid double-byte code, it stops processing double-byte extended characters and prints them as the undefined character.

**Double-byte data in an alphanumeric field:**

If you try to print double-byte data in a field that is described in DDS as alphanumeric, the system interprets the data as alphanumeric.

What happens depends on whether the printer being used is an alphanumeric or DBCS printer, and on the status of the replace-unprintable-characters option. This condition is a special case described under Improperly indicated DBCS files in the Database file management topic collection.

**Related information:**

Improperly indicated DBCS files

**Spanned lines:**

If a printed line of double-byte data exceeds its specified page width (line length), the system tries to continue printing the data.

To do this, the system ignores the FOLD parameter on the CRTPRTF, CHGPRTF, and OVRPRTF commands. As a result, the system might not print the double-byte data as you expected and the following occurs:

- If a record to be printed exceeds the page width, the printer wraps the data (continues printing the record on the next line). Because the system is not aware that the data is wrapped, the system does not skip lines and start new pages properly. A new page might start in the middle of a record.
- The printer does not split double-byte characters when there is not enough room at the end of a line and a field of double-byte data is continued on a second printed line, even if you specified the CHRISZ keyword. Instead, the system leaves a blank space on the first line where the character is printed and continues printing the complete character on the next line.

**Spanned pages:**

If data from a printed DBCS field spans to a second page, the system inserts a shift-in character at the beginning of each printed page of double-byte data, shifting the data out of DBCS mode. The printed data that follows does not make sense unless the data on the second page begins with a shift-out character.

To avoid this problem, break double-byte data fields that might span pages into several smaller fields.

**Print key use:**

If you want to print a display containing double-byte data by pressing the Print key, make sure that the associated display file or printer file is a DBCS file. If neither is a DBCS file, the display will not print properly.

One way to make sure that either the display or printer file is a DBCS file is to override the file using the OVRDSPF or the OVRPRTF command. For example, to override the system-supplied default printer file (the printer file used to print displays that are printed by pressing the Print key), enter:

```
OVRPRTF FILE(QSYSPRT) IGCDTA(*YES)
```

**Notes:**

1. If you do not plan to use double-byte data, do not change the printer file QSYSPRT to a DBCS file with a CHGPRTF command. This printer file is used to print a variety of system data, including alphanumeric data. A system performance degradation results if QSYSPRT is a DBCS file and it is processing only alphanumeric data.
2. If the Print key is used to print the image of a display containing DBCS-graphic data type fields, the system inserts shift-out and shift-in (SO/SI) characters around the graphic data. Depending on the IGCSOSI printer file value, the SO/SI characters can print as blanks, causing the printed alignment to be different from what was displayed.

**Related concepts:**

“Printer file overrides” on page 6

Overriding files (printer files, display files, diskette files, database files, and tape files) can be done by commands, from CL programs, or from high-level language programs. Overrides can be called from different levels (programs calling another program). This article discusses overriding printer files.

### **5553 printer end-of-forms:**

If you send the ignore (I) reply to the end-of-forms message that you receive when using continuous forms on the 5553 Printer, and if the printer has already printed within 2-1/2 inches of the bottom of the page, the system might not start printing subsequent pages where expected.

To avoid this problem, do the following when you receive the end-of-forms message:

1. Remove the current form from the tractor feed.
2. Insert new forms.
3. Align the first form to the first line.
4. Press the CANCEL button on the printer.
5. Press the SELECT button on the printer.
6. Respond to the end-of-forms message:
  - a. For spooled files, specify the page on which you want to continue printing when you enter a response to the message. Determine which page to continue printing as follows:
    - 1) If no data was printed on the last 2-1/2 inches of the last form, enter the number of the next page to be printed.
    - 2) If data was printed on the last 2-1/2 inches of the last form, enter the number of the last page printed. Reprinting the page ensures that all of the data is printed.Use the Work with Writer (WRKWTR) command to find out approximately which page was last printed. The WRKWTR command displays the number of pages that the writer has currently printed.
  - b. For direct printer output, enter RETRY to reprint the last page printed. This ensures that all of the data is printed.

### **Double-byte data printed on alphanumeric printers:**

Printing DBCS output on an alphanumeric printer can result in degradation of system performance.

In addition, the following events occur using printer files that are indicated to be DBCS-capable through DDS or with the IGCDTA parameter:

- For direct printer output, the system prints the file and sends a diagnostic message describing the situation to your program message queue.

Instead of printing double-byte data, the system prints double-byte characters as underscores ( `__` ) and prints shift-control characters as blanks (  ). Although the system does not print the individual double-byte characters, they are correctly stored in the system.
- For spooled printer output, the system sends an inquiry message to the message queue named on the Start Printer Writer (STRPRTWTR) command. This message lets you do the following:
  - Continue printing. When you continue printing, the system prints the file but does not print double-byte characters within it. Instead, the system prints double-byte characters as underscores ( `__` ) and prints shift-control characters as blanks (  ). Although the system does not print the individual double-byte characters, they are correctly stored in the system.
  - Hold the spooled file so that you can transfer it to an output queue used only for DBCS output.
  - Cancel printing altogether.

### **Related tasks:**

“Moving a spooled file” on page 214

To move the spooled file (printer output) from one output queue to another output queue, use either of these methods.

### **DBCS spooled support**

You can create separate output queues for double-byte and alphanumeric data. This might improve throughput (the rate at which the system processes work) because the system can process alphanumeric data more quickly than it can process double-byte data.

#### **Applying overrides in printing**

When starting a job, consider adding the OVRPRTF command to the initial program of the job:

```
OVRPRTF FILE(QSYSPRT) IGCDTA(*YES)
```

Override the printer file (QSYSPRT) to make it capable of printing double-byte data and to ensure that DBCS output printed as the result of pressing the Print key is printed properly.

#### **Related concepts:**

“Printer file overrides” on page 6

Overriding files (printer files, display files, diskette files, database files, and tape files) can be done by commands, from CL programs, or from high-level language programs. Overrides can be called from different levels (programs calling another program). This article discusses overriding printer files.

#### **Related information:**

Create Output Queue (CRTOUTQ) command

### **3130 printer resident font support**

When printing to a 3130 printer (Release 2 of printer microcode required), you can specify to use the double-byte fonts resident in the 3130 printer.

The QPQCHGCF program provides a way for you to indicate whether a particular section of a double-byte coded font is resident in the printer or should be downloaded.

#### **Related reference:**

“QPQCHGCF program” on page 394

The QPQCHGCF program provides a way for you to indicate whether a particular section of a double-byte coded font is resident in the printer or should be downloaded.

## **Internet Printing Protocol server**

The IBM Internet Printing Protocol (IPP) server for i5/OS allows IPP clients (such as Windows PCs) to submit and manage print jobs to any System i printer.

Print jobs are passed to an attached printer and might involve further processing with applications such as host print transform or PSF. The target printer does not need to support IPP.

The IPP server supports IPP version 1.1. The following required operations are supported:

- Print-job
- Validate-job
- Get-printer-attributes
- Get-jobs
- Cancel-job
- Get-job-attributes

Additionally, the following optional IPP operations are supported:

- Pause-printer

- Resume-printer
- Purge-jobs
- Hold-job
- Release-job
- Restart-job

The IPP server can be configured to provide security features for user authentication and encryption of print data. For more information about how to configure the IPP Server, see Set up the IPP server.

**Related concepts:**

“Setting up the IPP server” on page 127

You can administer and configure the Internet Printing Protocol (IPP) server using the IBM IPP Server Administrator for iSeries, a browser-driven graphical user interface.

## Network print server

The i5/OS network print server is the host or print server for print clients.

### Network print server objects

The network print server provides client access to i5/OS print objects and resources.

The following table lists the network print server objects and the actions that can be requested to be performed on these objects.

Objects	Actions
Spooled file	Create, seek, open, read, write, close, hold, release, delete, move, send, call exit program, change attributes, retrieve message, answer message, retrieve attributes, and list.
Writer job	Start, end, and list.
Printer device	Retrieve attributes and list.
Output queue	Hold, release, purge, list, and retrieve attributes.
Library	List.
Printer file	Retrieve attributes, change attributes, and list.
Network print server	Change attributes and retrieve attributes.

### How the network print server is accessed

Before a client can communicate with the network print server, a communications session must be established. A communications session is a logical connection between two systems through which a client program on a local system can communicate with a server program on a remote system.

After a communications session has been established, the client can start the network print server by sending a special record, called a program start request (PSR), to the System i platform. The communications session you use can use different protocols. Refer to the appropriate documentation for that communications type for information about using program start requests (PSRs).

After the network print server program has been started using a PSR, a communications transaction has been started. A communications transaction is a logical connection between two programs on a communications session. After this communications transaction starts, data can be exchanged between the client and the network print server.

### Prestart jobs and the network print server



Prestart jobs provide increased performance between a program start request (PSR) initiated by a client and the network print server. Prestart jobs are defined within a subsystem. Prestart jobs become active when that subsystem is started or they can be controlled with the Start Prestart Job (STRPJ) and End Prestart Job (ENDPJ) commands.

The program name contained in the PSR from the client must be defined in the prestart job entry. This is how the PSR attaches itself to a prestart job and therefore achieves better performance.

The network print server has prestart jobs defined in the QBASE and QCMN subsystems. The number of prestart jobs that automatically start to support the network print server is small and thus saves system resources.

### Monitoring prestart jobs for the network print server

Monitoring prestart jobs for the network print server in the QBASE or QCMN subsystem can be done using the Display Active Prestart Jobs (DSPACTPJ) command. For the network print server, you need to know the subsystem your prestart jobs are in (QBASE or QCMN) and the program for which the prestart jobs are started (QNPSERVER).

This command provides the following information:

- Prestart jobs:
  - Current number
  - Average number
  - Peak number
- Prestart jobs in use:
  - Current number
  - Average number
  - Peak number
- Program start requests:
  - Current number waiting
  - Average number waiting
  - Peak number waiting
  - Average wait time
  - Number accepted
  - Number rejected

**Note:** The network print server never rejects a PSR. All PSRs are handled, or they are queued and use the next available prestart job.

### Changing prestart job entries

The information presented for an active prestart job can be refreshed by pressing the F13 key while on the DSPACTPJ display. Of particular interest is the information about PSRs. This information can indicate to you whether or not you need to change the available number of prestart jobs. If you have information indicating PSRs are waiting for an available prestart job, you can change prestart jobs using the Change Prestart Job Entry (CHGPJE) command.

Following is an example of a prestart job entry in the QBASE subsystem for the network print server.

```
SBSD(QSYS/QBASE) +  
PGM(QSYS/QNPSERVER) +  
USER(QUSER) +  
STRJOBS(*YES) +  
INLJOBS(1) +
```

```

THRESHOLD(1) +
ADLJOBS(3) +
MAXJOBS(*NOMAX) +
JOB(*PGM) +
JOB(*USRPRF) +
MAXUSE(200) +
WAIT(*YES) +
POOLID(1) +
CLS(QGPL/QCASERV *CALC *NONE *CALC)

```

If the PSRs were not being acted on fast enough, you could:

- Increase the Initial number of jobs (INLJOBS) parameter value.
- Increase the Additional number of jobs (ADLJOBS) parameter value.

When the Threshold (THRESHOLD) parameter value is reached, the additional number of jobs are prestarted.

The key is to match the number of prestart jobs to the number of PSRs that are being sent. Keeping this association as close to one-to-one ratio ensures peak system performance.

Additional prestart jobs can be started if the number of program start requests to the network print server exceeds the number of available prestart jobs.

### Exit points and the network print server

An *exit point* is a specific point in a system function or program where control can be passed to an installation specified program or programs. These programs are called exit programs because they are providing an exit from the normal processing of the system function or program. The exit programs are typically programs created by users.

The following table contains the IBM-registered exit points that can be used with the network print server.

Type of support	Exit point name	Exit point format
Entry	QIBM_QNPS_ENTRY	ENTR0100
Spooled file	QIBM_QNPS_SPLF	SPLF0100

The functions provided by these exit points are:

- Exit point QIBM\_QNPS\_ENTRY  
This exit point is used by exit programs that want to control which users can access the network print server.
- Exit point QIBM\_QNPS\_SPLF  
This exit point is used by exit programs that process spooled files.

Exit programs must be registered with the i5/OS registration facility. Registration allows all users to associate exit programs with the exit points.

### i5/OS registration facility and the network print server

The registration facility provides storage and retrieval service for both i5/OS and non-i5/OS exit points and exit programs.

The network print server exit points are already registered with the i5/OS Registration Facility. To use these exit points, you have to register your exit programs with the i5/OS Registration Facility.

Exit programs are registered and deregistered using the following APIs:

- Add Exit Program (QUSADDEP) API  
Adds an exit program entry for a specific exit point.

**Note:** The network print server exit points do not define exit program data. When you register your exit program, you can provide exit program data. However, the data is not used by the network print server.

- Remove Exit Program (QUSRMEVP) API  
Removes an exit program entry for a specific exit point.

For information and examples on how to register exit programs and exit points, see the Application programming interfaces (APIs) topic collection.

### Verifying exit point and exit program registration

The Work with Registration Information (WRKREGINF) command displays information about exit points and exit programs. The command does not add, delete, or change exit points or exit programs within the registration facility. The APIs listed above must be used to initiate a change.

#### Related information:

Application programming interfaces

#### Exit point QIBM\_QNPS\_ENTRY:

Exit Point QIBM\_QNPS\_ENTRY is used by exit programs that want to control which users can access the network print server. The called exit program is passed two parameters. These parameters are pointers to structures in user domain storage. These structures, collectively, comprise format ENTR0100.

#### Required parameters

##### Required parameter group

1	Return code	Output	CHAR(1)
2	Server information	Input	CHAR(32)

#### Return code

OUTPUT; CHAR(1) The return code parameter is a pointer to a structure in user domain that contains the return code from the exit program. The return code is checked on return from the call to the exit program.

A return code value of hexadecimal F1 indicates that the network print server allows access and continues processing.

Any other return code value indicates that the network print server rejects access and stops processing. The exit program being called must log any specific errors.

The structure of the return code follows:

Offset		Type	Field
Dec	Hex		
0	0	CHAR(1)	Return code

#### Server information

INPUT; CHAR(32) The server information structure contains the user profile name, server identifier, format name, and server function identifier.

The structure of the server information follows:

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	User profile name
10	A	CHAR(10)	Server identifier
20	14	CHAR(8)	Format name
28	1C	BINARY(4)	Function identifier

## Parameter field descriptions

### Format name

The name of the format being used. For entry support the value is ENTR0100.

### Function identifier

This parameter identifies, within a server, the function of the exit point. For entry support, the value is hexadecimal 0802.

### Server identifier

This parameter identifies the i5/OS server being called. The value for the network print server is QNPSERVER.

### User profile name

The user, from System i Access for Windows, that is making the call to the network print server.

### Exit point QIBM\_QNPS\_SPLF:

Exit point QIBM\_QNPS\_SPLF is used by exit programs that process spooled files. The called exit program is passed four parameters. These parameters are pointers to structures in user domain storage. These structures, collectively, comprise format SPLF0100.

## Required parameters

### Required parameter group

1	Return code	Output	CHAR(1)
2	Server information	Input	CHAR(32)
3	Spooled file ID	Input	CHAR(61)
4	Spooled file exit program data	Input	CHAR(*)

### Return code

OUTPUT; CHAR(1) The return code is a pointer to a structure in user domain that contains the return code from the exit program. The return code is checked on return from the call to the exit program.

Any nonzero code indicates an error with the exit program. The exit program being called must log any specific errors. Processing by the network print server continues.

Offset		Type	Field
Dec	Hex		
0	0	CHAR(1)	Return code

### Server information

INPUT; CHAR(32) The server information structure contains the user profile name, server identifier, format name, and server function identifier.

The structure of server information is:

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	User profile name
10	A	CHAR(10)	Server identifier
20	14	CHAR(8)	Format name
28	1C	BINARY(4)	Function identifier

### Spooled file ID

INPUT; CHAR(61) The spooled file ID structure contains the job name, user name, job number, spooled file name, spooled file number, job system name, creation date, and creation time of a spooled file. This information uniquely identifies a spooled file on the IBM i.

The structure of a spooled file ID is:

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	Job name
10	A	CHAR(10)	User name
20	14	CHAR(6)	Job number
26	1A	CHAR(10)	Spooled file name
36	24	BINARY(4)	Spooled file number
40	28	CHAR(8)	Job system name
48	30	CHAR(7)	Creation date
55	37	CHAR(6)	Creation time

### Spooled file exit program data

INPUT; CHAR(\*) The spooled file exit program data information structure contains a 4 byte length and the spooled file exit program data. The contents of the spooled file exit program data is not known to the network print server. The receiving exit program knows the structure of the spooled file exit program data. For example, if the exit program was going to fax the spooled file specified by the spooled file ID, the spooled file exit program data can consist of a name, telephone number, building number, and office location.

The structure of the spooled file exit program data is:

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Length of the spooled file exit program data
4	4	CHAR(*)	Spooled file exit program data

### Parameter field descriptions

#### Creation date

The date that the spooled file was created in format CYYMMDD where C = century (0 = 19xx, 1 = 20xx), YY = year, MM = month, and DD = day.

| **Creation time**

The time that the spooled file was created in format HHMMSS where HH = hours, MM = minutes, SS = seconds.

**Format name**

The name of the format being used. For spooled file support, the value is SPLF0100.

**Function identifier**

This parameter identifies, within a server, the function of the exit point. For spooled file support, the value is hex 010D.

**Job name**

The name of the job that created the spooled file.

**Job number**

The number of the job that created the spooled file.

| **Job system name**

The name of the system where the job that created the spooled file ran.

**Server identifier**

This parameter identifies the IBM i server being called. The value for the network print server is QNPSERVER.

**Spooled file exit program data**

Additional information used by the exit program that is registered for exit point QIBM\_QNPS\_SPLF. The client application provides the spooled file exit program data.

**Spooled file name**

The name of the spooled file being requested.

**Spooled file number**

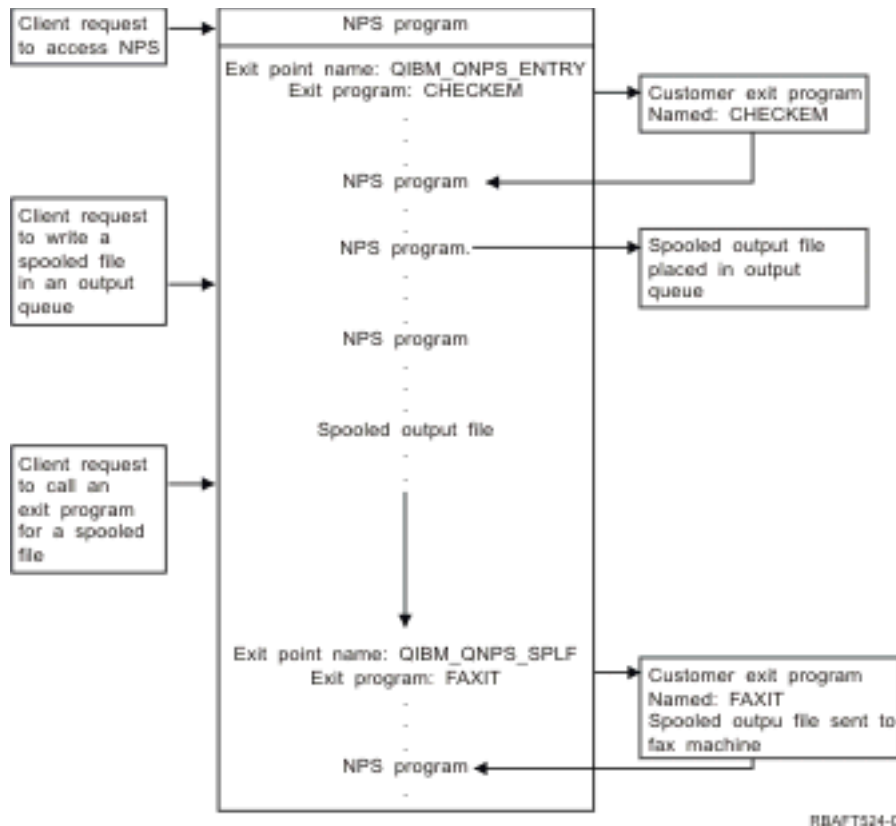
The number of the spooled file being requested.

**User profile name**

The user, from IBM i Access for Windows, that is calling the network print server.

**Use of network print server exit points**

The figure below provides an example of several requests from a client to the network print server using the exit points provided by the network print server. Those requests are: access the network print server, call a user-written exit program to verify access, put a spooled output file on an output queue, and call a user-written exit program that faxes a spooled output file.



## Communications network printing functions

You can use the communications network printing functions to send data within a communications network to be printed on a System i platform.

### 3270 printer emulation

When configuring 3270 printer emulation on i5/OS, you must decide which communications line protocol to use. System i products support both binary synchronous communication (BSC) and Systems Network Architecture (SNA) 3270 printer emulation. Therefore, the choice is normally dictated by the protocol supported by the System z communications software and controller.

#### BSC 3270 printer emulation

BSC 3270 printer emulation can be used with any System z platform that supports a 3274 control unit (Model 51C) in a BSC multipoint tributary network using a nonswitched line. BSC 3270 printer emulation supports the following features:

- Emulation on any line attached to a System i product.
- The ability to print information received from the host system on any printer normally supported by the System i product.
- Emulation of 3284, 3286, 3287, 3288, or 3289 printers.

#### SNA 3270 printer emulation

SNA 3270 printer emulation enables the System i product to connect to any existing SNA 3270 network. Any printers attached to the system print information received from the System z platform. The 3270 emulation session is linked to a printer file, which in turn is linked to a printer device. All 3270 printers are emulated as Logical Unit 3 printers with the exception of the 3287 and 3289 printers, which can be emulated as either LU 1 or LU 3 printers.

The difference between LU 1 and LU 3 printers is essentially the level of intelligence that the System z platform expects to find at the printer device. If the printer is emulated as an LU 1 printer, an LU-LU session can be established with the host. Depending on the host applications, such a printer can fully support SNA character string (SCS) data streams sent to it. If the System i printer emulates an LU 3 printer, only an LU-PU session can be established between it and the System z platform. This might cause unexpected results when using host print applications. For example, if the host application was expecting to communicate with an LU 1 printer, the printed output might not be correctly formatted for line spacing, page length, and so on.

## **Remote job entry printing**

Remote job entry (RJE) allows an i5/OS user to use z/OS applications and data in combination with i5/OS functions. RJE is part of the IBM Communications Utilities for System i licensed program (5761-CM1), which runs on all System i products that have communications capabilities installed.

When using RJE, the System i product functions as a remote workstation that submits jobs to a System z platform for processing under one or more of the following systems:

- MVS/SP JES2 (SNA and BSC)
- MVS/SP JES3 (SNA and BSC)
- VM/SP RSCS (BSC only)
- DOS/VSE POWER/VSE (SNA only)
- OS/VS1 RES (SNA and BSC)

RJE can communicate with the host system using either of the following line protocols:

- Systems Network Architecture (SNA) over point-to-point switched or nonswitched connections
- Binary synchronous communications (BSC) over point-to-point switched or nonswitched connections

For BSC, multiple user-defined RJE subsystems can be used concurrently on different lines, each subsystem operating independently of the other. For SNA, multiple subsystems can share the same line.

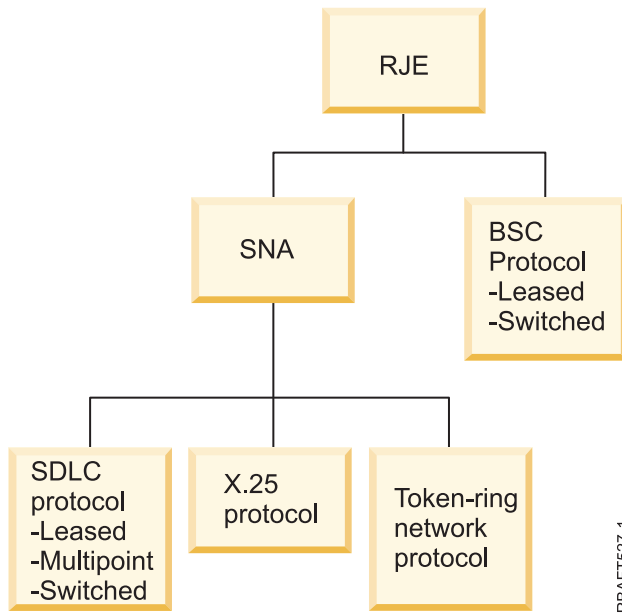
A writer (printer or punch) output stream can be directed to one of these:

- A printer output file (spooled or nonspooled)
- A physical database file member
- A diskette file (spooled)
- A DDM file
- User program

## **Communications line protocols for RJE**

A communications line physically connects a System i product to the host. It can be a remote link operating over a long distance or a cable connecting them on a local area network. The protocol used depends on whether you are in an SNA or a BSC environment.





RBAFT527-1

The physical interfaces supported by RJE follow.

	SDLC	X.25	Token-ring network	BSC
V.35	Yes	No	No	Yes
X.21bisV.35	Yes	Yes	No	Yes
RS232/V.24	Yes	Yes	No	Yes
X.21bisV.24	Yes	Yes	No	Yes
X.21	Yes	Yes	No	No
Medium Access Control (MAC) IEEE 802.5 / ISO 8802-5	No	No	Yes	No

### 3x74-attached printers

There is no 3270 emulation pass-through (*pipeline*) support for 3x74 remote-attached printers. Printing capability is limited to the Logical Unit 1 support. Printer data management and spool support is not provided for printers that use the 3270 Information Display System data-stream capability.

For i5/OS printing applications, the extent of the function is limited to that provided by a 3287 printer, regardless of the type and model of the printer actually attached. This is functionally equivalent to a 5256-003 printer. For example, a 4224 printer, which has all-points-addressable capability, only provides 3287 printer capability when attached to a 3x74 controller. However, if a 3284 printer is attached to the 3x74 controller, it does not provide the 6 or 8 lines per inch (LPI) support of the 3287 printer because it is not equipped with the necessary hardware features.

When using a 3x74 remote-attached printer (for example, a 4224) configured as a 3287 printer, make sure that the page length set on the printer hardware switch matches the length of the forms that are in the printer. If the length of the forms loaded does not match the printer hardware switch setting, the application will still print correctly on the forms. However, when the spooled file completes printing, the 4224 uses the default page size set on the printer hardware switches.

For example, assume your page length set on the printer hardware switches is 11 inches, your actual forms are 3 inches long, the spooled file just completed printing, and you press the Page Eject key to get the last printed form. By pressing the Page Eject key, you can advance the forms 11 inches instead of the 3 inches your form takes up.

### DBCS printer considerations

If the spooled file contains double-byte data (IGCDTA(\*YES) specified on the printer file), it can be printed on printers attached to a 3x74 controller.

However, the following attributes or DDS keywords are ignored if the target printer is attached to a 3x74 controller:

- DFNLIN (DDS keyword)
- IGCCHRRTT (DBCS attribute and DDS keyword)
- IGCCHRSIZ (DDS keyword)
- IGCCPI (DBCS attribute)
- IGCEXNCHR (DBCS attribute)

The IGCSOSI printer file parameter is processed in the i5/OS program and not in the printer being used to print the spooled file.

### Distributed data management printing

Distributed data management (DDM) enables users or application programs on the System i platform to access data files that reside on remote systems, and allows these remote systems to access data on the local system. This remote file access is transparent to the application program command or utility. Remote file requests are routed through a DDM file, which contains the name of the file on the remote system as well as the name of this system as it is known on the network.

There are, however, certain limitations to the DDM support, particularly where printing applications are concerned. For example:

- The IBM Query for i licensed program does not support DDM files.
- It is not possible to print on a remote System i platform using, for example, an Override with Printer File (OVRPRTF) command of the printer file QSYSPRT to the source DDM file. Because the print request is rejected by DDM, as shown by the following job log extract:

```
3 > OVRPRTF FILE(QSYSPRT) TOFILE(ITSCID03/DDMQSYSPRT)
DDM file DDMSRC in MYLIB uses remote file QSYS/QSYSPRT.
File DDMSRC in MYLIB not a database file.
Cannot open DDM file DDMSRC in MYLIB.
Function check. CPF4207 unmonitored by QWSGET at statement *N,
instruction X'0F9D'.
A function check was received while opening the print file.
The print operation used the default printer device file.
```

One way in which data is printed on a remote System i platform using DDM is as follows:

1. Create a DDM file on the source system using the Create DDM File (CRTDDMF) command.  
CRTDDMF FILE(MYLIB/DDMSRC) RMTFILE(QSYS/QSYSPRT)  
RMTLOCNAME(B20) TEXT('DDM file for remote printing on B20').
2. Create a physical file of the appropriate record length to receive the spooled data (80 bytes for print screen output, 132 bytes for normal spooled output), using the Create Physical File (CRTPF) command.  
CRTPF FILE(MYLIB/PFILE) RCDLEN(80) TEXT('CPYSPLF data file').
3. Create a similar file on the remote System i platform using the Submit Remote Command (SBMRMTCMD) command.  
SBMRMTCMD CMD('crtpf file (rlib/rfile) rcdlen(80)') DDMFILE(DDMSRC).

4. Copy the spool entry to the physical file using the Copy Spooled File (CPYSPLF) command, using the defaults (no control character).

```
CPYSPLF FILE(QSYSVRT) TOFILE(MYLIB/PFILE)
JOB(003049/USER/DSP06).
```

5. Use the Copy File (CPYF) command to copy the data from this physical file to the remote physical file created in step 3.

```
CPYF FROMFILE(MYLIB/PFILE) TOFILE(MYLIB/DDMSRC)
MBROPT(*ADD).
```

6. Use the SBMRMTCMD command once again to copy the data from the remote physical file to QSYS/QSYSVRT in order to create a spooled file on the remote System i platform's default output queue.

```
SBMRMTCMD CMD('copyf fromfile(rlib/rfile) tofile(qsysprt) mbropt(*add)')
DDMFILE(DDMSRC).
```

This procedure can be adapted to direct i5/OS printing to remote System/36 systems and System/38 systems with DDM installed. See the Distributed data management topic collection for details of the differences in DDM implementation on these other systems.

#### **Related information:**

Distributed database programming

### **Object distribution printing**

Object distribution provides the facility to send objects, messages, job streams, and spooled files across a suitably configured SNA link to another System i product or group of systems. File distribution (data, source, and spooled files) to System/36, System/38, and System z CICS<sup>®</sup> systems is also possible using i5/OS object distribution support.

### **Host print transform**

The host print transform is an i5/OS function that converts an SNA character string (SCS) or Advanced Function Presentation (AFP) data stream into an ASCII data stream. The ASCII data stream is then formatted and sent to an ASCII printer through one or more hardware connections, such as IBM i Access for Windows, or the 3477 or 3487 workstations.

This single location of the conversion allows for consistent ASCII printing through any of the hardware connections. The host print transform function can also be used to send the ASCII data stream to a printer or system, by using the Send TCP/IP Spooled File (SNDTCPSPLF) command, or to an ASCII LAN attached printer.

The host print transform is enabled by selecting certain values for parameters in the printer device description. If you need more detailed information about these parameters, see Local Device Configuration PDF, Create Device Description (Printer) (CRTDEVPRT) command topic, or Change Device Description (Printer) (CHGDEVPRT) command topic.

#### **Related information:**



Local Device Configuration PDF

Create Device Description (Printer) (CRTDEVPRT) command

Change Device Description (Printer) (CHGDEVPRT) command

### **Host print transform advantages**

The host print transform allows the data stream conversion of SNA character string (SCS) to American Standard Code for Information Interchange (ASCII) or Advanced Function Presentation (AFP) to ASCII to take place on the IBM i operating system instead of on an emulator.

Having the conversion take place on the IBM i operating system provides these advantages:

- Consistent output for most ASCII printers

The host print transform is capable of supporting many different types of ASCII data streams. For example, the Hewlett-Packard printer control language (PCL), the IBM personal printer data stream (PPDS), and the Epson FX and LQ data streams.

Having the conversion done on the IBM i operating system ensures that the resultant ASCII data stream provides the same printed output regardless of the emulator that the printer is physically attached to.

- 3812 SCS printer emulation

The host print transform is based on the 3812 SCS printer emulation of the IBM i Access for Windows workstation function. With the host print transform, all of the ASCII printers that are connected to a System i product can perform a 3812 SCS level of function.

**Note:** You cannot perform functions that your printer does not support. For example, you cannot print in 180-degree orientation when your printer only supports 0-degree and 90-degree orientations.

- Support for many different ASCII printers

Without host print transform, each emulator supports a limited number of ASCII printers. With the host print transform, most IBM printers and a large number of other printers are supported.

- Customized printer support

Workstation customization objects that come with the host print transform can be updated by the user to change or add characteristics to a particular printer. Also, if the host print transform does not have a workstation customization object for a printer you want to use, you can create your own.

- Support for the Image Object Content Architecture (IOCA) embedded in an SCS data stream.

Image output can be printed on PCL printers (Hewlett-Packard LaserJet) or PPDS laser printers (IBM 4019 or 4029).

This support allows you to print incoming faxes from Facsimile Support for i5/OS to PCL and PPDS laser printers. This support is available to all emulators discussed under host print transform.

Incoming faxes saved in a folder as RFT:DCA can be printed by using the Print Document (PRTDOC) command. You can also use the Print Fax (PRTFAX) command to direct the incoming faxes directly to an ASCII print device configured to use the host print transform. Host print transform has also been enhanced to do image resolution scaling which allows faxes to be printed in normal size.

- Support for conversion of double-byte SCS or AFP data stream into ASCII data stream

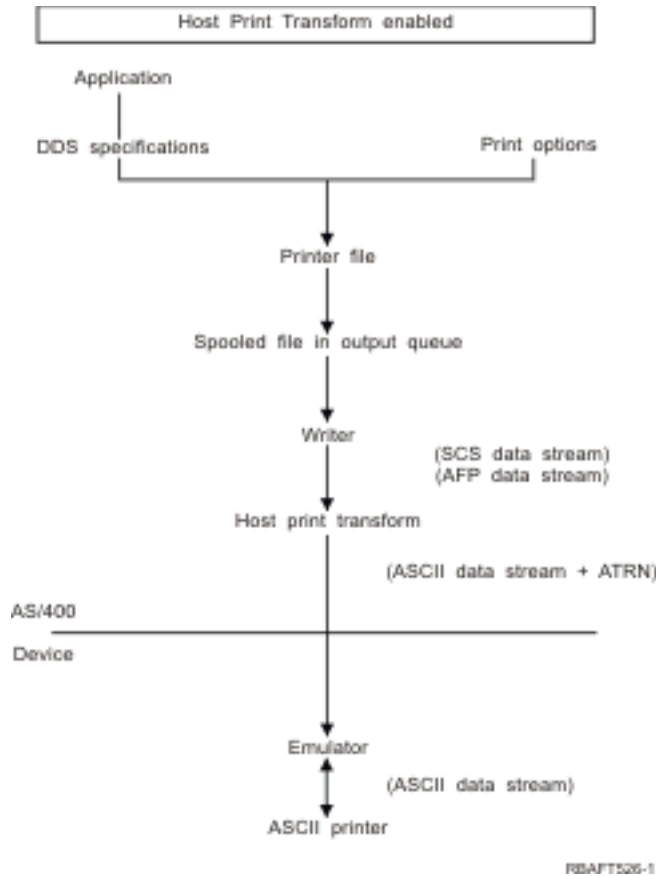
For the AFP-to-ASCII data stream conversion, there are additional advantages, such as support for AFP font, text, image, and bar code commands.

## How the host print transform function works

The host print transform converts the print data stream just before it is sent from the System i platform. The spooled file contains the print data and not the converted ASCII data.

**Note:** The host print transform works with jobs that are sent directly to the printer (SPOOL(\*NO) on the printer file).

The following figure shows the data stream origination, flow, and conversion for a spooled file when the host print transform is enabled.



The host print transform generates an ASCII printer data stream for a number of IBM and non-IBM printers. To generate the different ASCII data streams, the host print transform uses IBM i objects that describe characteristics of a particular ASCII printer. Using the SCS ASCII Transparency (ATRNL) command, it passes the ASCII data stream through the existing emulator. The existing emulator deletes the ASCII Transparency commands and passes the ASCII data stream that is generated by the host print transform to the personal printer. System i Access, PC5250, and IBM Personal Communications do not support partial ASCII transparency printing.

**Note:** The emulator must support the SCS ASCII Transparency (ATRNL) command to use it with the host print transform. All the emulators described in the Host print transform topic support the ATRNL command.

To enable the host print transform, you work with the printer device description.

### AFP-to-ASCII transform function

The AFP-to-ASCII transform function supports AFP font, text, bar code commands, as well as double-byte code pages and fonts.

The function works well with the following types of printers:

- IBM Network printers and IBM Infoprint printers
- HP laser and ink jet printers
- IBM pages printers

On other printers, images or bar code might not be supported by the AFP-to-ASCII transform function, and the text might not be positioned correctly.

## Bar codes

A bar code is a predetermined pattern of bars and spaces that represent numeric or alphanumeric information in a machine readable form. Bar codes are commonly used in many applications including, item tracking, inventory control, point of sale operations, patient care, and so on.

IBM's Advanced Function Presentation (AFP) data stream defines an architecture for presenting bar codes. Bar codes can be printed by using printer files and the BARCODE DDS keyword. They can also be printed by using the Advanced Function Presentation utilities. Bar code support works best on laser printers. Dot matrix printers and some ink jet printers do not support bar codes.

The AFP-to-ASCII transform supports the following industry standard bar code types:

- Code 39, AIM USS-39
- MSI
- UPC/CGPC Version A
- UPC/CGPC Version E
- UPC Two-digit Supplemental
- UPC Five-digit Supplemental
- EAN-8
- EAN-13
- Industrial 2-of-5
- Matrix 2-of-5
- Interleaved 2-of-5
- Interleaved 2-of-5, AIM USS-1 2/5
- Codabar 2-of-7, AIM USS-Codabar
- Code 128, AIM USS-128, UCC/EAN 128
- EAN Two-digit Supplemental
- EAN Five-digit Supplemental
- POSTNET
- PLANET
- Japan Postal
- Royal Mail
- Australian Postal 4-state
- Dutch Postal (KIX)
- Data Matrix
- MaxiCode
- PDF417
- QR Code

### Limitations of AFP-to-ASCII transform function

Currently, the AFP to ASCII transform has the following limitations:

- It does not support graphics (GOCA) commands.
- It does not support multi-up printing.
- It does not provide support for computer output reduction (COR) printing.
- It ignores the fidelity attribute of the spooled file and always performs content printing.
- It can only produce 240 dpi images and 300 dpi images.
- It does not produce color bar codes or images (IM or IOCA).

## Host print transform in raster mode

In raster mode, the host print transform builds a raster image for each page of the output. The raster image is then compressed. AFP fonts must be installed on i5/OS before using the raster mode. Both the SCS-to-ASCII and AFP-to-ASCII transforms support raster mode.

### Reasons to use raster mode

Raster mode preserves the relative position of text, images, barcodes, and overlays if the output is positioned within the no print border of your ASCII printer. Use raster mode to:

- Transform SCS or AFP output to formats other than printer data streams. The host print transform function can, for example, transform SCS or AFP output to TIFF (Tagged Image File Format) for viewing or archiving.
- Support SCS or AFP functions not available on your ASCII printer. For example, the IBM 4019 printer does not support multiple print orientations on the same page.
- Support the printing of AFP output spooled with DBCS or Unicode encoded data.
- Support printing of SCS output spooled with DBCS data on PCL printers

### Enable raster mode

To enable raster mode:

- Install IBM AFP Compatibility Fonts (option 8 of IBM i5/OS).
- For better font support, install IBM AFP Font Collection for IBM Operating System (5648-B45).
- For SCS or AFP output containing double byte characters (DBCS) install the Advanced Function Presentation DBCS Fonts for i5/OS (5769-FN1). You can also install one of the DBCS outline font features for the IBM AFP font collection. For AFP output containing Unicode encoded data, install the AFP Unicode migration fonts or Additional Fonts (Option 43 of i5/OS). For SCS output, you must also specify the i5/OS system value QIGCCDEFNT (double byte coded font).
- Configure the host print transform to use a workstation customization object with the following tag:  
:RASTERMODE  
SCS=YES  
AFP=YES.

Raster mode does not support all printer data streams. Refer to Workstation Customization Programming PDF for detailed information about raster mode support.

### Limitations of raster mode

The following are limitations of raster mode:

- Raster mode requires more i5/OS resources and is slower than the host print transform mapping mode.
- Raster mode does not support dot matrix printers.
- The host print transform can only produce raster images with 240 or 300 dots per inch (dpi). It does not support ink jet printers that have a resolution of 360 or 720 dpi.
- Raster mode does not support hardware justification of text.
- For SCS output that contains double-byte characters, you can specify only one double-byte coded font. Select this font by setting the QIGCCDEFNT system value.

### Related information:



Workstation Customization Programming PDF

## Host print transform and emulator products

Some emulator products provide printer emulation for ASCII printers on the IBM i operating system. The host print transform does not eliminate the need for emulators. Emulators are still necessary to attach the personal printer to the System i product.

### Related concepts:

“Configuring host print transform” on page 141

The host print transform is enabled when you specify \*YES for the TRANSFORM parameter in the printer device description.

## Image print transform

The image print transform is an i5/OS function that is capable of converting image or PostScript data streams into various ASCII and non-ASCII printer data streams.

The conversion takes place on the System i platform, which means the data stream generated is independent of any printer emulators or hardware connections.

### Image print transform overview

The image print transform function converts image or print data from one format into another. A supporting printer interprets the resultant printer data stream.

The image print transform function can convert the following data streams:

- Tag Image File Format (TIFF)
- Graphics Interchange Format (GIF)
- Windows Bitmap (BMP)
- PostScript Level 1 with DBCS extensions

The image print transform function can generate the following printer data streams:

- Advanced Function Presentation Data Stream
- Hewlett-Packard Printer Control Language (PCL)
- PostScript Level 1

The image print transform function converts the data on the System i platform instead of on an emulator.

When a data stream is converted by the image print transform function, the printer data stream that is created contains a bit-mapped image. A *bit-mapped image* is an array of numeric values. Each value represents part or all of a *pixel*. A *pixel* is a single point or dot of an image. An image is typically measured in terms of pixels, for both width and height. The *resolution* of an image is then defined as the number of pixels (dots) per unit of measure. For example, a resolution supported by many printers is 300 dots per inch (dpi). Therefore, an image having dimensions 1200 pixels by 1500 pixels can have a width of 4 inches and a height of 5 inches when printed at 300 dpi.

The image print transform function works with ASCII printers as well as Intelligent Printer Data Stream (IPDS) printers that have AFP(\*YES) specified in the configuration. When the image print transform function is used, the transform does not take place until after the data stream is spooled. Then, when the spooled file is printed or sent to a remote output queue, it is first sent to the image print transform function to be transformed. Printing with the image print transform function is done automatically when:

- A printer device is created with the host print transform function enabled.
- The device or remote output queue has an image configuration object defined.

### Related concepts:

“Configuring host print transform” on page 141

The host print transform is enabled when you specify \*YES for the TRANSFORM parameter in the printer device description.



## Image print transform advantages

There are many advantages for using the image print transform function.

- Support for Intelligent Printer Data Stream (IPDS) printers

Tag Image File Format (TIFF), Graphics Interchange Format (GIF), and Windows Bitmap (BMP) image files, as well as PostScript Level 1 files, can be converted to Advanced Function Presentation Data Stream format and printed on IPDS printers configured AFP(\*YES).

- Support for ASCII printers

TIFF, GIF, and BMP image files, as well as PostScript Level 1 files, can be converted to PCL-5 and PostScript Level 1 format and printed on ASCII printers supporting these languages.

**Note:** You cannot convert PostScript to PostScript using the image print transform function. When the input and output data streams are PostScript, the data is sent directly to the output destination without conversion.

- Customized printer support

Image configuration objects are used with the image print transform function to specify certain characteristics of the converted data streams. When associated with the device description information for a printer that is connected to a System i product, an image configuration object acts as a template for the converted data stream. Attributes, such as data stream format, color, and resolution, are all specified in the image configuration object.

- Additional capabilities

In addition to converting data from one format to another, other functions can be performed by the image print transform function. Among these are the ability to reduce color, compress data, and change photometricity. For more information about the features of the image print transform function, see the Print APIs topic.

**Note:** You cannot perform functions that your printer does not support. For example, you cannot print in landscape orientation when your printer only supports portrait orientation.

### Related information:

Print APIs

## Image configuration objects

An image configuration object contains various printer characteristics that the image print transform function and the Convert Image API use when creating output.

An image configuration object is a list of characteristics that are supported by the printer it represents, acting as a template that guides the transform process. Each image configuration object has values for the following attributes:

- Data stream format
- Photometric interpretation
- Resolution units
- Horizontal resolution
- Vertical resolution
- Compression type
- Bits per sample
- No-print borders (left, right, top, bottom)

All of these fields can be overridden by using the Convert Image API and specifying a value for the field of the same name.

### Output attributes

The following output attributes are derived from the image configuration object:

- Data stream format
- Photometric interpretation
- Resolution units
- Horizontal resolution
- Vertical resolution
- Compression type
- Bits per sample
- No print borders (left, right, top, bottom)

**Note:** If any of the above attributes are specified in the user-defined data attribute of the spooled file with the Convert Image API, then only those attributes that are not specified will be derived from the image configuration object.

The output paper size attribute is derived from the printer device description if the output data stream format is AFP data stream and the printer is an IPDS printer that has AFP(\*YES) specified in the configuration.

The Paper Size attribute is derived from the printer file (spooled file attributes) if the output data stream format is not AFP data stream and the printer is not an IPDS printer that has AFP(\*YES) specified in the configuration.

**Related information:**

Convert Image (QIMGCVTI, QimgCvtImg) API

## Convert Image API

The Convert Image API provides the same transform capabilities as the image print transform function.

In addition, printing with the Convert Image API gives the user more control over how the output looks than the image print transform function does. It gives the user the ability to immediately transform a data stream when delaying the transform is not desired. It also has more options regarding the type of input object and output object. The Convert Image API supports input from and output to an integrated file system file, a spooled file, or main storage. The Convert Image API can also generate a spooled file, which can then be transformed with the image print transform function. When this is done, the Convert Image API stores all the values needed to do the transform in the user-defined data attribute of the spooled file for later use by the image print transform function. For more information about how to use the Convert Image API, see the Convert Image (QIMGCVTI, QimgCvtImg) API in the Programming topic.

**Related information:**

Convert Image (QIMGCVTI, QimgCvtImg) API

## Conversion of PostScript data streams with image print transform

Converting PostScript data streams is performed differently from converting image data streams. PostScript conversion might require font files in order to rasterize the data.

PostScript data streams are converted by using a separate job named QIMGSERV. The QIMGSERV job runs in the QSYSWRK subsystem, and uses the QIMGJOB job description with a priority of 5. The QIMGSERV job is started by either the writer job or the job that calls the Convert Image API. The QIMGSERV job and the job that starts the QIMGSERV job communicate with each other so that control is automatic.

## Fonts

In order to convert PostScript files effectively, PostScript fonts are required to convert text and symbols into bit-mapped images. Below are lists of fonts supplied by IBM for use with the image print transform function. Each set of fonts is located in the integrated file system in the specified directory. For each font name, there is a corresponding font file containing rasterization information. This mapping information is stored in the psfonts.map file.

**Note:** Do not alter the IBM-supplied font files or the psfonts.map file included in the i5/OS operating system. Changing a font file or font mapping can cause unpredictable results.

The following table shows the Latin Fonts that are stored in the /QIBM/ProdData/OS400/Fonts/PSFonts/Latin directory.

Font	Associated file
TimesNewRoman	TNR.PFB
TimesNewRoman-Bold	TNRB.PFB
TimesNewRoman-BoldItalic	TNRBI.PFB
TimesNewRoman-Italic	TNRI.PFB
Helvetica	HEL.PFB
Helvetica-Bold	HELB.PFB
Helvetica-BoldItalic	HELBI.PFB
Helvetica-Italic	HELL.PFB
Courier	COU.PFB
Courier-Bold	COUB.PFB
Courier-BoldItalic	COUBI.PFB
Courier-Italic	COUI.PFB

The following table shows the symbol fonts that are stored in the /QIBM/ProdData/OS400/Fonts/PSFonts/Symbols directory.

Font	Associated file
CourierSymbols	COU_S.PFB
CourierSymbols-Bold	COU_SB.PFB
HelveticaSymbols	HEL_S.PFB
HelveticaSymbols-Bold	HEL_SB.PFB
TimesNewRomanSymbols	TNR_S.PFB
TimesNewRomanSymbols-Bold	TNR_SB.PFB

### User-supplied fonts

To enhance the capabilities of the image print transform function when converting PostScript data streams, you can add your own font files to be used in conjunction with the IBM-supplied fonts included in i5/OS. These fonts are called user-supplied fonts.

The user-supplied font mapping file, psfonts.map, is stored in the directory as /QIBM/UserData/OS400/Fonts. It behaves the same way as the psfonts.map file that is included in i5/OS. An important difference is that the image print transform function searches for fonts by looking first at the user-supplied font mapping file, and then at the i5/OS font mapping file.

To add a user-supplied font, follow these steps:

1. Use an ASCII text editor to open the `psfonts.map` file located in `/QIBM/UserData/OS400/Fonts`. If this file does not exist, you need to create it.
2. Add a new line to the file to include the new font name and associated path and file name. For example:

```
font MyNewFont    /QIBM/UserData/OS400/Fonts/PSFonts/MNF.PFB
```

where `MyNewFont` is the name of the font, and `MNF.PFB` is the associated font file.

3. Save the new `psfonts.map` file.
4. Copy the font file into the directory specified in the line added to the `psfonts.map` file.

To delete a user-supplied font, simply remove the line that maps the font name to its associated file in the `psfonts.map` file, and remove the font file from `i5/OS`.

### Font substitutions

When a font requested within a PostScript data stream is not available on `i5/OS`, a font substitution can be defined if there is a similar font available. A font substitution is the mapping of a font name to a font that is available and very similar (in terms of its rasterization properties) to the font file being replaced. You can also specify a font substitution if an existing font mapping is producing undesirable output.

The following table shows the font substitution mappings that are defined for Latin fonts.

Font	Associated file
Courier-BoldOblique	COUBI.PFB
Courier-Oblique	COUI.PFB
Helvetica-BoldOblique	HELBI.PFB
Helvetica-Oblique	HELL.PFB
Times-Bold	TNRB.PFB
Times-BoldItalic	TNRBI.PFB
Times-Italic	TNRI.PFB
Times-Roman	TNR.PFB

The following font symbol substitution is also defined.

Font	Associated file
Symbol	TNR_S.PFB

To define a font substitution, follow these steps:

1. Use an ASCII text editor to open the `psfonts.map` file that is located in `QIBM/UserData/OS400/Fonts`

If this file does not exist, you need to create it.

2. Add a new line to the file to include the font name and the path and file name of the font file that you want to use as a substitute. For example:

```
font Courier    /QIBM/UserData/OS400/Fonts/PSFonts/HEL.PFB
```

3. Save the new `psfonts.map` file.

### Page size

Page size is determined by the output data stream format and the printer device type. Depending on these values, the page size might be derived from various places.

If the page size is not specified with the Convert Image API, you can use the following guidelines to determine from where the page size is derived:

- If the output data stream format is not AFP data stream and the printer device is not an Intelligent Printer Data Stream (IPDS) printer configured as AFP(\*YES), the paper size from paper source 1 of the device description is used.
- If the output data stream format is not AFP data stream and the printer device is an IPDS printer configured as AFP(\*YES), a default value of letter is used.
- If the output data stream format is AFP data stream, the page size is calculated from the values in the specified printer file.

## Additional printing considerations

You also need to consider direct print jobs, print fonts, and output fields.

### Direct print considerations

When output data is written directly to the printer instead of the output queue, the job is called a direct print job. When you print a direct print job on an SCS printer, the IBM i operating system changes the file device type to \*SCS. When you print on an IPDS printer that is configured with AFP(\*NO), the operating system changes the file device type to \*IPDS. If the print job has the file device type \*AFPDS, \*USERASCII, \*LINE, or \*AFPDSLIN, the direct print job is not supported.

#### Related concepts:

“Allowing direct printing” on page 229

Direct printing is when the \*SPOOL parameter in the printer file is set to \*NO and output data is written directly to the printer.

### Open considerations

These considerations apply to opening printer files for spooled output.

- The output queue should be created for the type of output that your program produces so that system operator intervention can be kept to a minimum while a printer writer is producing output. Consider the following items when creating an output queue:
  - What form of output is being produced (printer or diskette).
  - What kind of forms the output is printed on.
  - What kind of protection you want to place on your data. (Do you want someone else to be able to display your data?)
  - How many job separators you want.
- The SCHEDULE parameter specifies when output is to be made available to a printer writer. When a printer writer processes a specific file, the printing process is dependent on things such as these:
  - When the writer is started
  - Other output files on the queue
  - If the writer or the output queue is being held
- The parameters specified to produce output are saved until they are used by the writer.

### Output considerations

These considerations apply to output operations performed on spooled files.

The force-end-of-data (FEOD) operation can make part of the spooled file available to the writer unless SCHEDULE(\*JOBEND) or HOLD(\*YES) are specified for the file. This operation allows you to write parts of a spooled file; for example, you can write one order at a time. You should not use the force-end-of-data operation for normal output. A new spooled file is started after each FEOD operation.

### Close considerations

When the schedule value is \*FILEEND, the output file is made available to the printer writer. The file resources used by the program are deallocated.

If an application program is writing data when the system ends abnormally, the spooled output is shown to contain 0 pages on spool displays such as the WRKOUTQ, WRKSPLFA, and WRKJOB displays. Records that were stored in internal system buffers are lost.

Spooled files that contain no records (open and close, but no output) are automatically deleted by the system when the application closes the device file. Writers cannot select these files for printing unless SCHEDULE(\*IMMED) is specified and the writer selects the file before it is closed.

### First-character forms-control data considerations

For program-described printer files, you can also specify the print control information in the data by including an American National Standard first-character forms-control code in position 1 of each data record in the printer file. (You cannot use first-character forms control and data description specifications on the same file.)

To include the print control information in the data, you specify one of the following American National Standard first-character forms-control codes in the first position of each data record:

Control code	Action before printing a line
' '	Space one line (blank code)
0	Space two lines
-	Space three lines
+	Suppress space
1	Skip to channel 12
2	Skip to channel 12
3	Skip to channel 12
4	Skip to channel 12
5	Skip to channel 12
6	Skip to channel 12
7	Skip to channel 12
8	Skip to channel 12
9	Skip to channel 12
A	Skip to channel 12
B	Skip to channel 12
C	Skip to channel 12

By default, any other character in position 1 of a record is a blank (the American National Standard code for spacing one line). If this occurs, the notify message CPF4916 is sent to the high-level language program once per file.

When you use first-character forms-control data for a printer file, the print control information created by the high-level language compiler is ignored. The character in position 1 of the record is used as the print control character for that record.

To create a program-described printer file that uses first-character forms-control data, specify the CTLCHAR parameter and, optionally, the CHLVAL parameter on the Create Printer File (CRTPRTF) command. CTLCHAR(\*FCFC) specifies that the first character in every record is an American National Standard forms-control code.

The CHLVAL parameter allows you to associate a specific skip-to line number with an American National Standard channel identifier. For example, if you specify CHLVAL(2 20), channel identifier 2 is allocated with line number 20; therefore, if you place the forms-control 2 in the first position of a record, the printer skips to line 20 before printing the line.

**Note:** If the printer stops at a particular line number and the line number is the same as the form-control number of the channel value parameter of the next record to be processed, the printer advances to that value (line number) on the next page. In the preceding example, if the printer is on line 20, the printer advances to line 20 of the next page.

There is one exception to this method of printer advancement.

If the printer is positioned at the top of the page (line 1) and the channel value forms-control value is line 1, the printer does not advance to a new page.

Each control identifier can be specified only once on the parameter. If no line number is defined for a channel identifier and that channel identifier is encountered in the data, the printer takes the default of spacing one line before printing.

In the following example, a file, PRTFCFC, that uses first-character forms-control data is created:

```
CRTPRTF FILE(QGPL/PRTFCFC) OUTQ(PRINT) CTLCHAR(*FCFC) +  
CHLVAL((1 1) (2 10) (12 60))
```

The printer output is spooled to the output queue PRINT. Channel identifier 1 is associated with line 1, channel identifier 2 is associated with line 10, and channel identifier 12 is associated with line 60.

#### **Related information:**

Machine code carriage control characters

### **Printer font considerations**

You might want to take these into consideration when deciding which font to use for printing.

#### **Characters per inch versus font**

Most SCS printers use the CPI parameter on the Create Printer File (CRTPRTF), Change Printer File (CHGPRTF), or Override with Printer File (OVRPRTF) command to determine the pitch (characters per inch) of the printed output. However, the 3812, 3816, 5219, and all IPDS printers use the FONT parameter to select both an implied pitch (characters per inch) and a font style. For these printers, the CPI parameter is ignored except when the page size (PAGESIZE) is converted from \*UOM to \*ROWCOL. Conversely, the FONT parameter is ignored on printers where it is not applicable.

To provide flexibility in your print job, the pitch of the font identifier (FONT parameter) should match, where possible, the value specified on the CPI parameter. By setting the FONT parameter to FONT(\*CPI), the system selects a font of the same pitch as the CPI parameter value. A print job intended for a printer that supports fonts can then be printed on another printer without significant changes in the appearance of the printed output. For example, a printer file that has FONT(222), Gothic font with 15 pitch, and CPI(15) can print on a 3812, 4224, or 5219 Printer (that uses the FONT parameter), or on a 4214, 4230, 4234, 4247, 5224, or 5225 Printer (that supports 15 characters per inch). If this print job is directed to a printer that supports only 10 characters per inch, printer file redirection is used. For more information about redirecting spooled files to SCS printers, see Redirecting output considerations.

When you use an SCS externally described printer file, the value specified in the CPI parameter is normally used to position fields on the printed page. For example, if a printer file has 10 characters per inch specified and FIELDA is specified to start in column 51, then there are 50 blanks to the left of FIELDA (50 blanks at 10 characters per inch is 5 inches).







- The data stream created might be longer than if a specific font were selected. This means the spooled file might take more storage in the output queue. Fields are positioned with spaces (hex 40) between them. Commands are not used to specify the location where a field is to be placed.
- If a proportionally spaced font is used with a field in the file, any following fields might not be positioned in the column used if a specific font were specified at the file level. This is caused by the variable width of the characters in the proportionally spaced font, which are followed by spaces (hex 40) to position the next field.
- If the value of the FONT parameter on the printer device description is \*DEV D or 0, font 011 is selected.

The maximum number of fonts that can be sent to a printer file is 48. When more than 48 fonts are requested, an error message is sent.

A slight adjustment is sometimes made to the first or last line of a page when the lines per inch (LPI) parameter value is greater than 6. This adjustment prevents IPDS printers from reporting position check errors due to part of a character printing off the top or bottom of the page. For the first line on a page, a slight downward adjustment is made. For the last line on a page, a slight upward adjustment is made. This adjustment is about 1/72 of an inch. No other lines on the page are adjusted. This adjustment is made only for spooled files with DEVTYPE of \*SCS or \*IPDS when printed on IPDS-capable printers. It is suggested that the first line on a page not be used for printing if the lines per inch (LPI) parameter on the printer file is 8 or greater.

**Note:** If an optical character recognition (OCR) font is specified with a non-OCR code page, the code page is changed to an OCR code page. If a non-OCR font is specified with an OCR code page, the font is changed to an OCR font.

**Related concepts:**

“Redirecting output considerations” on page 110

Spooled or nonspooled output intended for a printer can be redirected to another printer. However, each file is checked to ensure that the file attributes (device type, number of lines per inch, number of characters per inch, page length, and page width) and any advanced functions used by the file (such as variable LPI, variable font, or defined characters) are valid on the new printer.

**Related reference:**

“AFP compatibility fonts and font substitution” on page 306

This section contains information about Advanced Function Presentation (AFP) compatibility fonts, character identifiers and other printing characteristics as well as several tables to help you understand font substitution.

**Alternative character sets and code pages for printer output considerations**

Character sets are used with code pages to determine how each character shows in the printed output.

Code pages consist of hexadecimal identifiers (code points) assigned to character identifiers. For example, in code page 037 (EBCDIC), the letter e is assigned a code point of hex 85.

In multinational environments, data in one national graphics character set might need to be printed on devices that support another national character set. This is particularly true of characters with accents and other characters with diacritical marks (such as ç, ñ, and ü). In this topic, these characters are called *extended alphabets*.

For example, assume that a physical file on the system contains data in the Basic French character set and includes the character é. In the code page used with the Basic French character set, this character is hex C0. The data can be entered on a display device that can handle the character or can be sent to the system from another system over a communications line. When hex C0 is sent to a printer that is set up for the United States Basic character set, the hex C0 is printed as {. Depending on the printer and the hexadecimal value sent, the hexadecimal value can be an unprintable character. The way the printer

handles a specific hex code point (for example, hex C0) depends on the current value of the CHRID parameter in the printer file. You can specify the following parameter values for the CHRID parameter:

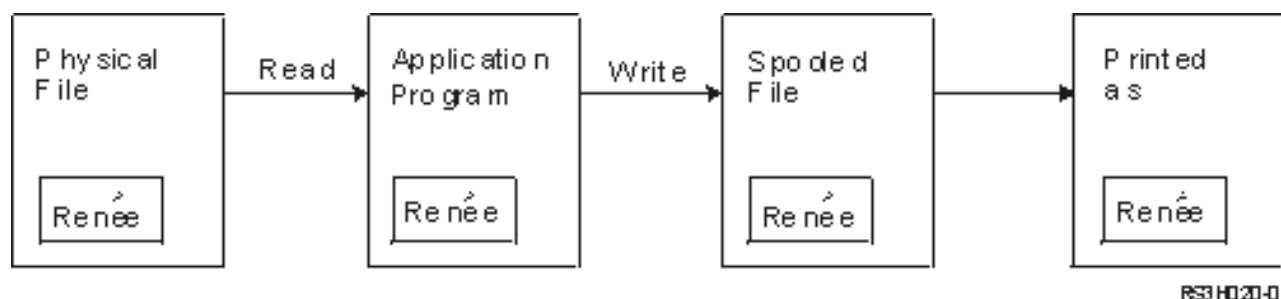
- With an explicit value specified for the CHRID parameter, the printer interprets the data as if the data were in the character set and code page specified.
- With CHRID(\*SYSVAL) specified, the printer file takes the value specified in the Graphic character set/Code page (QCHRID) system value when the output is created.
- With CHRID(\*DEVD) specified, the printer uses the CHRID that was set with the device control panel or that was specified when the printer device description was created.
- If you have specified CHRID(\*JOBCCSID), the printer interprets the data as if it were in the character set and code page that are associated with the CCSID for the current job. For more information, see the Globalization topic.
- With CHRID(\*CHRIDCTL) specified, the printer file checks the CHRIDCTL job definition attribute to determine whether to use \*JOBCCSID or \*DEVD on the CHRID command parameter for the job.

Not all printers can handle all CHRID parameter values. If a CHRID is specified for a printer on which that CHRID is not supported, a message is sent to the operator. See CHRID values supported topic for a description of which printers support which extensions.

For program-described printer files, the value of the CHRID parameter determines the code page and character set used to print the data. However, for externally described printer files, the CHRID parameter is used only for fields that also have the CHRID DDS keyword specified. Fields that do not have the CHRID DDS keyword use the code page and character as if CHRID(\*DEVD) had been specified for the CHRID parameter on the printer file.

For printers with AFP(\*YES) in the device description, the printer file CDEPAG and FNTCHRSET parameters can be used to select a code page when the printer file DEVTYPE parameter is \*AFPDS. This applies to all IPDS printers except 4224, 4230, 4234, 4247, and 64xx.

The following figure shows how extended alphabets are handled in printer output.



Assume that a record in a physical file contains a field with the value Renée. An application program reads the record from the physical file and writes a record that contains the data to the spooled file. The output field in the printer file that describes how Renée is to be printed has the CHRID DDS keyword specified, indicating that the printer is to interpret extended alphabets. The graphic character set 288 and code page 297 are specified for the interpretation in either the printer file or the Graphic character set/Code page (QCHRID) system value. Code page 297 is used for French language.

When printing the data, the printer interprets hex C0 as specified in character set 288 and code page 297. If character set 101 and code page 037 are selected, hex C0 (é) is printed as {.

One of the following CHRID values (graphic character set and code page) must be specified to print fonts OCR-A and OCR-B on the IPDS printers:

- 580 340
- 590 340

- 697 892
- 697 893

**Related reference:**

“CHRID values supported” on page 379

The table in this topic lists all the character identifiers (CHRID), the related national language groups, the correct code page, and which printers support which character identifier.

**Related information:**

i5/OS globalization

## **Output field considerations**

The system provides editing support that makes fields more readable when they are printed.

With the system editing support, you can do the following tasks:

- Suppress leading zeros.
- Punctuate a field with commas and periods to show decimal column and to group digits in sets of three.
- Print negative values with a minus sign or CR to the right.
- Print zero values as zeros or blanks.
- Print asterisks to the left of significant digits to provide asterisk protection.
- Print a currency symbol corresponding to the system value Currency symbol (QCURSYM).

The system provides this editing support with edit codes and edit words. Edit codes are a defined set of editing patterns. You identify these by name, and the system edits a field according to the pattern defined by the named edit code. Edit words are edit patterns that you define to produce the results that you want. Edit codes include most commonly used editing requirements. You need to use the edit word support only for those editing needs not included by edit codes.

There are two methods of using edit codes and edit words. Which one you use depends on how you define the printer file and how it is used in an application program. If your application is using program-described data, your high-level language might allow you to identify edit codes or create your own edit words. If your application is using externally described data, the edit code (EDTCDE) DDS keyword allows you to identify an edit code; the edit word (EDTWRD) DDS keyword allows you to define your own editing pattern.

The system provides several edit codes:

- 1 through 4
- A through D
- J through M
- X through Z

The editing patterns defined by these codes are described in Programming: DDS topic collection.

### **User-defined edit codes**

You can also define five edit codes to provide more editing functions than is available with the i5/OS edit codes, and to handle common editing functions that otherwise require the use of an edit word. These are called user-defined edit codes. For example, you might need to edit numbers that include hyphens (like some telephone numbers), or more than one decimal point. You can use user-defined edit codes for these functions. These edit codes are named QEDIT5, QEDIT6, QEDIT7, QEDIT8, and QEDIT9 and can be referred to in data description specifications or in a high-level language program by number (5, 6, 7, 8, or 9).

These edit codes are created with the Create Edit Description (CRTEDTD) command. Edit descriptions are always placed in library QSYS. They cannot be moved or renamed; only one occurrence of each is allowed. Edit descriptions have an object type of \*EDTD.

IBM supplies a version of each of the QEDIT edit codes. You can use these edit descriptions as they are, or you can delete them and create your own. See Programming: DDS topic collection for more information about using these edit descriptions.

Before using any of the user-defined edit codes, you need to check its contents on your system because it might have been changed from the IBM-supplied version. The Display Edit Description (DSPEDTD) command can be used to display the contents of a user-defined edit code.

Changing a user-defined edit code description does not affect any application or printer file that has already been created with that edit description. If you want your application to use the changed edit description, you must either create the high-level language program again (if the edit code is used in the program) or create the file again (if the application is using an externally described file that contains EDTCDE keywords).

**Related information:**

Programming: DDS

**Externally described printer-file record-format considerations**

When a program using externally described printer files is compiled, the compiler extracts the file descriptions for the files referred to in the program and makes these file descriptions part of the compiled program.

When you run the program, you can verify that the record formats with which the program was compiled are the current record formats. To do this, you use the LVLCHK parameter on a create file command when the file is created.

The system assigns a unique level identifier for each record format when the file it is associated with is created. The system uses the information in the record format description to determine the level identifier. This information includes the name of the record format, the names, attributes, and order of the fields in the format, the indicators used, and the names and the order of the indicators in the record format. If you use the INDARA keyword to remove the indicator from the output buffer, the indicators used are not included in the level identifier information.

When the file is opened, if level checking (the LVLCHK parameter) is specified, the system does a format-by-format comparison of the level-checking values specified in the program to the level-checking values specified in the printer file. If any of the formats specified in the program do not exist in the file, or if any of the level checking values are different, an error occurs. Formats can be added to or removed from a printer file without affecting existing application programs that do not use the added or deleted formats.

You should display the file description to determine if the changes affect your program. You can use the Display File Field Description (DSPFFD) command to display the file description or, if you have the source entry utility (SEU), you can display the source file. Not every change in a file necessarily affects your program. You might not have to recompile your program. If you do not have to recompile your program, you should specify LVLCHK(\*NO) for the file by using the Change Printer File (CHGPRTF) or Override Printer File (OVRPRTF) command.

You can add a field to the end of a printer file record format without having to recompile your program as long as you do not want to use the field in your program. If you delete a field from the end of the record format, you do not have to recompile your program if you are not using the field. However, if you add a field to or delete a field from a record format anywhere other than at the end, you must recompile your program. Otherwise, the field offsets in the record passed to and from the program are wrong for processing.

In general, anything that changes the length or position of any fields in the record format used by the program requires that the program be recompiled.

## Redirecting output considerations

Spooled or nonspooled output intended for a printer can be redirected to another printer. However, each file is checked to ensure that the file attributes (device type, number of lines per inch, number of characters per inch, page length, and page width) and any advanced functions used by the file (such as variable LPI, variable font, or defined characters) are valid on the new printer.

### Nonspooled output

When a nonspooled file is redirected and the printer file attributes do not match the new printer, one of the following events occurs:

- If the printer file specifies a characters-per-inch value that is not supported by the device being used, a diagnostic message (CPF4057) is sent to the program message queue and the data is printed at 10 characters per inch. If the page width is greater than 132 characters, the records are folded.

**Note:** Folding is not supported on IPDS printers.

- If the printer file specifies a lines-per-inch value that is not supported by the device being used, a diagnostic message (CPF4056) is sent to the program message queue, and the data is printed at 8 lines per inch.
- If the page length is greater than the maximum length allowed for the printer being used, the printing ends with an escape message (CPF4138).
- If the printer file specifies special device requirements (such as use of certain DDS keywords) that are not supported by the device being used, then a diagnostic message is sent to the program message queue and the special function is ignored.

### Spooled files

When a spooled file is redirected to another printer, the spooled file cannot be printed without changes if any of the spooled file attributes are not supported by the printer device. Here are some printer device attributes that might not be supported are:

- Page size
- Output drawers
- Print quality
- Lines per inch
- Characters per inch

### Spooled file redirected to SCS printers

The following actions are taken when a spooled file is redirected to an SCS printer and cannot be printed without changes (SCS printers include the 3812, 3816, 4214, 4234, 4245, 4247, 5219, 5224, 5225, 5256, 5262, 6252, and 6262 Printers):

- An inquiry message is sent to the message queue of the writer if one of the following conditions is true:
  - The spooled file uses the IPDS data stream (DEVTYPE(\*IPDS)).
  - The spooled file attributes are not supported by the printer.
  - The special device requirements used by the spooled file are not supported by the printer.

The inquiry message allows these options:

- End the writer.
- Print the spooled file with lines folded when the lines are longer than the width of IBM-supplied printer file QPSPLPRT.

- Print the spooled file with lines truncated when the lines are longer than the width of IBM-supplied printer file QPSPLPRT.
- Hold the spooled file and process the next file on the output queue.

If the spooled file is printed, results might be unpredictable because the file is printed with the printer attributes specified in the IBM-supplied printer file QPSPLPRT, and all advanced functions used by the spooled file are removed. Functions removed include these:

**DDS keywords:**

**CHRID**

Graphic character set and code page

**CHRSIZ**

Character size (width and height)

**CPI** Characters per inch

**DFNCHR**

Define character

**DRAWER**

Paper drawer selection

**FONT** Font selection

**LPI** Lines per inch

**PAGRTT**

Page rotation

**PRTQLTY**

Print quality

**TRNSPY**

Transparency

**Other print functions:**

- Drawer change in document
- Font change in document
- Lines-per-inch change in document
- Page rotation in document
- Subscript and superscript
- If the spooled file specifies a characters-per-inch value that is not supported by the printer, an inquiry message is sent to the message queue of the writer with the following options:
  - End the writer.
  - Print the spooled file at 10 characters per inch with lines folded when the lines are longer than the width of IBM-supplied printer file QPSPLPRT.
  - Hold the spooled file and process the next file on the output queue.
- The 5219 Printer is an exception to the above cases if the only mismatch between the spooled file and printer is the HIGHLIGHT special device requirement that the file contains. When this occurs, an inquiry message is sent to the message queue of the writer with the following options:
  - End the writer.
  - Print the spooled file without highlighting but keep all other advanced functions used by the file.
  - Attempt to print the spooled file without changing. (If this is not successful, the file will be held on the output queue.)
  - Hold the spooled file and process the next file on the output queue.

If the spooled file is printed, the resulting output closely resembles how the file was intended to look. This is because the attributes specified by the spooled file were used and advanced functions were kept.

- Documents created on other systems might contain print controls that are not supported by the 5219 or 3812 Printers. These controls might include variable form size, output drawer, print quality, lines per inch, characters per inch, character identifier, or justification. If this occurs, an inquiry message is sent to the message queue of the writer with the options to:
  - End the writer.
  - Print the spooled file with unsupported values changed to values which are supported by the printer.
  - Attempt to print the spooled file without changing. (If this is not successful, the file will be held on the output queue).
  - Hold the spooled file and process the next file on the output queue.

If the spooled file is printed, the file attributes from the spooled file are used and all advanced functions that are valid for the device are kept. The resulting output should closely resemble what the file was intended to look like, though it still might not print exactly as intended because of the unsupported values.

### **Spooled files redirected to IPDS printers**

IPDS printers include 3130, 3160, 3812, 3816, 3820, 3825, 3827, 3829, 3831, 3835, 3900, 3912, 3916, 3930, 3935, 4028, 4224, 4230, 4234, 4247, 4312, 4317, and 4324. InfoPrint 20, InfoPrint 32, InfoPrint 3000, and InfoPrint 4000 are also IPDS printers. The following actions are taken when you spool a file to an IPDS printer:

- The message queue will receive an inquiry if the spooled file uses the SCS data stream (DEVTYPE(\*SCS)) and contains DBCS (double-byte character set) data. It will also receive an inquiry if it has a page length greater than that supported by the printer (for both SNA character string (SCS) and IPDS files). You can choose any of the following options:
  - End the writer.
  - The spooled file will print with lines that are truncated when the lines are longer than the width of IBM-supplied printer file QPSPLPRT.
  - Hold the spooled file and process the next file on the output queue.

Printing results might be unpredictable because the printer will use the printer attributes that are specified in QSPLPR, the IBM-supplied printer file. All advanced functions that are specified in the spooled file will be removed. Functions removed include these:

#### **DDS keywords:**

##### **CHRSIZ**

Character size (width and height)

**CPI** Characters per inch

##### **DFNCHR**

Define character

##### **DRAWER**

Paper drawer selection

##### **PAGRTT**

Page rotation

##### **TRNSPY**

Transparency

#### **Other print functions:**



- Drawer change in document
  - Font change in document
  - Lines-per-inch change in document
  - Page rotation in document
  - Subscript and superscript
- The message queue will receive an inquiry message for the following condition: if the spooled file uses the SCS data stream (DEVTYPE(\*SCS)), does not contain DBCS data, and uses special device requirements. Special device requirements include graphics, defined characters, transparencies, variable font, and enhanced 3812 fonts. It will also receive an inquiry if it uses a proportionally spaced font for the FONT parameter of the file. You can select any of the following options:
    - End the writer.
    - Transform the spooled file to IPDS format and print.
    - Hold the spooled file and process the next file on the output queue.

If the file prints, its existing attributes will be used. Advanced functions remain intact, except graphics, defined characters, justification, and transparencies. The transformation to IPDS format should substantially keep the integrity of the text data. However, the file might still not print exactly as intended. Unsupported font pitches, font spacings, and character identifiers are changed to the closest approximation valid on the printer.
  - The writer automatically transforms the file to an IPDS file and prints it if the following conditions are true: the spooled file uses DEVTYPE(\*SCS), does not contain DBCS data, does not use defined characters, graphics, transparency, variable fonts, or enhanced 3812 fonts. The printer file also can not use a proportionally spaced font for the FONT parameter of the printer file. The spooled file might not print exactly as intended. For unsupported font pitches, font spacings, and character identifiers, the printer writer uses the closest approximation available on the printer.
  - The message queue of the writer receives an inquiry message if the spooled file uses the IPDS data stream (DEVTYPE(\*IPDS)), but uses advanced functions not supported by the printer. Options made available are as follows:
    - End the writer.
    - Print the spooled file, but dropping unsupported advanced functions from the file.
    - Hold the spooled file and process the next file on the output queue.

### **3812 and 3816 SCS printer considerations**

When automatic configuration is run for 3812 and 3816 printers, the printers report as a 5219 printer. The first time the printer is used, the IBM i operating system sends some commands to the printer that allow the system to distinguish between a 5219 printer and a 3812 or 3816 SCS printer.

However, this happens after the open processing has been done for the first output to print. The first output to print can be direct output or a spooled file. This means, for the first output printed, the system treats the printer as a 5219 printer. For example, this means that there is no page rotation for that first printed output.

In order for the system to recognize a 3812 or 3816 SCS printer, the printer writer must complete processing. After the first output has been printed and a new printer writer is started, the system recognizes the printer as a 3812 or 3816 SCS printer.

After the system recognizes a printer as a 3812 or 3816 SCS, it remembers the true printer type until the printer device description is deleted.

### **3835 printer considerations**

The 3835 Model 1 printer has a no-print border. In this area, about 1/6 inch from all edges of the page, data is not printed.

The IBM i operating system adjusts the positioning of the printed text on the page to compensate for the no-print border. For example, if your application program prints text in the top 1/6 inch or left 1/6 inch, all the text is printed. The system adjusts the start printing position to 1/6 inch from the top and left of the page, causing all printed data on the page to be shifted to the right and down by 1/6 inch. If your application relies on printing data at a certain point on the paper, you might need to change your application to compensate for this adjustment or use the margin values of 0 on the printer file.

**Note:** The 3835 Model 2 does not have a no-print border. The system does not adjust the position of printed text for this printer. When you direct output from printing on a 3835 Model 1 to a Model 2, consider the difference in how the no-print border is used.

### **3912, 3916, and 4028 printer considerations**

The 3912, 3916, and 4028 printers have a no-print border. In this area, about 1/6 inch from all edges of the page, data is not printed.

If the 3912, 3916, or 4028 is configured AFP(\*NO), you might need to adjust your application programs. For example, if your application program prints text in the top 1/6 inch or left 1/6 inch, that text will not appear on the page.

If the 4028 is configured with AFP(\*YES), the IBM i operating system adjusts the positioning of the printed text on the page to compensate for the no-print border. For example, if your application program prints text in the top 1/6 inch or left 1/6 inch, all the text is printed. The system adjusts the start printing position to 1/6 inch from the top and left of the page, causing all printed data on the page to be shifted to the right and down by 1/6 inch. If your application relies on printing data at a certain point on the paper, you might need to change your application to compensate for this adjustment.

To print as close to the edge of the page as possible:

- Use PAGRTT (\*COR)
- Use MULTIUP(2) or (4)
- Configure the printer with AFP(\*YES)

This positions the starting origin of the page to the edge of the printable area.

### **Special printer file considerations for AFP data stream**

If the device type (DEVTYPE) parameter for the printer file is \*AFPDS, certain considerations apply to some printer file parameters and to the sending of spooled files to other systems.

#### **Considerations for printer file parameters**

- Overflow (OVRFLW) parameter  
Overflow is not signaled for externally described printer files (DDS) for record formats that use absolute positioning. In addition, overflow is determined by using the margin offset down value. For example, if the margin offset down value is .5 inches, the overflow line is line 60, and the lines per inch value is 6, overflow is signaled when line 60 is printed on the page. This is 10.5 inches down the page.
- Character Identifier (CHRID) parameter  
The CHRID parameter of the printer file is ignored if a font character set (FNTCHRSET) or coded font (CDEFNT) parameter is specified. One exception to this is if a file contains UCS-2 data that will be converted to EBCDIC data. In this case, the CHRID parameter determines the target conversion CCSID.
- Page Rotation (PAGRTT) parameter  
Overlays, page segments, and resources stored in the integrated file system are not automatically rotated based on the PAGRTT parameter of the printer file.
- Page Size (PAGESIZE) parameter

If the unit of measure is \*ROWCOL, and either a coded font or font character set is specified on the printer file, the page width is calculated using 10 characters-per-inch.

### **Considerations for sending an AFP data stream spooled file to another system**

Because the actual printer device is not known when a spooled file is created, some parameters specified on the printer file cause a default setting to be used in the created spooled file. This is done to provide values in the data stream of the spooled file in the event it is sent to another system.

On the system the spooled file was created on, the correct values (as determined for the printer you want the spooled file printed on) are substituted before the spooled file is printed.

The following parameters cause a default setting to be used:

- If CHRID(\*DEV) or an externally described printer file is used, the Graphic character set/Code page (QCHRID) system value is substituted.
- If FONT(\*DEV) is used, font 11 is substituted.
- If FORMFEED(\*DEV) is used, drawer 1 is substituted.
- If PAGRTT(\*DEV), PAGRTT(\*AUTO), or PAGRTT(\*COR) is used, page rotation of 0 is substituted.

#### **Notes:**

1. On the IBM i operating system, offset stacking of printed output is used. When the job finishes, the paper tray moves, offsetting the stack of paper to make it easier to distinguish between finished jobs. Because of this, the data stream that is created on the operating system contains the control to indicate that offset stacking should be used. If the spooled file is sent to a system that does not support offset stacking, an error message might be issued.
2. Use of the DDS DRAWER and PAGRTT keywords cause the i5/OS to generate an AFP data stream that is not completely supported by the AFP viewer and some of the PSF products available on other IBM platforms. If the spooled file needs to be viewed by the AFP viewer or printed on other IBM platforms, do not use the DRAWER and PAGRTT keywords in DDS. Use the INVMMAP keyword instead to change the drawer or page rotation within the spooled file.

### **Special data description specifications considerations for AFP data stream**

To use Advanced Function Presentation (AFP) support on the IBM i operating system to print on IPDS printers, you must install Print Services Facility (PSF).

If you have additional questions about PSF, contact your IBM representative.

The following list of data description specifications keywords are valid for printer files that have the printer device type (DEVTYPE) parameter value specified as \*AFPDS. Restrictions on DDS keywords are contained in this list as well.

- AFPRSC
- ALIAS
- BARCODE
- BOX
- CCSID
- CDEFNT
- CHRID - Only applies to output printed using a printer resident font. If a coded font (CDEFNT) or a font character set and code page combination (FNTCHRSET) is specified, the CHRID keyword is ignored and a message is issued.
- CHRSIZ
- COLOR - Color is ignored if your printer does not support color printing.
- CVTDTA

- DATE OUTBIN
- DATFMT
- DATSEP
- DFT
- DLTEDT
- DOCIDXTAG
- DRAWER
- DTASTMCMD
- DUPLEX
- EDTCDE
- EDTWORD
- ENDPAGE
- ENDPAGGRP
- FLTFIXDEC
- FLTPCN
- FONT
- FONTNAME
- FORCE
- FNTCHRSET
- GDF
- HIGHLIGHT - Only applies to output printed using a printer resident font. If a coded font (CDEFNT) or a font character set and code page combination (FNTCHRSET) is specified, the HIGHLIGHT keyword is ignored and a message issued.
- IGCCDEFNT
- INDARA
- INDTXT
- INVMMAP
- LINE
- MSGCON
- OVERLAY
- OUTBIN
- PAGNBR
- PAGRTT
- PAGSEG
- POSITION
- PRTQLTY
- REF
- REFFLD
- SKIPA - Not allowed at the file level in a spooled file with printer device type \*AFPDS.
- SKIPB - Not allowed at the file level in a spooled file with printer device type \*AFPDS.)
- STRPAGGRP
- TEXT
- TIME
- TIMFMT
- TIMSEP

- TXTRTT
- UNDERLINE
- UNISCRIP
- ZFOLD

**Related information:**



iSeries Guide to Output PDF

DDS for printer files

## Performance considerations

These are printer file performance considerations.

- For externally described printer files, the fewer the number of fields in a record, the faster the processing of that record. Also, you can reduce the system overhead involved with the processing of each record by putting several lines of text within a record instead of each line as a separate record.
- When you code the data description specifications for externally described printer files, define the fields in sequential order. The output is not changed if fields are not defined in sequential order, but the extra travel time of the printer head might be noticeable.
- For externally described printer files, specify a specific font or FONT(\*CPI) on the CRTPRTF, CHGPRTF, or OVRPRTF command instead of FONT(\*DEV). This helps keep the data stream as small as possible.
- If a spooled file is intended to be printed on an IPDS printer configured AFP(\*NO), specify DEVTYPE(\*IPDS) on the CRTPRTF, CHGPRTF, or OVRPRTF command to avoid the extra system processing required to transform the data stream from SCS to IPDS.
- If a spooled file is intended to be printed on an IPDS printer configured AFP(\*YES), specify DEVTYPE(\*AFPDS) on the CRTPRTF, CHGPRTF, or OVRPRTF command.
- When you print on an IPDS printer configured AFP(\*YES), large spooled files begin to print sooner if the print while convert (PRTCVT) parameter on the printer device description is set to \*YES. However, some printing might occur before the syntax of the entire spooled file is checked. A data stream error might be found after printing has started. This causes printing to end. If you want all data stream syntax checking to complete before the spooled file starts printing, set the PRTCVT parameter of the printer device description to \*NO.

---

## Planning printing

System i products offer a wide range of printing solutions. They range from high-function IPDS printers that support bar-codes and complex document formatting on high performance printers to simple document printing that uses a directly attached printer.


Before you configure a System i printing solution, you need to have a clear understanding of your organization's requirements, the type of printed output you need, and the capabilities of the hardware you currently have or intend to purchase. After you have determined your requirements, you need to decide whether to use IPDS or host print transform to convert and send print data. Generally, IPDS (typically used with PSF) offers higher performance and additional functions, but at a higher price. See the following topics for more information about determining what type of printing solution best meets your business needs.

“Printer attachment methods” on page 54

This topic describes some of the common printer attachment methods.

InfoPrint Solutions Company - Software - i5/OS  ([http://www.infoprintsolutionscompany.com/internet/wwsites.nsf/vwWebPublished/soft\\_i5oshome\\_us](http://www.infoprintsolutionscompany.com/internet/wwsites.nsf/vwWebPublished/soft_i5oshome_us))

Use this Web site to help identify which printing solution best meets your needs.

InfoPrint Solutions Company - Select from a wide variety of InfoPrint printers  ([http://www.infoprintsolutionscompany.com/internet/wwsites.nsf/vwWebPublished/print\\_home\\_us](http://www.infoprintsolutionscompany.com/internet/wwsites.nsf/vwWebPublished/print_home_us))

This Web-based tool can help you select the best printing hardware for your requirements.

Information on Printers from Various Manufacturers 

One of the most common causes of problems in a printing environment is hardware compatibility. See this IBM Software Knowledge Base document (search for the document title at [www.ibm.com/support/](http://www.ibm.com/support/)) to find out which printers support the print protocols you require.

**Related concepts:**


“Printer attachment methods” on page 54

You can use many methods to attach a printer to a System i product, such as a TCP/IP network, a personal computer, a twinaxial workstation controller, and an ASCII workstation controller.


---

## Configuring printing

Configuring System i printing requires that all the hardware components, network configuration settings, and software involved in the connection work together effectively. You can avoid many potential problems with this process by verifying that your printer is fully compatible with the printing protocol that you are using.

The IBM Software Knowledge Base document Information on Printers from Various Manufacturers  (search for the document title at [www.ibm.com/support/](http://www.ibm.com/support/)) contains a list of known protocols supported by some common printers.

See the following for more information about how to configure printing:

IBM - Print Services Facility for i5/OS  The Print Services Facility for i5/OS (PSF) home page provides information about purchasing and configuring IPDS (PSF) printing.

Print shares This topic provides information about sharing output queues with PC clients in the network.

*System i Access for Windows User's Guide* Provides information about configuring a printer that is locally attached to an System i Access client. This information is installed as part of the online help system on the System i Access client.

## Automatically creating printer device descriptions

You can have printer device descriptions created automatically if the automatic configuration (QAUTOCFG) system value is \*YES.

The printer device description is automatically created when:

- A display or printer is powered on.
- The personal computer or emulation programs are started.

## Adding a printer with System i Navigator


You can use the Add Printer wizard in System i Navigator to create a printer device description and its associated objects.


To start the Add Printer wizard, follow these steps.


1. In System i Navigator, expand **My connections** > *your system* > **Basic operations**.
2. Right-click **Printers**.
3. On the menu, select **Add a printer**. The Add Printer wizard then starts. Follow the instructions in the wizard to add your printer.

## Configuring PJI printers

To set up Hewlett Packard print job language (PJI) printers, follow these steps.

1. Ensure that you have met the following requirements:
  - The printer must support Hewlett Packard print job language (PJI) protocol. For a list of printers that are known to support PJI, see the IBM Software Knowledge Base document Information on Printers from Various Manufacturers  (search for the document title at [www.ibm.com/support/](http://www.ibm.com/support/)).
  - If the printer is attached to a network adapter, the network adapter must be compatible with the printer. The network adapter must provide bidirectional communication with the printer, and all PJI commands must pass to and from the printer unmodified.
  - If the printer is attached to an external network adapter, the printer must be connected with an IEEE 1284 bidirectional printer cable and the print server must be in the BITRONICS, BI-DIRECTIONAL or ECP/MLC mode.
2. Use the Create Device Description (Printer) (CRTDEVPRT) CL command to create a printer device description. From a character-based interface, type CRTDEVPRT.
3. On the Create Device Description (Printer) Display, press F9 to display all parameters.
4. Specify the following values.

Parameter	Value
Device description (DEVVD)	The name of your device description; for example, <i>mypjlpri</i> .
Device class (DEVCLS)	This must be set to *LAN.
Device type (TYPE)	This must be set to 3812.
Device model (MODEL)	This must be set to 1.
LAN attachment (LANATTACH)	This must be set to *IP.
Port number (PORT)	The port number that the printer network interface card is expecting for PJI data. Refer to your manufacturer's information or to the IBM Software Knowledge Base document Recommended PJI Port Numbers Values for *LAN 3812 PJI Device Descriptions  (search for the document title at <a href="http://www.ibm.com/support/">www.ibm.com/support/</a> ) for the correct value. Typically this value is 9100 or 2501.
Font (FONT)	This is typically set to 011, which is the default 10 CPI font.
Form feed (FORMFEED)	Set to *AUTOCUT, if single-cut sheets are used by the printer. Set to *CONT, if printing to a continuous forms printer.
Printer error message (PRTERMSG)	Set to *INFO so that the printer's message queue will receive informational messages whenever possible. This can prevent a condition that might cause the printer writer to end.

Parameter	Value
Message queue (MSGQ)	The default value for this parameter is *CTLD, which indicates that the message queue is defined in the attached controller. There is no attached controller for *LAN printer device descriptions, so leaving the Message queue (MSGQ) parameter set to *CTLD will in effect set it to the QSYSOPR message queue in library QSYS. If you want to use another message queue for this printer device, verify that the message queue exists before entering the message queue and library in the MSGQ parameter. Otherwise, the CRTDEVPRT command will fail with message CPF2799, <b>Message queue &amp;1 in library &amp;2 not found.</b>
Host print transform (TRANSFORM)	Set to *YES so that SNA character string (SCS) and AFP data stream spooled files are converted to the printer's data stream.
Manufacturer type and model (MFRTYPMDL)	This value indicates the type of ASCII printer and is used by the host print transform when converting spooled files. For recommendations on the value to use for some popular printers, see the IBM Software Knowledge Base document Information on Printers from Various Manufacturers  (search for the document title at <a href="http://www.ibm.com/support/">www.ibm.com/support/</a> ).
Remote location (RMTLOCNAME)	The name of your printer on the TCP/IP network; for example, <i>192.168.0.1</i> or <i>mynmpprt.myco.com</i> . You should verify that the system can reach the printer by using either the ping or Verify TCP/IP Connection (VFYTCPCNN) command.
System driver program (SYSDRVPGM)	This must be set to *IBMPJLDRV.

All other values can be left as the default value.

5. Press Enter.
6. Make the printer available.
7. Start the printer writer.

#### Related tasks:

“Making a printer available” on page 226

To make the printer available (vary on the printer), follow these steps.

“Starting a printer writer” on page 226


To start the printer writer, follow these steps.

#### Related information:

Create Device Desc (Printer) (CRTDEVPRT) command


## Configuring SNMP printers


To set up Simple Network Management Protocol (SNMP) printers, follow these steps.

1. Ensure that you have met the following requirements:
  - The printer must support the Host Resource Management Information Base (Host Resources MIB, RFC1514). The printer should also support the Printer Management Information Base (Printer MIB, RFC1759). For a list of printers that are known to support SNMP, see the IBM Software Knowledge Base document Information on Printers from Various Manufacturers  (search for the document title at [www.ibm.com/support/](http://www.ibm.com/support/)).
  - If the printer is connected to a network interface card, the network interface card must be compatible with the Host Resource Management Information Base (Host Resources MIB, RFC1514).



- If the printer is connected to an external network adapter that has more than one port, the printer should be connected to the first parallel port and there should be no other SNMP-capable devices attached to the network adapter.
  - The printer and any network adapter must be set to have, at a minimum, a community name of public. The community name of public is typically a factory default setting. Community names are a means of access control for SNMP devices. Read-only access to the public community is sufficient.
2. Use the Create Device Description (Printer) (CRTDEVPRT) CL command to create a printer device description. From a character-based interface, type CRTDEVPRT.
  3. On the Create Device Description (Printer) Display, press F9 to display all parameters.
  4. Specify the following values.

Parameter	Value
Device description (DEVVD)	The name of your device description. For example, <i>mysnmpprt</i> .
Device class (DEVCLS)	This must be set to *LAN.
Device type (TYPE)	This must be set to 3812.
Device model (MODEL)	This must be set to 1.
LAN attachment (LANATTACH)	This must be set to *IP.
Port number (PORT)	The port number that the printer network interface card accepts for SNMP data. Refer to your manufacturer's information or to the IBM Software Knowledge Base document Recommended SNMP Port Numbers Values for *LAN 3812 SNMP Device Descriptions  (search for the document title at <a href="http://www.ibm.com/support/">www.ibm.com/support/</a> ) for the correct value. Typically this value is 9100 or 2501.
Font (FONT)	This is typically set to 011, which is the default 10 CPI font.
Form feed (FORMFEED)	Set to *AUTOCUT, if single-cut sheets are used by the printer. Set to *CONT, if printing to a continuous forms printer.
Printer error message (PRERRMSG)	Set to *INFO so that the printer's message queue will receive informational messages whenever possible. This can prevent a condition that might cause the printer writer to end.
Message queue (MSGQ)	The default value is *CTLD, which indicates that the message queue is defined in the attached controller. There is no attached controller for *LAN printer device descriptions, so leaving the Message queue (MSGQ) parameter set to *CTLD will in effect set it to the QSYSOPR message queue in library QSYS. If you want to use another message queue for this printer device, verify that the message queue exists before entering the message queue and library in the MSGQ parameter. Otherwise, the CRTDEVPRT command will fail with message CPF2799, <b>Message queue &amp;1 in library &amp;2 not found.</b>
Host print transform (TRANSFORM)	Set to *YES so that SNA character string (SCS) and AFP data stream spooled files are converted to the printer's data stream.

Parameter	Value
Manufacturer type and model (MFRTYPMDL)	This value indicates the type of ASCII printer and is used by the host print transform when converting spooled files. For recommendations on the value to use for some popular printers, see the IBM Software Knowledge Base document Information on Printers from Various Manufacturers  (search for the document title at <a href="http://www.ibm.com/support/">www.ibm.com/support/</a> ).
Remote location (RMTLOCNAME)	The name of your printer on the TCP/IP network; for example, <i>192.168.0.1</i> or <i>mynmpprt.myco.com</i> . You should verify that the system can reach the printer by using either the ping or Verify TCP/IP Connection (VFYTCPCNN) command.
User-defined options (USRDFNOPT)	Set to *IBMSHRCNN
System driver program (SYSDRVPGM)	This must be set to *IBMSNMPDRV.

All other values can be left as the default value.

5. Press Enter.
6. Make the printer available.
7. Start the printer writer.

**Related tasks:**

“Making a printer available” on page 226

To make the printer available (vary on the printer), follow these steps.

“Starting a printer writer” on page 226

To start the printer writer, follow these steps.

**Related information:**

Create Device Desc (Printer) (CRTDEVPRT) command



## Configuring LPR/LPD

To configure line printer requester/line printer daemon (LPR/LPD) printing, follow these steps.

**Note:** You should only consider using this method if your printer does not support the Printer Job Language (PCL), Simple Network Management Protocol (SNMP), or Internet Printing Protocol (IPP) printing methods. This method of printing has a number of limitations. Restrictions when using this method include, but are not limited to:


- No page range support.
- Limited or no printer status.
- Limited or no error recovery.
- Limited or no control of a spooled file after it is sent.

1. Start LPD on the print server, PC, or System i product that you are printing to. If you are printing to another System i product, you can start LPD with System i Navigator:
  - a. Click **Network > Servers > TCP/IP**.
  - b. Right-click **LPD** and click **Start**.
2. Use the Create Output Queue (CRTOUTQ) CL command to create a remote output queue. From a character-based interface, type CRTOUTQ.
3. On the CRTOUTQ display, press F9 to display all parameters.
4. Specify the following values:

Parameter	Value
Output queue (OUTQ)	The name of your output queue; for example, <i>myoutq</i> .
Library	The name of the library where your output queue is located. It is recommended that you set this to QUSRSYS in case you need to use a dummy device description for some of your applications; for example, <i>qusrsys</i> .
Remote system (RMTSYS)	<p>When configuring a remote output queue, you can either specify the printer's IP address, host name, or DNS name. If an IP address is used, the Remote System (RMTSYS) parameter should be set to *INTNETADR, and then the printer's IP address should be specified in the Internet address (INTNETADR) parameter. If a host name or DNS name is used, that name should be entered here.</p> <p>Verify that the system can reach the printer by using either the PING or Verify TCP/IP Connection (VFYTCPCNN) command with the IP address, host name, or DNS name.</p>
Remote printer queue (RMTPRTQ)	<p>This specifies the print queue value that the printer or print server accepts for LPD print requests. The remote printer queue is hardware specific. Refer to your manufacturer's information or see the IBM Software Knowledge Base document Recommended Remote Printer Queue Values for Remote Output Queues (RMTOUTQs)  (search for the document title at <a href="http://www.ibm.com/support/">www.ibm.com/support/</a>) for recommendations on the value to use for some popular printers.</p>
Writers to autostart (AUTOSTRWTR)	This should be set to 1 to automatically start one writer.
Connection type (CNNTYPE)	This must be set to *IP.
Destination type (DESTTYPE)	This should be set to *OTHER.
Host print transform (TRANSFORM)	Set to *YES so that SNA character string (SCS) and AFP data stream spooled files are converted to the printer's data stream.
Manufacturer type and model (MFRTYPMDL)	<p>This value indicates the type of ASCII printer and is used by the host print transform when converting spooled files. For recommendations on the value to use for some popular printers, see the IBM Software Knowledge Base document Information on Printers from Various Manufacturers  (search for the document title at <a href="http://www.ibm.com/support/">www.ibm.com/support/</a>).</p>
Internet address (INTNETADR)	This must be set to the printer's IP address when the Remote system (RMTSYS) parameter is set to *INTNETADR; for example, <i>192.168.0.2</i> .

Parameter	Value
Destination options (DESTOPT)	<p>This should be set to 'XAIX' to allow multiple copies to be printed on printers that start printing right away. When XAIX is specified, the remote printer writer opens the connection to the printer and sends multiple copies of the data file and control file. Each control file contains one print command for the preceding data file.</p> <p>This should be set to *NONE on printers that close the connection after the first copy is sent and that are capable of buffering all of the print data. The data file and the control file are sent once. The control file contains multiple print commands for the preceding data file. Depending on the specific printer, this might result in multiple copies being printed.</p> <p>If you specify 'XAIX' when using a printer that closes the connection after the first copy is sent, a condition might result where the print data is resent, causing a loop condition where the data is printed continuously.</p>
Print separator page (SEPPAGE)	This setting might be overridden by the print server, so setting this parameter might not have the effect you want.

All other values can be left as the default value.

5. Press Enter.
6. The remote writer associated with the remote output queue should automatically be started. Use the following tasks to manage the remote output queue and the remote writer:
  - Start the remote writer.
  - Stop the remote writer.
  - Display the remote writer status.
  - Change the remote output queue description.
7. Optional: Create a dummy device description. Dummy device descriptions are typically used when printing from applications that do not support printing to an output queue. Dummy device descriptions can also be used if you want to manage output associated with a printer rather than an output queue. For more information, see the IBM Software Knowledge Base document [Creating a Dummy Device Description to Use with a Remote Output Queue \(RMTOUTQ\)](#)  (search for the document title at [www.ibm.com/support/](http://www.ibm.com/support/)).

**Related concepts:**

“Starting a remote writer” on page 227

To start the remote writer, use the Start Remote Writer (STRRMTWTR) command to start the remote writer.

“Stopping a remote writer” on page 227

To stop the remote writer, use the End Writer (ENDWTR) command.

“Displaying remote writer status” on page 227

To display the writers, use the Work with All Writers (WRKWTR) command.

**Related tasks:**

“Changing remote output queue description” on page 232


To change a remote output queue description, follow these steps.

**Related information:**



Create Output Queue (CRTOUTQ) command

## Configuring IPP printers

To set up Internet Printing Protocol (IPP) printers, follow these steps.

1. Ensure that you have met the following requirements:
  - The internal network interface card or external print server must support the IPP. For a list of network interface cards or external print servers that are known to support IPP, see the IBM Software Knowledge Base document Recommended Remote Location (RMTLOCNAME) Values for \*LAN 3812 IPP Device Descriptions  (search for the document title at [www.ibm.com/support/](http://www.ibm.com/support/)).
  - If the printer is attached to an external network adapter, the printer must be connected with an IEEE 1284 bidirectional printer cable and the print server must be in the BITRONICS, BI-DIRECTIONAL or ECP/MLC mode.
2. Use the Create Device Description (Printer) (CRTDEVPRT) CL command to create a printer device description. From a character-based interface, type CRTDEVPRT.
3. On the Create Device Description (Printer) Display, press F9 to display all parameters.
4. Specify the following values.

Parameter	Value
Device description (DEVD)	The name of your device description. For example, <i>myippprt</i> .
Device class (DEVCLS)	This must be set to *LAN.
Device type (TYPE)	This must be set to 3812.
Device model (MODEL)	This must be set to 1.
LAN attachment (LANATTACH)	This must be set to *IP.
Port number (PORT)	This should be set to well-known port 631 for a nonsecure IPP connection. A secured connection can be set to 631, 6310, or any unused port.
Font (FONT)	This is typically set to 011, which is the default 10 CPI font.
Form feed (FORMFEED)	Set to *AUTOCUT, if single-cut sheets are used by the printer. Set to *CONT, if printing to a continuous forms printer.
Printer error message (PRTERMSG)	Set to *INFO so that the printer's message queue will receive informational messages whenever possible. This can prevent a condition that might cause the printer writer to end.
Message queue (MSGQ)	This value defaults to *CTLD, which indicates that the message queue is defined in the attached controller. There is no attached controller for *LAN printer device descriptions, so leaving the Message queue (MSGQ) parameter set to *CTLD will in effect set it to the QSYSOPR message queue in library QSYS. If you want to use another message queue for this printer device, verify that the message queue exists before entering the message queue and library in the MSGQ parameter. Otherwise, the Create Device Description (Printer) (CRTDEVPRT) command will fail with message CPF2799, Message queue &1 in library &2 not found.
Host print transform (TRANSFORM)	Set this parameter to *YES so that SNA character string (SCS) and AFP data stream spooled files are converted to the printer's data stream.

Parameter	Value
Manufacturer type and model (MFRTYPMDL)	This value indicates the type of ASCII printer and is used by the host print transform when converting spooled files. For recommendations on the value to use for some popular printers, see the IBM Software Knowledge Base document Information on Printers from Various Manufacturers  (search for the document title at <a href="http://www.ibm.com/support/">www.ibm.com/support/</a> ).
Remote location (RMTLOCNAME)	The name of your printer on the TCP/IP network; for example, <i>192.168.0.1</i> or <i>mynmpprt.myco.com</i> . You should verify that the system can reach the printer by using either the ping or Verify TCP/IP Connection (VFYTCPCNN) command. For recommendations on the value to use for some popular printers, see the IBM Software Knowledge Base document Recommended Remote Location (RMTLOCNAME) Values for *LAN 3812 IPP Device Descriptions  (search for the document title at <a href="http://www.ibm.com/support/">www.ibm.com/support/</a> ).
System driver program (SYSDRVPGM)	This must be set to *IBMIPDRV.
Secure connection (SECURECNN)	If you want a secure connection to the printer, set this value to *YES.
Validation list (VLDDL)	Optional. If specified, users must be validated before the printer will accept communication. See Set up validation lists for the IPP print driver topic for more information.

All other values can be left as the default value.

5. Press Enter.
6. Make the printer available.
7. Start the printer writer.

**Related tasks:**

“Making a printer available” on page 226

To make the printer available (vary on the printer), follow these steps.

“Starting a printer writer” on page 226

To start the printer writer, follow these steps.

**Related information:**

Create Device Desc (Printer) (CRTDEVPRT) command

**Setting up validation lists for the IPP print driver**

An Internet Printing Protocol (IPP) printer or IPP print server can require user authentication on requests from the IBM i IPP print driver. If this is the case, the validation list, specified in the VLDDL parameter of the printer device description, is used to look up the password for a given user ID.

A validation list holds pairs of user IDs and passwords. The IPP print driver checks the validation list for an entry in the following order:

1. The user profile that owns the spooled file
2. The printer device description name
3. The system name

If a match is found, the password will be retrieved for that entry and passed in subsequent requests to the printer. You might choose not to include individual user IDs in the list and instead provide the printer device description name or the system name as a generic entry to be used by multiple users.

If a match is not found, then the user authentication cannot be done.

Use the Create Validation List (CRTVLDL) command to create an empty validation list, as shown in the following example:

```
CRTVLDL VLDL(MYLIB/MYUSRS) AUT(*EXCLUDE) TEXT('My users')
```

This command creates an empty validation list named MYUSRS in the MYLIB library. The validation list is specified in the VLDL parameter of the printer device description. Use the AUT parameter to limit user access to the validation list.

After you create the validation list to be used by the IPP print driver, you populate the list with entries consisting of a user ID and a password that the system encrypts when the list is stored. You can add, change, and remove entries with the Validation list APIs.

Use the QsyAddValidationLstEntry() API to add an entry to a validation list. You must specify the QsyEncryptData attribute when adding entries to permit the retrieval of the encrypted password.

**Related information:**

Create Validation List (CRTVLDL) command

Validation list APIs

## Setting up the IPP server

You can administer and configure the Internet Printing Protocol (IPP) server using the IBM IPP Server Administrator for iSeries, a browser-driven graphical user interface.

Before using the interface, verify that the IBM HTTP Server for iSeries is installed on your system. For upgrade to Transport Layer Security (TLS), or Secure Sockets Layer (SSL) support, you must also install the Digital Certificate Manager and a Cryptographic Access Provider licensed program.

After the configurations previously mentioned are completed, you can manage the IPP server from the IBM IPP Server Administrator interface.

You can use a special configuration file to initialize the IPP server. The file is named QIBM/UserData/OS400/Ipp/conf/qippsvr-cust.conf. Do not manually edit this file. The configuration file is changed through the IBM IPP Server Administrator browser interface.

**Related tasks:**

“Managing IPP server” on page 227

You can administer the Internet Printing Protocol (IPP) server using the IBM IPP Server Administrator for iSeries, a browser-driven graphical user interface. Before using the interface, you must configure it.

## Verifying IPP server prerequisite programs

Before configuring the Internet Printing Protocol (IPP) server, you need to have these programs on your system.

The following licensed programs are required to configure, manage, and administer the IPP server:

- IBM HTTP Server for i5/OS (5761–DG1)
- IBM Developer Kit for Java (5761–JV1)

If you require printing over a secure connection, you also require Secure Sockets Layer (SSL) support. SSL support is obtained using the following licensed programs:

- Digital Certificate Manager (option 24 of 5761–SS1)
- Cryptographic access provider

## Configuring your Web browser

To allow the administrator interface to function correctly, JavaScript must be enabled for your Web browser. This can typically be enabled by accessing the preferences or options feature of your browser.

To select the display language for the administrator, access the language selection for your Web browser, and select or enter a supported locale that is installed on your system. Normally, there is a list to choose from. If a list is not displayed, you can manually enter a locale. The locale is typically a two letter abbreviation that represents a language. For example, English is represented by the letters "en". Sometimes the locale can further be defined by a country or region. For example, French spoken in Switzerland is represented by "fr\_CH". If the locale defined for the browser is not installed or supported by the administrator interface, the language associated with the user profile is used. If this language is not supported, English is used by default.

## Starting the administrator interface

You can use this command to start the \*ADMIN server instance.

```
STRTCPSVR SERVER(*HTTP) HTTPSVR(*ADMIN)
```

## Configuring an IPP server

To set up the Internet Printing Protocol (IPP) server, follow these steps.

1. Use the following URL to access the iSeries Tasks page:

```
http://system:2001
```

where *system* is the name of the system you are working with.

2. On the iSeries Tasks page, click the icon for the IBM IPP Server.

You should now be presented with the IBM IPP Server Administrator interface. To use the administrator interface, you must have \*IOSYSCFG authority, and you must have read and write access to the following property files:

- QIBM/UserData/OS400/Ipp/conf/qippsvr-cust.conf
- QIBM/UserData/OS400/Ipp/conf/printer.properties

In order to change preferences, you must have read and write access to the preference property file, QIBM/UserData/OS400/Ipp/conf/preferences.properties, if it exists.

3. If SSL connections are desired for any IPP printer, you must first enable the IPP server to use SSL by specifying an SSL port. To enable SSL connections:
  - a. Click the **Configuration** tab to load the configuration navigational menu.
  - b. Click **Basic Configuration** to load the basic configuration form.
  - c. Select the **Enabled** radio button.
  - d. Fill in a valid port number in the **SSL Port** text field.
  - e. Click the **Apply** button.

The IPP server is automatically configured to port 631 for non-SSL data. To support upgrade to TLS on port 631, or to support SSL on a specified port, a digital certificate must be associated with the IPP server. To do this, use Digital Certificate Manager for i5/OS (DCM) and associate a certificate with the IPP server, listed as server application QIBM\_IPP\_QIPPSVR.

To disable SSL connections, select the **Disabled** radio button on the form and click the **Apply** button.

**Note:** Whenever a change is made to the configuration of the IPP server, whether it is for SSL connections or logging information, the IPP server must be stopped and then started again for the change to become effective.

4. To configure how you want access logs created, written, and archived, follow these steps:
  - a. Click the **Configuration** tab to load the configuration navigational menu.
  - b. Expand the **Logs and Errors** menu item.
  - c. click **Access Logs** to load the access logs form.



- d. Fill in the form with the appropriate information.
  - e. Click the **Apply** button.
5. To configure how you want error logs created, written, and archived, perform the following steps:
    - a. Click the **Configuration** tab to load the configuration navigational menu.
    - b. Expand the **Logs and Errors** menu item.
    - c. click **Error Logs** to load the error logs form.
    - d. Fill in the form with the appropriate information.
    - e. Click the **Apply** button.

**Note:** The IBM IPP Server Administrator and the IBM IPP Server are two different applications. However, error logging must be enabled for the IPP server to allow administrator interface error logging to be enabled. If an error occurs using the IPP server, check the IPP server error log, specified using the error log form of the administrator interface, to help locate the problem.

### Configuring an IPP printer

To configure an Internet Printing Protocol (IPP) printer using the IPP Server Administration, follow these steps.

1. Click the **Configuration** tab to load the configuration navigational menu.
2. Expand the **Internet Printers** menu item.
3. Click **Create configuration** to start the create task wizard.
4. Follow the task steps to create an IPP printer configuration.
5. Click the **Finished** button on the confirmation panel to create the configuration.
6. Click **Display configuration**. Note the URL of the printer. You will need the URL when you configure your client.

### Configuring an IPP client

To configure your Internet Printing Protocol (IPP) client, specify the URL of the IPP printer.

For example, in Windows 2000, follow these steps:

1. Click **Start > Settings > Printers > Add Printer**.
2. Click **Next**.
3. Select **Network printer** and click **Next**.
4. Select **Connect to a printer on the Internet or on your intranet** and type the URL of your IPP printer into the **URL** field. Click **Next**. You might be prompted for a user ID and password, and for a security certificate, depending on the level of security that you chose.

If you receive a message about an unknown printer, click **OK** and select the appropriate printer driver.


5. Specify if you want this printer to be the Windows default printer and click **Next**.
6. Click **Finish**.

### Configuring Lexlink printers

To set up Lexlink printers, follow these steps.

1. Ensure that you understand the following requirements and restrictions:
  - All ASCII Lexlink protocol LAN-attached printers must be attached to the IBM 4033 LAN adapter device or the MarkNet XLe device, or the printer must have a MarkNet or MarkNet XL Internal Network Adapter (INA) card.
  - Only spooled files with device type attributes of \*SCS, \*USERASCII, or \*AFPDS are supported.
  - Direct printing (SPOOL parameter value = \*NO on the CRTPRTF, CHGPRTF, and OVRPRTF commands) is not allowed to ASCII Lexlink protocol LAN-attached printers.

- Printer sharing through the allow direct print (ALWDRTPRINT) parameter on the STRPRTWTR command is not supported. Printer sharing means having the SPOOL parameter value = \*NO on the CRTPRTF command and the ALWDRTPRINT parameter value = \*YES on the STRPRTWTR command.
  - Only Token Ring or Ethernet communications lines (values in the Switched Line List (SWTLINLST) parameter of the printer device description) are supported.
  - Error recovery will not be detailed.
  - Spooled files of type other than \*SCS, \*USERASCII, or \*AFPDS that reach an output queue associated with a ASCII Lexlink protocol LAN-attached printer are held. A message indicating the spooled file is held is sent to the message queue specified on the STRPRTWTR command.
2. Use the Create Device Desc (Printer) (CRTDEVPRT) CL command to create a printer device description. From a character-based interface, type CRTDEVPRT.
  3. On the Create Device Description (Printer) Display, press F9 to display all parameters.
  4. Specify the following values:

Parameter	Value
Device description (DEVDD)	The name of your device description. For example, <i>mylexprt</i> .
Device class (DEVCLS)	This must be set to *LAN.
Device type (TYPE)	This must be set to 3812.
Device model (MODEL)	This must be set to 1.
LAN attachment (LANATTACH)	This must be set to *LEXLINK.
LAN remote adapter address (ADPTADR)	Set to the MAC address of the LAN adapter, MarkNet device or MarkNet INA. The MAC address is typically specified on the packaging or on a label on the device.
Adapter type (ADPTTYPE)	Set to *INTERNAL for an internal MarkNet card. Set to *EXTERNAL for an external MarkNet device.
Port number (PORT)	The port number that the external device is expecting to receive data. For the MarkNet XLe, use the following values: <ul style="list-style-type: none"> <li>• 0 for serial port</li> <li>• 1 for parallel port 1</li> <li>• 2 for parallel port 2</li> </ul>
Font (FONT)	This is typically set to 011, which is the default 10 CPI font.
Form feed (FORMFEED)	Set to *AUTOCUT, if printing to a cut sheet printer. Set to *CONT, if printing to a continuous forms printer.
Manufacturer type and model (MFRTYPMDL)	This value indicates the type of ASCII printer and is used by the host print transform when converting spooled files. For recommendations on the value to use for some popular printers, see the IBM Software Knowledge Base document Information on Printers from Various Manufacturers  (search for the document title at <a href="http://www.ibm.com/support/">www.ibm.com/support/</a> ).

All other values can be left as the default value.

5. Press Enter.
6. Make the printer available.
7. Start the printer writer.

See Lexlink network attached printers for additional information about Lexlink network attached printing.

**Related concepts:**

“Lexlink-network-attached printers” on page 57

ASCII LAN-attached printers that use the Lexlink protocol must be attached to an IBM 4033 LAN adapter device or a MarkNet XLe device, or the printer must have a MarkNet or MarkNet XL Internal Network Adapter (INA) card in the printer. (The IBM 4039 Printer is an example of a printer with an INA card.)

**Related tasks:**

“Making a printer available” on page 226

To make the printer available (vary on the printer), follow these steps.

“Starting a printer writer” on page 226

To start the printer writer, follow these steps.

**Related information:**

Create Device Desc (Printer) (CRTDEVPRT) command

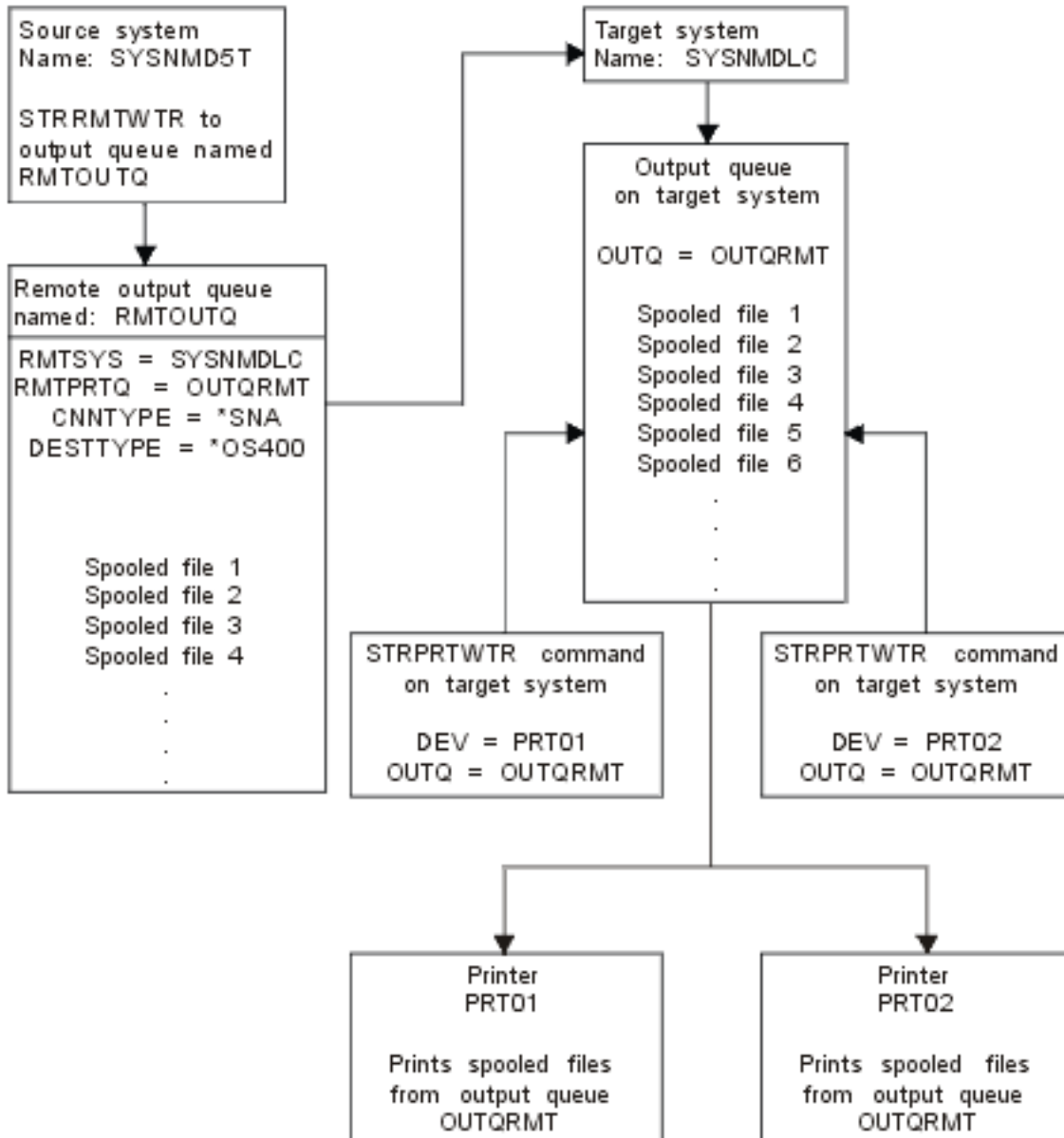
## **Configuring remote system printing**

Remote system printing includes options such as i5/OS to i5/OS and i5/OS to VM/MVS.

### **i5/OS to i5/OS for remote system printing**

The diagram in this topic illustrates i5/OS-to-i5/OS remote system printing with SNADS.

The values (or names) in the diagram are used in this example.



Preparation work must be done on both the source system and target system for the remote system printing to work. The following table provides a list of things that must be present or created before remote system printing can be used.

Table 4. Source system and target system preparation for remote system printing

Source system	Target system
Line, controller, and device descriptions created	Line, controller, and device descriptions created
Additional information for these configuration descriptions can be obtained in Communications Configuration PDF.	Additional information for these configuration descriptions can be obtained in Communications Configuration.

Table 4. Source system and target system preparation for remote system printing (continued)

Source system	Target system
<p>Identify the user profiles to be used for remote system printing.</p> <p>This information is needed if you want a corresponding user profile on the target system to own the sent spooled files. If you do not care who owns the spooled files, they can be sent to the target system and spooled to the QNETSPLF user profile.</p>	<p>Depending on where (to a user ID or output queue) you want the spooled files sent, you need to create a matching user profile or a specific output queue. Use the CRTUSRPRF or CRTOUTQ commands.</p>
<p>Set up the SNADS network.</p> <ul style="list-style-type: none"> <li>• Create distribution queues and routing entries using the Configure Distribution Services (CFGDSTSRV) command. <ul style="list-style-type: none"> <li>– Create a distribution queue.</li> <li>– Create a routing entry and associate the target system name (SYSNMDLC) with the distribution queue.</li> </ul> </li> <li>• Add users to the system directory. Use the Add Directory Entry (ADDDIRE) command. <ul style="list-style-type: none"> <li>– Add a directory entry for the QNETSPLF user profile. (This entry is used to send spooled files to the remote system.)</li> </ul> </li> </ul> <p><b>User ID</b> This must be QNETSPLF.</p> <p><b>Address</b> This is the value specified on the RMTSYS parameter (SYSNMDLC).</p> <p><b>User profile</b> This must be *NONE.</p> <p><b>System name</b> This is the value specified on the RMTSYS parameter (SYSNMDLC).</p> <p><b>System group</b> Not used in this example.</p> <ul style="list-style-type: none"> <li>– Add a directory entry for the QNETSPLF user profile. (This entry is the user ID used by SNADS on the source system.)</li> </ul> <p><b>User ID</b> This must be QNETSPLF.</p> <p><b>Address</b> This is the name of the source system (SYSNMD5T).</p> <p><b>User profile</b> This must be QNETSPLF.</p> <p><b>System name</b> This is the name of the source system (*LCL).</p> <p><b>System group</b> Not used in this example.</p>	<p>Set up the SNADS network.</p> <ul style="list-style-type: none"> <li>• Create distribution queues and routing entries using the Configure Distribution Services (CFGDSTSRV) command. <ul style="list-style-type: none"> <li>– Create a distribution queue.</li> <li>– Create a routing entry; associate the name of the distribution queue and the network ID of the target system (SYSNMD5T).</li> </ul> </li> <li>• Add users to the system directory. Use the Add Directory Entry (ADDDIRE) command. <ul style="list-style-type: none"> <li>– Add a directory entry for QNETSPLF. (This entry is used to receive spooled files from the source system.)</li> </ul> </li> </ul> <p><b>User ID</b> This value must be QNETSPLF.</p> <p><b>Address</b> This is the value specified on the RMTSYS parameter (SYSNMDLC).</p> <p><b>User profile</b> This must be QNETSPLF.</p> <p><b>System name</b> This must be *LCL.</p> <p><b>System group</b> Not used in this example.</p> <ul style="list-style-type: none"> <li>– Add another directory entry for user profile QNETSPLF. (This entry is used to send messages back to the source system.)</li> </ul> <p><b>User ID</b> This must be QNETSPLF.</p> <p><b>Address</b> This is the name of the source system (SYSNMD5T).</p> <p><b>User profile</b> This value must be *NONE.</p> <p><b>System name</b> This is the name of the source system (SYSNMD5T).</p> <p><b>System group</b> Not used in this example.</p>

Table 4. Source system and target system preparation for remote system printing (continued)

Source system	Target system
Create a remote output queue for remote system printing.	Create an output queue to receive the spooled files using the value on the RMTPRTO parameter (OUTQRMT).
Use the Create Output Queue (CRTOUTQ) command.	

### Source system activity - creating the remote output queue

The result of using this example is that spooled output files are sent to output queue OUTQRMT on the target system (SYSNMDLC).

Type CRTOUTQ and press F4 (Prompt). The following display appears. Type the values shown and press the Enter key. The remote output queue RMTOUTQ is created. The other parameter values are needed to ensure connection and delivery to the designated system.

```

Create Output Queue (CRTOUTQ)

Type choices, press Enter.

Output queue . . . . . RMTOUTQ   Name
Library . . . . . MYLIB         Name, *CURLIB
Maximum spooled file size:
Number of pages . . . . . *NONE   Number, *NONE
Starting time . . . . .           Time
Ending time . . . . .           Time
+ for more values

Order of files on queue . . . . . *FIFO   *FIFO, *JOBNBR
Remote system . . . . . SYSNMDLC
Remote printer queue . . . . . OUTQRMT
Queue for writer messages . . . . QSYSOPR   Name, QSYSOPR
Library . . . . . *LIBL         Name, *LIBL, *CURLIB
Connection type . . . . . *SNA    *SNA, *IP
Destination type . . . . . *OS400  *OS400, *OS400V2, *PSF2...
Text 'description' . . . . . @@ss1s@@ to @@ss1s@@

Bottom

F3=Exit  F4=Prompt  F5=Refresh  F10=Additional parameters  F12=Cancel
F13=How to use this display  F24=More keys

```

To begin sending spooled output files from the RMTOUTQ remote output queue, run the Start Remote Writer (STRRMTWTR) command. Type STRRMTWTR and press F4 (Prompt).

The following display appears. Type the values for the parameters displayed and press the Enter key. Spooled output files are then sent to the OUTQRMT on the target system (SYSNMDLC).

### Start Remote Writer (STRRTWTR)

Type choices, press Enter.

Output queue . . . . .	<i>RMTOUTQ</i>	Name, *ALL
Library . . . . .	<i>stumpf</i>	Name, *LIBL, *CURLIB
Queue for writer messages . . . .	<i>*OUTQ</i>	Name, *OUTQ, *REQUESTER
Library . . . . .		Name, *LIBL, *CURLIB
Form type options:		
Form type . . . . .	<i>*ALL</i>	Form type, *ALL, *STD, *FORMS
Message option . . . . .	<i>*NOMSG</i>	*NOMSG, *INQMSG, *MSG...

**Bottom**

F3=Exit F4=Prompt F5=Refresh F10=Additional parameters F12=Cancel  
F13=How to use this display F24=More keys

### Target system activity - printing spooled output files from the source system

The spooled output files were sent to the OUTQRMT output queue on the target system. To begin printing the spooled output files, start a printer writer to the OUTQRMT output queue. Type STRPRTWTR and press F4 (Prompt).

Type the name of the printer (CHEROKEE2) you want to use and the name of the output queue with the received spooled output files (OUTQRMT). Press the Enter key and spooled output files will begin to print on the printer named CHEROKEE2.

### Start Printer Writer (STRPRTWTR)

Type choices, press Enter.

Printer . . . . .	<i>cherokee2</i>	Name, *ALL, *SYSVAL
Output queue . . . . .	<i>OUTQRMT</i>	Name, *DEV
Library . . . . .	<i>stumpf</i>	Name, *LIBL, *CURLIB
Queue for writer messages . . . .	<i>*DEV</i>	Name, *DEV, *REQUESTER
Library . . . . .		Name, *LIBL, *CURLIB
Form type options:		
Form type . . . . .	<i>*ALL</i>	Form type, *ALL, *STD, *FORMS
Message option . . . . .	<i>*INQMSG</i>	*INQMSG, *MSG, *NOMSG...
File separators . . . . .	<i>*FILE</i>	0-9, *FILE
Drawer for separators . . . . .	<i>*DEV</i>	1-255, *DEV, *FILE

**Bottom**

F3=Exit F4=Prompt F5=Refresh F10=Additional parameters F12=Cancel  
F13=How to use this display F24=More keys

Multiple printer writers can be started to same output queue. This allows the spooled output files from a particular output queue to be printed on more than one printer.

To see the number of writers started to the OUTQRMT output queue, run the WRKOUTQ \*ALL command. The Work with All output queues screen appears. Type option 9 next to the OUTQRMT

output queue and press the Enter key. The Work with All Writers display appears. If the output queue has more than one writer started to it, the writers are listed and they both are associated with the same output queue (OUTQRMT).

**Work with All output queues**

Type options, press Enter.  
 2=Change 3=Hold 4=Delete 5=Work with 6=Release 8=Description  
 9=Work with Writers 14=Clear

Opt	Queue	Library	Files	Writer	Status
	SCCOUTQ	SCCLARK	156		RLS
	T93	SCCLARK	0		RLS
	AFP	SKS	23		RLS
	AFP2	SKS	0		RLS
	SKS2	SKS	0		RLS
	DEFERQ	STANGLER	5		HLD
	STANGLER	STANGLER	53		RLS
	ANGELIKA	STUMPF	0		RLS
9	OUTQRMT	STUMPF	2	*CHEROKEE2	RLS
	RMTOUTQ1	STUMPF	0		RLS
	TAAOUTQ	TAATool	0		RLS
	TIEMENS	TIEMENS	0		RLS

**More...**

Command  
 ==>

F3=Exit F4=Prompt F5=Refresh F12=Cancel F24=More keys

**Work with All Writers**

Type options, press Enter.  
 2=Change 3=Hold 4=End 5=Work with 6=Release 7=Display messages  
 8=Work with output queue

Opt	Writer	Type	Device	Queue	Library	Status	Form Type
	CHEROKEE2	PRT	CHEROKEE2	OUTQRMT	STUMPF	STR	*ALL
	CHIEF	PRT	CHIEF	OUTQRMT	STUMPF	STR	*ALL

**Bottom**

Parameters for options 2, 3, 4, 6 or command  
 ==>

F3=Exit F4=Prompt F12=Cancel F22=Start printer writer F24=More keys

You can list the names of writers started to the OUTQRMT output queue by typing WRKOUTQ OUTQRMT and pressing F4 (Prompt). The Work with Output Queue display appears. Press F20 (Writers) and the Work with All Writers display appears with a list of all writers that are started to the OUTQRMT output queue. This sequence is shown in the following two displays.



### Work with Output Queue

Queue: OUTQRMT      Library: STUMPF      Status: RLS/WTR

Type options, press Enter.

1=Send   2=Change   3=Hold   4=Delete   5=Display   6=Release   7=Messages  
8=Attributes      9=Work with printing status

Opt	File	User	User Data	Sts	Pages	Copies	Form Type	Pty
	DMB18R2	XZZ0136	*BEFORE	HLD	4	1	ENTN	5
	QPRINT	STUMPF		SAV	2	1	*STD	5

**Bottom**

Parameters for options 1, 2, 3 or command

====>

F3=Exit   F11=View 2   F12=Cancel   F20=Writers   F22=Printers  
F24=More keys

### Work with All Writers

Type options, press Enter.

2=Change   3=Hold   4=End   5=Work with   6=Release   7=Display messages  
8=Work with output queue

Opt	Writer	Type	Device	Queue	Library	Status	Form Type
	CHEROKEE2	PRT	CHEROKEE2	OUTQRMT	STUMPF	STR	*ALL
	CHIEF	PRT	CHIEF	OUTQRMT	STUMPF	STR	*ALL

**Bottom**

Parameters for options 2, 3, 4, 6 or command

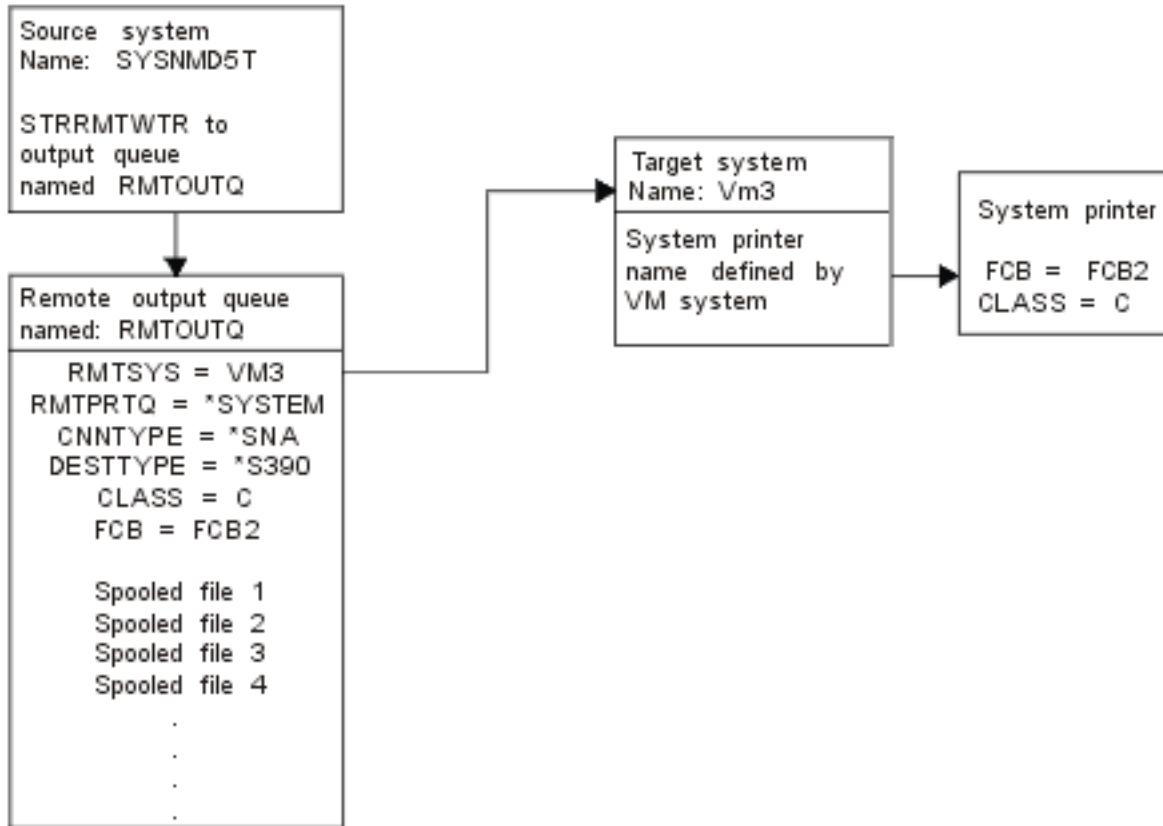
====>

F3=Exit   F4=Prompt   F12=Cancel   F22=Start printer writer   F24=More keys

## i5/OS to VM/MVS for remote system printing

The diagram in this topic illustrates i5/OS-to-VM/MVS remote system printing with Systems Network Architecture distribution services (SNADS).

The values (or names) in the diagram are used in this example.



Preparation work must be done on both the source system and target system for remote system printing to work. The following table provides a list of things that must be present or created before remote system printing can be used.

Table 5. Source system and target system preparation for remote system printing

Source system at V3R1 or later	VM/MVS target system
Ensure that licensed program 5769-CM1 is installed. This is needed for communications. For more details about configuring SNADS connections between these systems, see SNA Distribution Services PDF.	Network job entry (NJE) configuration for VM/RSCS and JES2 or JES3 for MVS.  Contact with the host system will be critical to establishing the correct connection with VM/MVS. System node name, printer ID, and user ID are some of the information i5/OS will require.
Line, controller, and device descriptions created  Additional information for these configuration descriptions can be obtained in Communications Configuration PDF.	Communication descriptions must exist or be created to connect the VM/MVS system to i5/OS. Typically this would be a line, physical unit (similar to controller), and a logical unit (similar to device). The VM or MVS system configuration will require information from i5/OS.
Identify the user profile to be used for remote system printing.	This user must also exist on the target system if *USER is specified on the RMTprtQ parameter on i5/OS.

Table 5. Source system and target system preparation for remote system printing (continued)

Source system at V3R1 or later	VM/MVS target system
<p>Set up the SNADS network.</p> <ul style="list-style-type: none"> <li>• Create distribution queues and routing entries using the Configure Distribution Services (CFGDSTSRV) command. <ul style="list-style-type: none"> <li>– Create a distribution queue.</li> <li>– Create a routing entry and associate the target system name (VM3) with the distribution queue.</li> </ul> </li> <li>• Add users to the system directory. Use the Add Directory Entry (ADDDIRE) command. <ul style="list-style-type: none"> <li>– Add a directory entry for the QNETSPLF user profile. (This entry is the user ID used by SNADS on the source system.)</li> </ul> </li> </ul> <p><b>User ID</b> This must be QNETSPLF.</p> <p><b>Address</b> This is the name of the source system (SYSNMD5T).</p> <p><b>User profile</b> This must be QNETSPLF.</p> <p><b>System name</b> This is the name of the source system (*LCL).</p> <p><b>System group</b> Not used in this example.</p> <ul style="list-style-type: none"> <li>– Add a directory entry for QNETSPLF. (This entry is the user ID used by SNADS to route to the target system.)</li> </ul> <p><b>User ID</b> This must be QNETSPLF.</p> <p><b>Address</b> This is the name of the target system (VM3).</p> <p><b>User profile</b> This must be *NONE.</p> <p><b>System name</b> This is the name of the target system (VM3).</p> <p><b>System group</b> Not used in this example.</p>	<p>For MVS, identification of the System i product must be entered in the job control language (JCL) or Job Entry Subsystem (JES) and associated with a printer defined to the MVS system.</p> <p>For VM, identification of the System i server must be entered in RSCS and associated with a printer defined to the VM system.</p>
<p>Create a remote output queue for remote system printing.</p> <p>Use the Create Output Queue (CRTOUTQ) command.</p>	
<p>When sending spooled output files to VM/MVS, the default forms type must be STANDARD (not to be confused with *STD for i5/OS). The forms type can be changed by overriding the printer file used to create the spooled output files. Also, you could use the Change Spooled File Attributes (CHGSPLFA) command and change the forms type parameter value to standard.</p>	

## Source system activity - creating the remote output queue

The result of using this example would be spooled output files sent to the system printer on the target system (VM3).

**Note:** When a remote output queue is created, writers are automatically started to it if the AUTOSTRWTR parameter value is other than \*NONE.

Type CRTOUTQ and press F4 (Prompt). The following display appears. Type the values shown and press the Enter key. The remote output queue TEST1 is created. The other parameter values are needed to ensure connection and delivery to the designated system.

```

Create Output Queue (CRTOUTQ)

Type choices, press Enter.

Output queue . . . . . RMTOUTQ Name
Library . . . . . MYLIB Name, *CURLIB
Maximum spooled file size:
  Number of pages . . . . . *NONE Number, *NONE
  Starting time . . . . . Time
  Ending time . . . . . Time
    + for more values
Order of files on queue . . . . . *FIFO *FIFO, *JOBNBR
Remote system . . . . . VM3
Remote printer queue . . . . . *SYSTEM
Queue for writer messages . . . . . QSYSOPR Name, QSYSOPR
  Library . . . . . Name, *LIBL, *CURLIB
Connection type . . . . . *SNA *SNA, *IP
Destination type . . . . . *S390 *OS400, *OS400V2, *PSF2...
VM/MVS class . . . . . C
FCB . . . . . FCB2
Text 'description' . . . . . i5/OS to VM System named VM3

F3=Exit F4=Prompt F5=Refresh F10=Additional parameters F12=Cancel
F13=How to use this display F24=More keys
Bottom
```

To begin sending spooled output files from the RMTOUTQ remote output queue, run the Start Remote Writer (STRRMTWTR) command by typing STRRMTWTR and pressing F4 (Prompt).

The following display appears. Type RMTOUTQ and press the Enter key. Spooled output files are then sent to the printer designated as the system printer on the target system (VM3).

```

Start Remote Writer (STRRMTWTR)

Type choices, press Enter.

Output queue . . . . . RMTOUTQ Name, *ALL
Library . . . . . MYLIB Name, *LIBL, *CURLIB
Queue for writer messages . . . . . *OUTQ Name, *OUTQ, *REQUESTER
  Library . . . . . *LIBL Name, *LIBL, *CURLIB
Form type options:
  Form type . . . . . *ALL Form type, *ALL, *STD, *FORMS
  Message option . . . . . *NOMSG *NOMSG, *INQMSG, *MSG...

F3=Exit F4=Prompt F5=Refresh F10=Additional parameters F12=Cancel
F13=How to use this display F24=More keys
Bottom
```

## Target system activity - printing spooled output files from the source system

The spooled files were sent to the system printer on the target system (VM3).

## Configuring host print transform

The host print transform is enabled when you specify \*YES for the TRANSFORM parameter in the printer device description.

The TRANSFORM parameter can be specified when the printer device description is created or when you change an existing printer device description. Because of the complexity of creating a device description manually, it is recommended that you use automatic configuration. Then, after the device description has been created, change the device description to enable the host print transform.

### Notes:

1. Automatic configuration of devices (printers, in this case) attached to the ASCII workstation controller is not supported. The Create Device Description (Printer) (CRTDEVPRT) command must be used to create the printer device description.
2. IBM i Access for Windows PC5250 creates or changes its printer device descriptions based on the printer's session configuration. For this emulator, the host print transform should be enabled by changing the session configuration on the personal computer and not the device description in the IBM i operating system.

## Customizing ASCII printers that use the host print transform function

You can use the workstation customization functions to customize the functions of ASCII printers that use the host print transform function.

Using the workstation customization functions, you can do the following customizations:

- Customize the functional characteristics of a supported ASCII printer
- Customize the functional characteristics and specify all necessary parameters required to support a normally unsupported ASCII printer

To customize the functional characteristics of an ASCII printer, follow these steps:

1. Prepare for the customization.
2. Retrieve the workstation customization source.
3. Change the workstation customization source.
4. Create a customization object that contains the changed printer attributes.
5. Change the printer device description to specify the customization object.

### Preparing to customize an ASCII printer:

You must gather source materials, complete printer setup, and plan adequate time in your schedule to customize an ASCII printer.

#### *Gathering source materials:*

Before you can begin customizing an ASCII printer, you must have information about the functions the ASCII printer supports. (You can only add or change printing functions that a printer supports.)

You also need the hexadecimal values for these functions. The hexadecimal code information for your device is critical to the workstation customization process. Often, the reference manual for the printer provides this information.

### *Completing printer setup:*

Before you begin printer customization, complete these steps to set up both supported and unsupported printers.

- Set up all the necessary hardware to connect the printer to the system.
- Set up any programmable features that are provided by the printer.  
This might involve some internal programming on the printer itself, setting device independent programming (DIP) switches, or selecting a printer to emulate. If you are using a non-IBM printer, check the reference manual to determine if it emulates any IBM printers. If it does, set the emulation for the IBM printer. This might simplify the customization process.
- Create the necessary controller and device descriptions, if they do not already exist. Some device descriptions for printers can be automatically created using automatic configuration.

After setting up and turning on the ASCII printer, use one of your applications to print a short test document. This is the starting point for workstation customization.

### *Planning the customization schedule:*

Customizing an ASCII printer might involve a trial-and-error process.

The amount of work that is required to customize a printer depends on these conditions:

- The type of printer
- Whether the printer is already printing
- The completeness of the manual for the printer

You need to plan anywhere from 1 to 5 days to complete a successful ASCII printer customization.

### *Customizing unsupported ASCII printers:*

To customize an unsupported ASCII printer, ask these questions.

- Can the printer emulate a supported ASCII printer?  
If so, set it up to use the emulation. It could make your customizing easier.
- What printer functions and national characters do I want this printer to support?  
Write these down so that you can answer the next question.
- Does the printer itself support the functions I need?  
Check the manual to determine this. If the printer cannot support the functions you need, you cannot customize the printer to provide these functions.

### **Retrieving the workstation customization source:**

To begin customizing an ASCII printer that uses the host print transform function, you must retrieve the source file member that contains the workstation customization source.

You can then change your printer's characteristics. When retrieving the source file member, specify the manufacturer, type, and model of the ASCII printer you want to customize. The source code you retrieve is a copy of the transform table for the printer manufacturer, type, and model you specify.

### *Understanding the transform table:*

The host print transform function uses a transform table to transform the printer data stream that is sent by the IBM i operating system to an attached ASCII printer.

These are some of the functional characteristics within the transform table:

- Line spacing
- Pitch (characters per inch)
- Page size
- Highlighting characteristics (bold, underline)
- Draft, letter, or text quality printing
- Subscripting and superscripting
- Initialization and reset sequences

*Choosing the customizing source:*

When you decide to customize an ASCII printer, you need to retrieve the correct source for your printer.

Use the Retrieve WSCST source (RTVWSCST) command to retrieve the source.

1. Type RTVWSCST on a command line and press Enter.
2. Specify a device type of \*TRANSFORM or \*CTXFORM.
3. Specify the printer manufacturer, type, and model (MFRTYPMDL parameter). To see a complete list of ASCII printers supported, press F4 (Prompt) on the **Manufacturer type and model** field.

**Notes:**

- If you are customizing a laser printer that is not listed, you might want to select the \*HP11 value. Many laser printers have printer functions similar to those provided by the Hewlett-Packard LaserJet Series II.
  - The MFRTYPMDL special values of \*PDF, \*PDFEMBEDTT, and \*PDFIBMWT refer to \*WSCST objects with the attribute of CTXFORM.
4. Specify a name for the source file member to be created for the transform table you want to retrieve. This should be a name you can easily remember.
  5. Specify a library and source file name in which to store the source file member you specified in the previous step. The library you specify must exist.
  6. Specify a text description for the source file member if it does not already have one. This description should be unique.

The system provides you with a transform table for a specific ASCII printer.

**Customizing WSCST object of \*TRANSFORM transform type:**

You can do the following customization to your ASCII printer when you specify \*TRANSFORM as the WSCST object's transform type.

*Changing the source:*

You can use printer function tags to change the ASCII control sequence for an individual printer function.

Using the source entry utility (SEU), you can change, add, or delete the printer function tags by editing the source file member for the retrieved source.

The source file member has the following format:

```
:WSCST DEVCLASS=TRANSFORM.
  :TRNSFRMTBL.
  .
  .
  tag or comment
  .
  .
:EWSCST.
```

**Note:** A period must follow the last data element within a tag entry.

*Transform Table (TRNSFRMTBL) tag:*

The Transform Table (TRNSFRMTBL) tag defines a transform table for an ASCII printer that uses the host print transform function.

The syntax for this tag is as follows:

:TRNSFRMTBL.

There are no parameters that are associated with this tag. The tag is immediately followed by a number of separate printer function tags that make up the table entries.

**Related information:**

*“Using the tags”*

Here are the descriptions of the valid printer functions and corresponding tags for a device class of TRANSFORM. These tags must follow the transform table tag (TRNSFRMTBL) in your source.

*Using the tags:*

Here are the descriptions of the valid printer functions and corresponding tags for a device class of TRANSFORM. These tags must follow the transform table tag (TRNSFRMTBL) in your source.

The syntax for each tag is shown and is followed by the tag description. Some tags have only a data parameter. Other tags have more than one parameter to describe variables within the functions.

**Note:** A period must follow the last data element within a tag entry.

The printer function tags are divided into the following general tasks for customizing:

- Printer information
- Printer controls
- Print media size
- Highlighting
- Horizontal movement
- Vertical movement
- Indexing
- Color
- Paper input
- Paper output
- Print orientation
- Print quality
- Fonts
- Code page support
- Fax support
- Double-byte character set (DBCS) support

**Related information:**

*“Transform Table (TRNSFRMTBL) tag”*

The Transform Table (TRNSFRMTBL) tag defines a transform table for an ASCII printer that uses the host print transform function.

*“Customizing Double-Byte Character Set (DBCS) support” on page 187*  
You can customize DBCS support with these tags.



*Programming considerations:*

In general, when you customize a transform table, the host print transform function uses only tags that are defined in your customization object.

You can delete function tags from the retrieved source when creating your customization object. The host print transform function then assumes a null value for the ASCII control sequence for that function.

The absence of a specific printer function tag after the TRNSFRMTBL tag implies that particular printer function is not mapped and cannot be used. If the same printer function tag occurs multiple times following a TRNSFRMTBL tag, warning messages are not sent to the job log. The customization object uses the last occurrence of the tag in the source to map the printer function.

Programming considerations for specific tags, or specific tag sets, are included in the description of those tags.

*Customizing printer information:*

By customizing printer information, you can control how the host print transform processes IBM i print jobs.

You can customize the following printer information:

- Printer data stream
- Raster mode
- No-print borders
- Drawer page size

*Printer Data Stream (PRTDTASTRM) tag:*

The Printer Data Stream (PRTDTASTRM) tag identifies the printer data stream or raster image format generated by the host print transform.

The syntax for this tag is as follows:

```
:PRTDTASTRM
  DATASTREAM = NULL|IBMPPDS1|IBM3812|HPPCL4|
               IBMPPDS2|EPSON|NEC|IBMGRAPHICS|
               HPPCL5|HPPCL4I|HPPCL5I|IBMPPDS3|
               IBMPPDS3I|IBMPDS4|IBMPPDS4I|
               TIFF_PB|TIFF_G4|IOCA_G3MH|IOCA_G3MRK2|
               IOCA_G3MRK4|IOCA_IBMMMR|IOCA_NOCOMP.
```

**DATASTREAM**

A required parameter. The DATASTREAM tag specifies the type of printer data stream supported. The tag can have the following values.

Value	Description	Raster Mode Supported
EPSON	The Epson printer data stream is supported.	No
HPPCL4	The Hewlett-Packard PCL4 printer data stream is supported.	Yes
HPPCL4I 1	The Hewlett-Packard PCL4 printer data stream is supported in raster mode. This value means the same to the SCS-to-ASCII transform as HPPCL4. For the AFP-to-ASCII transform, this value causes it to go into raster mode for a PCL4 printer.	Yes
HPPCL5	The Hewlett-Packard PCL5 printer data stream is supported.	Yes

Value	Description	Raster Mode Supported
HPPCL5I <sub>1</sub>	The Hewlett-Packard PCL5 printer data stream is supported in raster mode. This value means the same to the SCS-to-ASCII transform as HPPCL5. For the AFP-to-ASCII transform, this value causes it to go into raster mode for a PCL5 printer.	Yes
IBMGRAPHICS	The IBM graphics printer data stream is supported.	No
IBMPPDS1	The IBM page printer data stream level 1 is supported.	No
IBMPPDS2	The IBM page printer data stream level 2 is supported.	No
IBMPPDS3	The IBM personal printer data stream level 3 is supported. This is used for the IBM 4019 printer. Supported functions over level 2 are rotation and image.	Yes
IBMPPDS3I <sub>1</sub>	The IBM personal printer data stream level 3 is supported in raster mode. This value means the same to the SCS-to-ASCII transform as IBMPPDS3. For the AFP-to-ASCII transform, this value causes the printer data stream to go into raster mode for a PDS level 3 (4019) printer.	Yes
IBMPPDS4	The IBM personal printer data stream level 4 is supported. This is used for the IBM 4029 printer. Supported functions over level 3 are multiple rotations on a page and compressed image.	Yes
IBMPPDS4I <sub>1</sub>	The IBM personal printer data stream level 4 is supported in raster mode. This value means the same to the SCS-to-ASCII transform as IBMPPDS4. For the AFP-to-ASCII transform, this value causes the printer data stream to go into raster mode for a PPDS level 3 (4029) printer.	Yes
IBM3812	The IBM 3812 printer data stream is supported.	No
IOCA_G3MH	IBM IOCA raster image format with G3 modified Huffman compression.	Yes
IOCA_G3MRK2	IBM IOCA raster image format with G3 modified read K=2 compression.	Yes
IOCA_G3MRK4	IBM IOCA raster image format with G3 modified read K=4 compression.	Yes
IOCA_IBMMMR	IBM IOCA raster image format with IBM modified read compression.	Yes
IOCA_NOCOMP	IBM IOCA raster image format with no compression.	Yes
NEC	The NEC printer data stream is supported.	No
NULL	The printer data stream supported is not known.	No
TIFF_G4	Tagged Image File Format with G4 compression.	Yes
TIFF_PB	Tagged Image File Format with packed bit compression.	Yes
<b>Note:</b>		
1. These types are valid, but it is recommended that you should use the raster mode tag.		

#### Related information:

“Raster Mode (RASTERMODE) tag”

The host print transform supports two output modes: mapping mode and raster mode.

*Raster Mode (RASTERMODE) tag:*

The host print transform supports two output modes: mapping mode and raster mode.

In *mapping mode*, IBM i printer commands and data are converted to similar ASCII printer commands and data. The default mode is mapping mode.

In *raster mode*, a raster image of the IBM i print job is created. The Raster Mode (RASTERMODE) tag specifies that the host print transform should generate raster image output. The PRTDTASTRM (printer data stream) tag determines the format of the raster image output.

**Note:** Not all data streams values support raster mode.

The RASTERMODE tag enables raster mode that is based on the IBM i print job type. The syntax is as follows:

```
:RASTERMODE
  SCS = YES | NO
  AFP = YES | NO.
```

### SCS

This is a required parameter. SCS determines if the host print transform converts the SCS print data stream to the raster mode image format that is specified in the printer data stream tag.

### AFP

This is a required parameter. AFP determines if the host print transform converts the AFP print data stream to the raster mode image format that is specified in the printer data stream tag.

### Related information:

“Printer Data Stream (PRTDTASTRM) tag” on page 145

The Printer Data Stream (PRTDTASTRM) tag identifies the printer data stream or raster image format generated by the host print transform.

*No-print Border (NOPRTBDR) tag:*

You can use the No-print Border (NOPRTBDR) tag to define the physical area on which printing cannot be done.

The no-print border size is based on the physical restrictions of your printer. You might find the no-print border referred to as the unprintable area when looking it up in your printer manual.

If the no-print border is defined, it is included in the top, bottom, left, and right margins that are used for an IBM i print job. For example, if you specify a value of 1/2 inch for the top, portrait, no-print border, and the IBM i document sets a top margin of 1 inch, the host print transform only advances your ASCII printer position 1/2 inch. The effect is that the printed document has a 1-inch top margin.

The syntax for this tag is as follows:

```
:NOPRTBDR
  OPTION = TOP|LEFT|RIGHT|BOTTOM
  ORIENT = PORTRAIT|LANDSCAPE
  DATA = no print border size in
          1440ths of an inch (integer).
```

### OPTION

A required parameter. This parameter defines the border type.

TOP	Sets no-print border at the top of the paper.
LEFT	Sets no-print border on the left side of the paper.
RIGHT	Sets no-print border on the right side of the paper.
BOTTOM	Sets no-print border at the bottom of the paper.

## ORIENT

A required parameter. This parameter defines the type of paper orientation. Some printers have different unprintable areas based on the orientation of the printed output.

PORTRAIT	The no-print border specified is for portrait-oriented print.
LANDSCAPE	The no-print border specified is for landscape-oriented print.

## DATA

A required parameter. This parameter specifies the size of the no-print area in 1/1440-inch increments. This value must be an integer.

### *Drawer Page Size:*

The Drawer Page Size (DWRPAGSIZ) tag defines the size of the paper present in a specific paper drawer.

In raster mode, this paper size is used to allocate the raster page buffer. The SCS-to-ASCII transform uses this paper size to determine if COR mode is required.

If the drawer size is not defined, the host print transform uses the page size that is specified in the IBM i print job.

You can define a drawer size for one or more of the drawers. You begin the syntax for the drawer page size with the DWRPAGSIZ tag and end the syntax with the EDWRPAGSIZE tag. The DWRPAGSIZE tags fall between, defining a drawer and its size.

If you use the drawer page size tags, your syntax could look something like this for a printer supporting two different drawers.

```
:DWRPAGSIZ.  
  :DWRPAGSIZE...  
  :DWRPAGSIZE...  
:EDWRPAGSIZ.
```

The following information describes the tags used to support different drawer sizes.

### *Drawer Page Size (DWRPAGSIZ) tag:*

The Drawer Page Size (DWRPAGSIZ) tag defines the page size for a drawer. This tag must be followed by one or more DWRPAGSIZE tags that contain the drawer size.

There are no parameters for this tag. The syntax is as follows:

```
:DWRPAGSIZ.
```

### *Drawer Page Size Entry (DWRPAGSIZE) tag:*

The Drawer Page Size Entry (DWRPAGSIZE) tag defines the page size for a particular source drawer.

The DWRPAGSIZ tag must be followed by one or more DWRPAGSIZE tags. The EDWRPAGSIZE tag must follow a group of DWRPAGSIZE tags. The syntax is as follows:

```
:DWRPAGSIZE  
  DRAWER = 0-255  
  PAGWTH = Page width  
  PAGLEN = Page length.
```

## DRAWER

This is a required parameter. This parameter defines the drawer.

0 The envelope drawer is specified.

1-255 The drawer 1-255 is specified.

#### **PAGWTH**

This is a required parameter. This parameter specifies the drawer page width in 1/1440 inch increments. This value must be an integer. The width is the left-to-right dimension of a piece of paper.

#### **PAGLEN**

This is a required parameter. This parameter specifies the drawer page length in 1/1440 inch increments. This value must be an integer. The width is the top-to-bottom dimension of a piece of paper.

*End Drawer Page Size (EDWRPAGSIZ) tag:*

The End Drawer Page Size (EDWRPAGSIZ) tag is used to end a group of drawer page size entries in the customizing syntax.

There are no parameters for this tag. The syntax is as follows:

```
:EDWRPAGSIZ.
```

*Customizing printer controls:*

You can customize these general printer control functions.

- Bell
- Carrier return
- Initialize printer
- Reset printer
- End raster graphics
- Print all characters
- Print next character

*Bell (BELL) tag:*

The Bell (BELL) tag defines the ASCII control sequence for the bell function for an ASCII printer.

The bell function sounds the printer bell or beeper briefly. The syntax for this tag is as follows:

```
:BELL  
    DATA = ASCII control sequence.
```

#### **DATA**

A required parameter. This parameter specifies the ASCII control sequence for the bell function. This must be a hexadecimal value.

*Carrier Return (CARRTN) tag:*

The Carrier Return (CARRTN) tag defines the ASCII control sequence for the carrier return function for an ASCII printer.

The CARRTN function returns the carrier to the left margin, without advancing a line. The syntax for this tag is as follows:

```
:CARRTN  
    DATA = ASCII control sequence.
```

#### **DATA**

A required parameter. This parameter specifies the ASCII control sequence for the carrier return function. This must be a hexadecimal value.

*Initialize Printer (INITPRT) tag:*

The Initialize Printer (INITPRT) tag defines the ASCII control sequence for the initialize printer function for an ASCII printer.

The host print transform function uses the INITPRT tag at the beginning of each print job. The syntax for this tag is as follows:

```
:INITPRT  
    DATA = ASCII control sequence.
```

**DATA**

A required parameter. This parameter specifies the ASCII control sequence for the initialize printer function. This must be a hexadecimal value.

*Reset Printer (RESETPRT) tag:*

The Reset Printer (RESETPRT) tag defines the ASCII control sequence for the reset printer function for an ASCII printer.

The RESETPRT tag is used by the host print transform at the end of each print job. The syntax for this tag is as follows:

```
:RESETPRT  
    DATA = ASCII control sequence.
```

**DATA**

A required parameter. This parameter specifies the ASCII control sequence for the reset printer function. This must be a hexadecimal value.

*Raster Graphics End (RASEND) tag:*

The Raster Graphics End (RASEND) tag causes the printer to end raster graphics (image) mode.

The syntax of this tag follows:

```
:RASEND  
    DATA = ASCII control sequence.
```

**DATA**

A required parameter. This parameter specifies the printer control sequence for the raster-graphics end-printer function. This must be a hexadecimal value.

*Print All Characters (PRTALLCHR) tag:*

The Print All Characters (PRTALLCHR) tag causes the printer to interpret the bytes that follow as printable characters rather than control codes.

The PRTNXTCHR provides the same function for 1 byte. The syntax is:

```
:PRTALLCHR  
    VAROFFSET = variable offset in control sequence  
    VARLEN = variable length  
    VARTYPE = HIGHLOW|LOWHIGH|CHRDEC|CHRHEX|CHRAN  
    DATA = ASCII control sequence.
```

**VAROFFSET**

This is a required parameter. This parameter specifies the offset for the variable portion of the control sequence. This value must be an integer.

**Note:** This offset is relative to the beginning of the control sequence. Therefore, a value of 0 implies that the variable starts in the first byte of the control sequence.

**VARLEN**

This is a required parameter. This parameter specifies the length of the variable portion of the control sequence. This value must be an integer (number of bytes).

**VARTYP**

A required parameter. This parameter specifies the type of variable that is used with the category of printer function tags.

HIGHLOW	The byte order of the variable is in high-low order. The most significant byte is first.
LOWHIGH	The byte order of the variable is in low-high order. The most significant byte is last.
CHRDEC	The variable is in character decimal format with no byte order consideration. All characters are in the range of 0 to 9. Many Hewlett-Packard ASCII printers use this type of variable.
CHRHEX	The variable is in character hexadecimal format with no byte order consideration. All characters are in the range of A to F.
CHRAN	The variable is in character alphanumeric format with no byte order consideration. All characters are in the range of 0 to 9, and A to Z.

**DATA**

A required parameter. This parameter specifies the ASCII control sequence for the set top margin (inches) printer function. This must be a hexadecimal value.

*Print Next Character (PRTNXTCHR) tag:*

The Print Next Character (PRTNXTCHR) tag causes the next byte of data to be treated as a character rather than a printer control.

The syntax of this tag follows:

```
:PRTNXTCHR
  DATA      = ASCII control sequence.
```

**DATA**

A required parameter. This parameter specifies the ASCII control sequence for the print-next character printer function. This must be a hexadecimal value.

*Customizing print media size:*

With the workstation customizing functions, you can customize print media size for an ASCII printer.

You can perform the following customizations:

- Customize envelope size
- Customize paper size
- Customize page length
- Customize printable area

*Envelope size:*

You can customize selecting various sizes of envelopes.

You begin the syntax for various envelope sizes with an ENVSIZE tag, and end the syntax with an EENVSIZEXFM tag. The ENVSIZE tags fall between the start and end tags to define each different envelope size.

If you use the envelope definition tags, your source could look something like this for a printer supporting two different envelope sizes:

```
:ENVSIZXFM.  
    :ENVSIZ. . . .  
    :ENVSIZ. . . .  
:EENVSIZXFM.
```

Following is a description of each of the tags used to support different envelope sizes for an ASCII printer.

### **Envelope Size for Transform (ENVSIZXFM) tag**

The Envelope Size for Transform (ENVSIZXFM) tag begins a group of different envelope size entry tags.

The ENVSIZXFM tag must be followed by one or more envelope size entry tags (ENVSIZ) containing the envelope sizes. The syntax for the ENVSIZXFM tag is as follows:

```
:ENVSIZXFM.
```

### **Envelope Size Entry (ENVSIZ) tag**

The Envelope Size Entry (ENVSIZ) tag defines an ASCII control sequence to set one envelope size for the transform table.

One or more of these tags follow an ENVSIZXFM tag. A group of one or more of these tags must be followed by an EENVSIZXFM tag. The syntax for the ENVSIZ tag is as follows:

```
:ENVSIZ  
    ENVWTH = envelope width  
    ENVLEN = envelope length  
    DATA = ASCII control sequence.
```

#### **ENVWTH**

A required parameter. This parameter specifies the envelope width in 1/1440-inch increments. This value must be an integer. The width is the left-to-right dimension when the envelope is in its normal orientation.

#### **ENVLEN**

A required parameter. This parameter specifies the envelope length in 1/1440-inch increments. This value must be an integer. The length is the top-to-bottom dimension when the envelope is in its normal orientation.

#### **DATA**

A required parameter. This parameter specifies the ASCII control sequence for the specified envelope size entry. This must be a hexadecimal value.

### **End Envelope Size for Transform (EENVSIZXFM) tag**

The End Envelope Size for Transform (EENVSIZXFM) tag ends a group of envelope size entries.

The syntax for this tag is as follows:

```
:EENVSIZXFM.
```

*Paper size:*

You can customize selecting various sizes of paper.

You begin the syntax for various paper sizes with a PAGESIZXFM tag, and end the syntax with an EPAGESIZXFM tag. The PAGESIZE tags fall between the start and end tags to define each different page size.



If you use the page size definition tags, your source could look something like this for a printer supporting five separate page sizes:

```
:PAGSIZXFM.  
    :PAGSIZE....  
    :PAGSIZE....  
    :PAGSIZE....  
    :PAGSIZE....  
    :PAGSIZE....  
:EPAGSIZXFM.
```

Following is a description of each of the tags you can specify for different page sizes on ASCII printers.

### **Page Size for Transform (PAGSIZXFM) tag**

The Page Size for Transform (PAGSIZXFM) tag defines the ASCII control sequences for a group of different page sizes.

The PAGSIZXFM tag must be followed by one or more PAGSIZE (page size entry) tags containing the page sizes. The syntax for the PAGSIZXFM tag is as follows:

```
:PAGSIZXFM.
```

### **Page Size Entry (PAGSIZE) tag**

The Page Size Entry (PAGSIZE) tag defines the ASCII control sequence to set one page size.

One or more of these tags must follow the PAGSIZXFM (set page size for transform) tag. A group of PAGSIZE tags must be followed by an EPAGSIZXFM (end set page size for transform) tag. The syntax for this tag is as follows:

```
:PAGSIZE  
    PAGWTH = page width  
    PAGLEN = page length  
    DATA = ASCII control sequence.
```

#### **PAGWTH**

A required parameter. This parameter specifies the page width in 1/1440-inch increments. This value must be an integer. The width is the left-to-right dimension of a piece of paper when it is in its normal orientation.

#### **PAGLEN**

A required parameter. This parameter specifies the page length in 1/1440-inch increments. This value must be an integer. The length is the top-to-bottom dimension of a piece of paper when it is in its normal orientation.

#### **DATA**

A required parameter. This parameter specifies the ASCII control sequence for selecting a specific page size entry. This must be a hexadecimal value.

### **End Page Size for Transform (EPAGSIZXFM) tag**

The end page size for transform (EPAGSIZXFM) tag ends a group of page size entries in the customizing source.

The syntax for this tag is as follows:

```
:EPAGSIZXFM.
```

Page length:

You can set the page length in lines or in inches.

Here are the tags used to customize the page length.

Page Length in Inches (PAGLENI) tag:

The Page Length in Inches (PAGLENI) tag sets the page length in terms of inches.

The syntax for this tag follows:

```
:PAGLENI
  VAROFFSET = variable offset in
              control sequence
  VARLEN = variable length
  VARTYPE = HIGHLOW|LOWHIGH|CHRDEC|
           CHRHEX|CHRAN
  CNVNUM = conversion ratio
           numerator
  CNVDEN = conversion ratio
           denominator
  DATA = ASCII control sequence.
```

#### **VAROFFSET**

A required parameter. This parameter defines the offset for the variable portion of the control sequence. This value must be an integer.

**Note:** This offset is relative to the beginning of the control sequence. Therefore, a value of 0 implies that the variable starts in the first byte of the control sequence.

#### **VARLEN**

A required parameter. This parameter defines the length of the variable portion of the control sequence. This value must be an integer (number of bytes).

**Note:** When specifying this parameter, ensure that the length of the offset specified is large enough to accommodate the maximum size of your variable data.

#### **VARTYPE**

A required parameter. This parameter defines the type of variable that is used with the set page length in inches tag. Possible values follows.

HIGHLOW	The byte order of the variable is in high-low order. The first byte is the most significant.
LOWHIGH	The byte order of the variable is in low-high order. The last byte is the most significant.
CHRDEC	The variable is in character decimal format with no byte order consideration. All characters are in the range from 0 to 9. Many Hewlett-Packard printers use this variable type.
CHRHEX	The variable is in character hexadecimal format with no byte order consideration. All characters are in the range from 0 to 9, A to F.
CHRAN	The variable is in character alphanumeric format with no byte order consideration. All characters are in the range from 0 to 9, A to Z.

#### **CNVNUM**

A required parameter. This parameter defines the numerator of the conversion ratio. This value must be an integer. The most common value for CNVNUM is 1.

**CNVDEN**

A required parameter. This parameter defines the denominator of the conversion ratio. This value must be a nonzero integer. The most common value for CNVDEN is 1.

**DATA**

A required parameter. This parameter specifies the ASCII control sequence for setting the page length in inches. This must be a hexadecimal value.

*Page Length in Lines (PAGLENL) tag:*

The Page Length in Lines (PAGLENL) tag sets the page length in terms of the number of lines.

The number of lines is carried as a variable in the control sequence for page length. The syntax for this tag follows:

```
:PAGLENL
  VAROFFSET = variable offset in
              control sequence
  VARLEN = variable length
  VARTYPE = HIGHLOW|LOWHIGH|CHRDEC|
            CHRHEX|CHRAN
  DATA = ASCII control sequence.
```

**VAROFFSET**

A required parameter. This parameter defines the offset for the variable portion of the control sequence. This value must be an integer.

**Note:** This offset is relative to the beginning of the control sequence. Therefore, a value of 0 implies that the variable starts in the first byte of the control sequence.

**VARLEN**

A required parameter. This parameter defines the length of the variable portion of the control sequence. This value must be an integer (number of bytes).

**VARTYPE**

A required parameter. This attribute defines the type of variable. Possible values follow.

HIGHLOW	The byte order of the variable is in high-low order. The first byte is the most significant.
LOWHIGH	The byte order of the variable is in low-high order. The last byte is the most significant.
CHRDEC	The variable is in character decimal format with no byte order consideration. All characters are in the range 0 to 9. Many Hewlett-Packard printers use this variable type.
CHRHEX	The variable is in character hexadecimal format with no byte order consideration. All characters are in the range 0 to 9, A to F.
CHRAN	The variable is in character alphanumeric format with no byte order consideration. All characters are in the range 0 to 9, A to Z.

**DATA**

A required parameter. This parameter specifies the ASCII control sequence for setting the page length in lines. This must be a hexadecimal value.

*Printable area:*

You can set the printable area in lines or in inches.

*Text Length in Lines (TEXTLENL) tag:*

The Text Length in Lines (TEXTLENL) tag specifies the text length in lines.

The syntax of this tag follows:

```
:TEXTLENL
  VAROFFSET = variable offset in
              control sequence
  VARLEN    = variable length
  VARTYPE   = HIGHLOW|LOWHIGH|CHRDEC|
              CHRHEX|CHRAN
  DATA     = ASCII control sequence.
```

#### **VAROFFSET**

A required parameter. This parameter specifies the offset for the variable portion of the control sequence. This value must be an integer.

**Note:** This offset is relative to the beginning of the control sequence. Therefore, a value of 0 implies that the variable starts in the first byte of the control sequence.

#### **VARLEN**

A required parameter. This parameter specifies the length of the variable portion of the control sequence. This value must be an integer (number of bytes).

#### **VARTYP**

A required parameter. This parameter specifies the type of variable that is used with the category of printer function tags. Possible values follow.

HIGHLOW	The byte order of the variable is in high-low order. The most significant byte is first.
LOWHIGH	The byte order of the variable is in low-high order. The most significant byte is last.
CHRDEC	The variable is in character decimal format with no byte order consideration. All characters are in the range of 0 to 9. Many Hewlett-Packard ASCII printers use this type of variable.
CHRHEX	The variable is in character hexadecimal format with no byte order consideration. All characters are in the range of A to F.
CHRAN	The variable is in character alphanumeric format with no byte order consideration. All characters are in the range of 0 to 9, A to Z.

#### **DATA**

A required parameter. This parameter specifies the ASCII control sequence for the set-text-length-in-lines printer function. This must be a hexadecimal value.

*Top Margin in Inches (TOPMARGINI) tag:*

The Top Margin in Inches (TOPMARGINI) tag specifies the top margin in inches.

The syntax of this tag follows:

```
:TOPMARGINI
  VAROFFSET = variable offset in
              control sequence
  VARLEN    = variable length
  VARTYPE   = HIGHLOW|LOWHIGH|CHRDEC|
              CHRHEX|CHRAN
  CNVNUM    = conversion ratio
```

CNVDEN =  $\frac{\text{numerator}}{\text{conversion ratio denominator}}$   
 DATA = ASCII control sequence.

**VAROFFSET**

This is a required parameter. This parameter specifies the offset for the variable portion of the control sequence. This value must be an integer.

**Note:** This offset is relative to the beginning of the control sequence. Therefore, a value of 0 implies that the variable starts in the first byte of the control sequence.

**VARLEN**

This is a required parameter. This parameter specifies the length of the variable portion of the control sequence. This value must be an integer (number of bytes).

**VARTYP**

A required parameter. This parameter specifies the type of variable that is used with the category of printer function tags. Possible values follow.

HIGHLOW	The byte order of the variable is in high-low order. The most significant byte is first.
LOWHIGH	The byte order of the variable is in low-high order. The most significant byte is last.
CHRDEC	The variable is in character decimal format with no byte order consideration. All characters are in the range of 0 to 9. Many Hewlett-Packard ASCII printers use this type of variable.
CHRHEX	The variable is in character hexadecimal format with no byte order consideration. All characters are in the range of A to F.
CHRAN	The variable is in character alphanumeric format with no byte order consideration. All characters are in the range of 0 to 9, A to Z.

**CNVNUM**

A required parameter. This parameter specifies the numerator of the conversion ratio. This value must be an integer. The most commonly used value for CNVNUM is 1.

**CNVDEN**

A required parameter. This parameter specifies the denominator of the conversion ratio. This value must be a nonzero integer.

CNVDEN defines the units that are used for the variable portion. For example, if movement is in 1/300 of an inch, CNVNUM should be defined as 1 and CNVDEN should be defined as 300.

**DATA**

A required parameter. This parameter specifies the ASCII control sequence for the set top margin (in inches) printer function. This must be a hexadecimal value.

*Customizing highlighting:*

You can customize highlighting functions, such as underlining and bold printing for an ASCII printer.

*Bold:*

You can use the Start Bold Printing (STRBOLD) and the End Bold Printing (ENDBOLD) tag to define the bold text range.

### **Start Bold Printing (STRBOLD) tag**

The Start Bold Printing (STRBOLD) tag defines the ASCII control sequence for starting the bold printing function for an ASCII printer.

The syntax for this tag follows:

```
:STRBOLD  
    DATA = ASCII control sequence.
```

#### **DATA**

A required parameter. This parameter specifies the ASCII control sequence for the start bold printing function. This must be a hexadecimal value.

### **End Bold Printing (ENDBOLD) tag**

The End Bold Printing (ENDBOLD) tag defines the ASCII control sequence for ending the bold printing function for an ASCII printer.

The syntax for this tag follows:

```
:ENDBOLD  
    DATA = ASCII control sequence.
```

#### **DATA**

A required parameter. This parameter specifies the ASCII control sequence for the end bold printing function. This must be a hexadecimal value.

### *Underscore:*

You can use the Start Underscore Function (STRUS) and the End Underscore (ENDUS) tag to define the underscore text range.

### **Start Underscore Function (STRUS) tag**

The Start Underscore Function (STRUS) tag defines the ASCII control sequence for starting the underscore function for an ASCII printer.

The syntax for this tag follows:

```
:STRUS  
    DATA = ASCII control sequence.
```

#### **DATA**

A required parameter. This parameter specifies the ASCII control sequence for the start underscore function. This must be a hexadecimal value.

### **End Underscore (ENDUS) tag**

The End Underscore (ENDUS) tag defines the ASCII control sequence for ending the underscore function for an ASCII printer.

The syntax for this tag follows:

```
:ENDUS  
    DATA = ASCII control sequence.
```

#### **DATA**

A required parameter. This parameter specifies the ASCII control sequence for the end underscore function. This must be a hexadecimal value.

### *Customizing horizontal movement:*

You can customize these horizontal movement functions.

- Backspace
- Characters per inch
- Characters per inch in computer output reduction (COR) mode
- Horizontal relative movement
- Horizontal absolute movement
- Proportional spacing
- Space

### *Backspace (BSP) tag:*

The Backspace (BSP) tag defines the ASCII control sequence for the backspace function for an ASCII printer.

The syntax for this tag follows:

```
:BSP  
    DATA = ASCII control sequence.
```

#### **DATA**

A required parameter. This parameter specifies the ASCII control sequence for the backspace function. This must be a hexadecimal value.

### *Characters per inch:*

You can set the number of characters per horizontal inch of printing.

You can set the number of characters per inch in normal printing mode or in computer output reduction (COR) mode. COR consists of these items:

- Rotating the paper to a landscape right orientation
- Reducing the character size
- Changing the line density
- Changing the left and top margins

Following is a description of the tags used to customize horizontal character spacing.

### *Characters per Inch in Normal Print Mode (CPI) tag:*

The Characters per Inch in Normal Print Mode (CPI) tag defines the ASCII control sequence for setting the number of characters per inch.

The syntax for this tag follows:

```
:CPI  
    CPI = 5|10|12|133|15|166|  
          171|20|25|27  
    DATA = ASCII control sequence.
```

#### **CPI**

A required parameter. This parameter specifies the number of characters per inch.

- 5**        5 characters per inch
- 10**       10 characters per inch
- 12**       12 characters per inch

- 133 13.3 characters per inch
- 15 15 characters per inch
- 166 16.6 characters per inch
- 171 17.1 characters per inch
- 20 20 characters per inch
- 25 25 characters per inch
- 27 27 characters per inch

#### **DATA**

A required parameter. This parameter specifies the ASCII control sequence for setting the number of characters per inch. This must be a hexadecimal value.

*Characters per Inch in COR Mode (CPICOR) tag:*

The Characters per Inch in COR Mode (CPICOR) tag defines the control sequence for setting the number of characters per inch while in COR mode.

For example, you can define that all print jobs that normally use a 10-pitch type style use a particular 17-pitch type style when COR (reduction) is applied. The syntax for this tag follows:

```
:CPICOR
  CPI = 10|12|15
  ASCIIFNT = font identifier number (integer)
  FNTWTH = font width (integer)
  FNTATR = font attribute (integer)
  DATA = ASCII control sequence.
```

#### **CPI**

A required parameter. This is the current number of characters per inch before the reduction is applied. For example, to define the type style to use for COR in a 10-pitch print job, specify CPI=10.

- 10 10 characters per inch
- 12 12 characters per inch
- 15 15 characters per inch

#### **ASCIIFNT**

A required parameter. This parameter specifies the ASCII type style (the global font ID (FGID)) to be used when reduction is applied. This value must be an integer.

For example, the FGID for a Courier 17-pitch font (17 characters per inch) is 254. Therefore, you specify ASCIIFNT=254 for a Courier 17-pitch font.

#### **FNTWTH**

A required parameter. This parameter specifies the width, in 1/1440-inch increments, of the font used when reduction is applied. This value must be an integer.

For example, if you use a 17-pitch font when reduction is applied, divide the increment by the number of characters per inch (1440 divided by 17). This gives you a font width of 85 (rounded off) for the substituted font. Therefore, you specify FNTWTH=85.

#### **FNTATR**

A required parameter. This parameter specifies the attribute value for the substituted font. This value must be an integer. In most cases, you specify a fixed-pitch type style (FNTATR=1) for the substitute font attribute value.

##### **Integer Attribute value**

- 01 Fixed-pitch font



02 Proportional-spaced font

04 Typographic font

#### DATA

An optional parameter. This parameter specifies the ASCII control sequence to select the substituted font. This must be a hexadecimal value. You find the ASCII control sequence in the technical reference manual for your printer under the description of selecting type styles. If there is no control sequence, "X must be specified. When a control sequence of "X is specified, the values specified for ASCIIFNT, FNTWTH, and FNTATR are used to select the COR font.

*Horizontal Absolute Movement (HORAMOV) tag:*

The Horizontal Absolute Movement (HORAMOV) tag sets the current horizontal print position.

The host print transform requires the HORAMOV tag to be defined for the AFPDS-to-ASCII transform. The syntax of this tag follows:

```
:HORAMOV
  VAROFFSET = variable offset in
              control sequence
  VARLEN    = variable length
  VARTYPE   = HIGHLOW|LOWHIGH|CHRDEC|
              CHRHEX|CHRAN
  CNVNUM    = conversion ratio
              numerator
  CNVDEN    = conversion ratio
              denominator
  DATA     = ASCII control sequence.
```

#### VAROFFSET

A required parameter. This parameter specifies the offset for the variable portion of the control sequence. This value must be an integer.

**Note:** This offset is relative to the beginning of the control sequence. Therefore, a value of 0 implies that the variable starts in the first byte of the control sequence.

#### VARLEN

A required parameter. This parameter specifies the length of the variable portion of the control sequence. This value must be an integer (number of bytes).

#### VARTYP

A required parameter. This parameter specifies the type of variable that is used with the category of printer function tags. Possible values follow.

HIGHLOW	The byte order of the variable is in high-low order. The most significant byte is first.
LOWHIGH	The byte order of the variable is in low-high order. The most significant byte is last.
CHRDEC	The variable is in character decimal format with no byte order consideration. All characters are in the range of 0 to 9. Many Hewlett-Packard ASCII printers use this type of variable.
CHRHEX	The variable is in character hexadecimal format with no byte order consideration. All characters are in the range of A to F.
CHRAN	The variable is in character alphanumeric format with no byte order consideration. All characters are in the range of 0 to 9, A to Z.

#### CNVNUM

A required parameter. This parameter specifies the numerator of the conversion ratio. This value must be an integer. The most commonly used value for CNVNUM is 1.

## CNVDEN

A required parameter. This parameter specifies the denominator of the conversion ratio. This value must be a nonzero integer.

CNVDEN defines the units to be used for the variable portion. For example, if movement is 1/300, CNVNUM should be defined as 1 and CNVDEN should be defined as 300.

## DATA

A required parameter. This parameter specifies the ASCII control sequence for the horizontal absolute movement printer function. This must be a hexadecimal value.

*Horizontal Relative Movement (HORRMOV) tag:*

The Horizontal Relative Movement (HORRMOV) tag specifies the print position backward or forward relative to the current print position.

You might find the horizontal relative movement function referred to as horizontal positioning when looking it up in your printer manual.

You can have up to two HORRMOV tags in the same source file. If you specify two HORRMOV tags in the same source file, one must have a direction of forward and the other a direction of backward. The syntax of this tag follows:

```
:HORRMOV
  DIRECTION = FWD|BCK|FWDBCK
  VAROFFSET = variable offset in
              control sequence
  VARLEN = variable length
  VARTYPE = HIGHLOW|LOWHIGH|CHRDEC|
           CHRHEX|CHRAN
  CNVNUM = conversion ratio
          numerator
  CNVDEN = conversion ratio
          denominator
  DATA = ASCII control sequence.
```

## DIRECTION

A required parameter. This parameter specifies the direction of the relative movement command.

FWD	This parameter specifies the horizontal relative movement forward.
BCK	This parameter specifies the horizontal relative movement backward.
FWDBCK	This parameter specifies the horizontal relative movement command forward and backward.

## VAROFFSET

A required parameter. This parameter specifies the offset for the variable portion of the control sequence. This value must be an integer.

**Note:** This offset is relative to the beginning of the control sequence. Therefore, a value of 0 implies that the variable starts in the first byte of the control sequence.

## VARLEN

A required parameter. This parameter specifies the length of the variable portion of the control sequence. This value must be an integer (number of bytes).

## VARTYPE

A required parameter. This parameter specifies the type of variable used with this category of printer function tags. Possible values follow.

HIGHLOW	The byte order of the variable is in high-low order. The most significant byte is first.
LOWHIGH	The byte order of the variable is in low-high order. The most significant byte is last.
CHRDEC	The variable is in character decimal format with no byte order consideration. All characters are in the range from 0 to 9. Many Hewlett-Packard ASCII printers use this type of variable.
CHRHEX	The variable is in character hexadecimal format with no byte order consideration. All characters are in the range from 0 to 9, A to F.
CHRAN	The variable is in character alphanumeric format with no byte order consideration. All characters are in the range from 0 to 9, A to Z.

#### **CNVNUM**

A required parameter. This parameter defines the numerator of the conversion ratio. This value must be an integer. The most commonly used value for CVNUM is 1.

#### **CNVDEN**

A required parameter. This parameter specifies the denominator of the conversion ratio. This value must be a nonzero integer.

CNVDEN defines the units to be used for the variable portion. For example, if movement is 1/300, CNVNUM should be defined as 1 and CNVDEN should be defined as 300.

#### **DATA**

A required parameter. This parameter specifies the ASCII control sequence for the horizontal relative movement printer function. This must be a hexadecimal value.

*Proportional spacing:*

You can use the Start Proportional Space (STRPROP) and the End Proportional Space (ENDPROP) tag to define the range of text to use proportional spacing.

#### **Start Proportional Space (STRPROP) tag**

The Start Proportional Space (STRPROP) tag defines the ASCII control sequence for starting the proportional space mode for an ASCII printer.

The syntax for this tag follows:

```
:STRPROP
    DATA = ASCII control sequence.
```

#### **DATA**

A required parameter. This parameter specifies the ASCII control sequence for the start proportional space function. This must be a hexadecimal value.

#### **End Proportional Space (ENDPROP) tag**

The End Proportional Space (ENDPROP) tag defines the ASCII control sequence for ending the proportional space mode for an ASCII printer.

The syntax for this tag follows:

```
:ENDPROP
    DATA = ASCII control sequence.
```

#### **DATA**

A required parameter. This parameter specifies the ASCII control sequence for the end proportional space function. This must be a hexadecimal value.

*Space (SPACE) tag:*

The Space (SPACE) tag defines the ASCII control sequence for the space control function for an ASCII printer.

The syntax for this tag follows:

```
:SPACE  
    DATA = ASCII control sequence.
```

**DATA**

A required parameter. This parameter specifies the ASCII control sequence for the space function. This must be a hexadecimal value.

*Customizing vertical movement:*

You can customize these vertical movement functions.

- Form feed
- Half-line feed
- Line feed
- Relative vertical movement
- Absolute vertical movement
- Reverse half-line feed
- Reverse line feed
- Vertical line spacing

*Form Feed (FORMFEED) tag:*

The Form Feed (FORMFEED) tag defines the ASCII control sequence for the form feed function for an ASCII printer.

The FORMFEED control advances the paper to the top of the next page. The syntax for this tag follows:

```
:FORMFEED  
    DATA = ASCII control sequence.
```

**DATA**

A required parameter. This parameter specifies the ASCII control sequence for the form feed function. This must be a hexadecimal value.

*Half-Line Feed (HLFLINEFEED) tag:*

The Half-Line Feed (HLFLINEFEED) tag defines the ASCII control sequence for the half-line feed function for an ASCII printer.

The HLFLINEFEED control advances the paper one half of a line. The syntax for this tag follows:

```
:HLFLINEFEED  
    DATA = ASCII control sequence.
```

**DATA**

A required parameter. This parameter specifies the ASCII control sequence for the half-line feed function. This must be a hexadecimal value.

*Line Feed (LINEFEED) tag:*

The Line Feed (LINEFEED) tag defines the ASCII control sequence for the line feed function for an ASCII printer.

The line feed function advances the paper one line. The syntax for this tag follows:

```
:LINEFEED  
    DATA = ASCII control sequence.
```

#### DATA

A required parameter. This parameter specifies the ASCII control sequence for the line feed function. This must be a hexadecimal value.

*Vertical Absolute Movement (VERAMOV) tag:*

The Vertical Absolute Movement (VERAMOV) tag is used to set the current vertical print position.

The host print transform requires the VERAMOV tag to be defined for the AFPDS-to-ASCII transform. The syntax of this tag follows:

```
:VERAMOV  
    VAROFFSET = variable offset in  
                control sequence  
    VARLEN    = variable length  
    VARTYPE   = HIGHLOW|LOWHIGH|CHRDEC|  
                CHRHEX|CHRAN  
    CNVNUM    = conversion ratio  
                numerator  
    CNVDEN    = conversion ratio  
                denominator  
    DATA     = ASCII control sequence.
```

#### VAROFFSET

A required parameter. This parameter specifies the offset for the variable portion of the control sequence. This value must be an integer.

**Note:** This offset is relative to the beginning of the control sequence. Therefore, a value of 0 implies that the variable starts in the first byte of the control sequence.

#### VARLEN

A required parameter. This parameter specifies the length of the variable portion of the control sequence. This value must be an integer (number of bytes).

#### VARTYP

A required parameter. This parameter specifies the type of variable used with this category of printer function tags. Possible values follow.

HIGHLOW	The byte order of the variable is in high-low order. The most significant byte is first.
LOWHIGH	The byte order of the variable is in low-high order. The most significant byte is last.
CHRDEC	The variable is in character decimal format with no byte order consideration. All characters are in the range of 0 to 9. Many Hewlett-Packard ASCII printers use this type of variable.
CHRHEX	The variable is in character hexadecimal format with no byte order consideration. All characters are in the range of A to F.
CHRAN	The variable is in character alphanumeric format with no byte order consideration. All characters are in the range of 0 to 9, A to Z.

#### CNVNUM

A required parameter. This parameter specifies the numerator of the conversion ratio. This value must be an integer. The most commonly used value for CNVNUM is 1.

**CNVDEN**

A required parameter. This parameter specifies the denominator of the conversion ratio. This value must be a nonzero integer.

CNVDEN defines the units to be used for the variable portion. For example, if movement is 1/300, CNVNUM should be defined as 1 and CNVDEN should be defined as 300.

**DATA**

A required parameter. This parameter specifies the ASCII control sequence for the horizontal absolute movement printer function. This must be a hexadecimal value.

*Vertical Relative Movement (VERRMOV) tag:*

The Vertical Relative Movement (VERRMOV) tag is used to move the print position upward or downward relative to the current print position.

You might find the relative movement function referred to as vertical positioning, when looking it up in your printer manual.

You can have up to two VERRMOV tags in the same source file. If you specify two VERRMOV tags in the same source file, one must have a direction of upward and the other a direction of downward. The syntax of this command follows:

```
:VERRMOV
  DIRECTION = UP|DOWN|UPDOWN
  VAROFFSET = variable offset in
              control sequence
  VARLEN = variable length
  VARTYPE = HIGHLOW|LOWHIGH|CHRDEC|
           CHRHEX|CHRAN
  CNVNUM = conversion ratio
          numerator
  CNVDEN = conversion ratio
          denominator
  DATA = ASCII control sequence.
```

**DIRECTION**

A required parameter. This parameter defines the direction of the relative movement command. Possible values follow.

UP	This parameter defines the vertical relative movement upward.
DOWN	This parameter defines the vertical relative movement downward.
UPDOWN	This parameter defines the vertical relative movement command upward and downward.

**VAROFFSET**

A required parameter. This parameter defines the offset for the variable portion of the control sequence. This value must be an integer.

**Note:** This offset is relative to the beginning of the control sequence. Therefore, a value of zero (0) implies that the variable starts in the first byte of the control sequence.

**VARLEN**

A required parameter. This parameter defines the length of the variable portion of the control sequence. This value must be an integer (number of bytes).

**VARTYPE**

A required parameter. This parameter defines the type of variable used with this category of printer function tags. Possible values follow.

HIGHLOW	The byte order of the variable is in high-low order. The first byte is the most significant.
LOWHIGH	The byte order of the variable is in low-high order. The last byte is the most significant.
CHRDEC	The variable is in character decimal format with no byte order consideration. All characters are in the range from 0 to 9. Many Hewlett-Packard printers use this variable type.
CHRHEX	The variable is in character hexadecimal format with no byte order consideration. All characters are in the range from 0 to 9, A to F.
CHRAN	The variable is in character alphanumeric format with no byte order consideration. All characters are in the range from 0 to 9, A to Z.

#### **CNVNUM**

A required parameter. This parameter defines the numerator of the conversion ratio. This value must be an integer. The most common value is 1.

#### **CNVDEN**

A required parameter. This parameter defines the denominator of the conversion ratio. This value must be a nonzero integer, which defines the units of measure to be used for the variable portion. For example, if movement is 1/300, specify CNVNUM as 1 and CNVDEN as 300.

#### **DATA**

A required parameter. This parameter specifies the ASCII control sequence for the printer function. This must be a hexadecimal value.

#### *Reverse Half-Line Feed (RVSHLFLINEFEED) tag:*

The Reverse Half-Line Feed (RVSHLFLINEFEED) tag defines the ASCII control sequence for the reverse half-line feed function for an ASCII printer.

The reverse half-line feed function moves the paper up one half line. The syntax for this tag follows:

```
:RVSHLFLINEFEED
  DATA = ASCII control sequence.
```

#### **DATA**

A required parameter. This parameter specifies the ASCII control sequence for the reverse half-line feed function. This must be a hexadecimal value.

#### *Reverse Line Feed (RVSLINEFEED) tag:*

The Reverse Line Feed (RVSLINEFEED) tag defines the ASCII control sequence for the reverse line feed function for an ASCII printer.

The reverse line feed function moves the paper up one line. The syntax for this tag follows:

```
:RVSLINEFEED
  DATA = ASCII control sequence.
```

#### **DATA**

A required parameter. This parameter specifies the ASCII control sequence for the reverse line feed function. This must be a hexadecimal value.

#### *Vertical line spacing:*

You can set the distance between printed lines in one of two ways: specifying the number of lines per inch or setting line spacing to a variable value.

These tags are used to customize vertical line spacing.

*Lines per Inch (LPI) tag:*

The Lines per Inch (LPI) tag defines the control sequence for setting the number of lines per inch you want to print.

Variable line spacing, if defined, overrides fixed-pitch line spacing. The syntax for this tag follows:

```
:LPI
    LPI = 3|4|6|8
    DATA = ASCII control sequence.
```

**LPI**

A required parameter. This parameter defines the number of lines per inch. The valid values for this parameter are 3, 4, 6, and 8. The most common vertical spacing is 6 or 8 lines per inch.

**DATA**

A required parameter. This parameter specifies the ASCII control sequence for setting lines per inch. This must be a hexadecimal value.

*Variable Line Spacing (VARLSPC) tag:*

The Variable Line Spacing (VARLSPC) tag is used to set variable line spacing on the printer.

Variable line spacing, if defined, overrides fixed-pitch line spacing. If you do not define variable line spacing, the customizing source is searched for a fixed pitch tag, for example, 8 lines per inch. The syntax for this tag follows:

```
:VARLSPC
    VAROFFSET = variable offset in
                control sequence
    VARLEN = variable length
    VARTYPE = HIGHLOW|LOWHIGH|CHRDEC|
             CHRHEX|CHRAN
    CNVNUM = conversion ratio
            numerator
    CNVDEN = conversion ratio
            denominator
    DATA = ASCII control sequence.
```

**VAROFFSET**

A required parameter. This parameter defines the offset for the variable portion of the control sequence. This value must be an integer.

**Note:** This offset is relative to the beginning of the control sequence. Therefore, a value of 0 implies that the variable starts in the first byte of the control sequence.

**VARLEN**

A required parameter. This parameter defines the length of the variable portion of the control sequence. This value must be an integer (number of bytes).

**VARTYPE**

A required parameter. This parameter defines the type of variable used with the set page length in inches tag. Possible values follow.

HIGHLOW	The byte order of the variable is in high-low order. The first byte is the most significant.
LOWHIGH	The byte order of the variable is in low-high order. The last byte is the most significant.
CHRDEC	The variable is in character decimal format with no byte order consideration. All characters are in the range from 0 to 9. Many Hewlett-Packard printers use this variable type.



CHRHEX	The variable is in character hexadecimal format with no byte order consideration. All characters are in the range from 0 to 9, A to F.
CHRAN	The variable is in character alphanumeric format with no byte order consideration. All characters are in the range from 0 to 9, A to Z.

#### **CNVNUM**

A required parameter. This parameter defines the numerator of the conversion ratio. This value must be an integer. The most common value for CNVNUM is 1.

#### **CNVDEN**

A required parameter. This parameter defines the denominator of the conversion ratio. This value must be a nonzero integer, which defines the units of measure used for the variable portion.

#### **DATA**

A required parameter. This parameter specifies the ASCII control sequence for variable line spacing. This must be a hexadecimal value.

#### *Customizing indexing:*

You can customize superscript and subscript controls for an ASCII printer.

On some printers, these controls move the print position up or down 1/2 line spaces. Other printers create superscripts and subscripts by compressing the height of the characters.

#### *Subscript:*

You can use the Start Subscript Function (STRSUBS) and End Subscript Function (ENDSUBS) tag to define the range of subscripts.

#### **Start Subscript Function (STRSUBS) tag**

The Start Subscript Function (STRSUBS) tag defines the ASCII control sequence for starting the subscript function for an ASCII printer.

The syntax for this tag follows:

```
:STRSUBS
    DATA = ASCII control sequence.
```

#### **DATA**

A required parameter. This parameter specifies the ASCII control sequence for the start subscript function. This must be a hexadecimal value.

#### **End Subscript Function (ENDSUBS) tag**

The End Subscript Function (ENDSUBS) tag defines the ASCII control sequence for ending the subscript function for an ASCII printer.

The syntax for this tag follows:

```
:ENDSUBS
    DATA = ASCII control sequence.
```

#### **DATA**

A required parameter. This parameter specifies the ASCII control sequence for the end subscript function. This must be a hexadecimal value.

### *Superscript:*

You can use the Start Superscript Function (STRSUPS) and the End Superscript Function (ENDSUPS) tag to define the range of the superscripts.

#### **Start Superscript Function (STRSUPS) tag**

The Start Superscript Function (STRSUPS) tag defines the ASCII control sequence for starting the superscript function for an ASCII printer.

The syntax for this tag follows:

```
:STRSUPS  
    DATA = ASCII control sequence.
```

#### **DATA**

A required parameter. This parameter specifies the ASCII control sequence for the start superscript function. This must be a hexadecimal value.

#### **End Superscript Function (ENDSUPS) tag**

The End Superscript Function (ENDSUPS) tag defines the ASCII control sequence for ending the superscript function for an ASCII printer.

The syntax for this tag follows:

```
:ENDSUPS  
    DATA = ASCII control sequence.
```

#### **DATA**

A required parameter. This parameter specifies the ASCII control sequence for the end superscript function. This must be a hexadecimal value.

### *Customizing color:*

You can use the Foreground Color (FOREGRND) tag to customize color.

#### **Foreground Color (FOREGRND) tag**

The Foreground Color (FOREGRND) tag defines the ASCII control sequence for setting different colors when the printer supports color.

The syntax for this tag follows:

```
:FOREGRND  
    COLOR = BLACK|BLUE|RED|PINK|GREEN|  
           CYAN|YELLOW|WHITE|DRKBLUE|  
           ORANGE|PURPLE|DRKGREEN|  
           TURQ|MUSTARD|GREY|BROWN  
    DATA = ASCII control sequence.
```

#### **COLOR**

A required parameter. This parameter defines the color to be used by the printer when it supports different colors for printing. Possible values follow.

BLACK	DRKBLUE (dark blue)	MUSTARD	RED
BLUE	DRKGREEN (dark green)	ORANGE	TURQ (turquoise)
BROWN	GREEN	PINK	WHITE
CYAN	GREY	PURPLE	YELLOW

**DATA**

A required parameter. This parameter specifies the ASCII control sequence for setting the foreground color. This must be a hexadecimal value.

*Customizing paper input:*

You can customize these paper input functions.

- Paper drawer selection
- Selecting paper drawer number
- Additional paper drawer selection

*Paper Drawer Selection (DWRSLT) tag:*

The Paper Drawer Selection (DWRSLT) tag defines the control sequences available for paper drawer selection.

This tag is only used for drawer 1, drawer 2, and the envelope drawer. The syntax for this tag follows:

```
:DWRSLT
    DRAWER = PAPER|ENVELOPE|
            DRAWER1|DRAWER2
    DATA = ASCII control sequence.
```

**DRAWER**

A required parameter. This parameter defines the drawer selection. Possible values follow.

PAPER	The drawer selection is manual paper feed.
ENVELOPE	The drawer selection is envelope.
DRAWER1	The drawer selection is drawer 1.
DRAWER2	The drawer selection is drawer 2.

**DATA**

A required parameter. This parameter specifies the ASCII control sequence for paper drawer selection. This must be a hexadecimal value.

*Selecting Paper Drawer Number (DWRNBR) tag:*

The Selecting Paper Drawer Number (DWRNBR) tag selects the printer paper drawer.

The drawer number is a variable in the ASCII control sequence. This tag is only used if the drawer number is greater than 2. The syntax for this tag follows:

```
:DWRNBR
    VAROFFSET = variable offset in
              control sequence
    VARLEN = variable length
    VARTYPE = HIGHLOW|LOWHIGH|CHRDEC|
            CHRHEX|
    DATA = ASCII control sequence.
```

**VAROFFSET**

A required parameter. This parameter defines the offset for the variable portion of the control sequence. This value must be an integer.

**Note:** This offset is relative to the beginning of the control sequence. Therefore, a value of 0 implies that the variable starts in the first byte of the control sequence.

**VARLEN**

A required parameter. This parameter defines the length of the variable portion of the control sequence. This value must be an integer (number of bytes). If this value is 0, the ASCII sequence is used without a variable.

**VARTYPE**

A required parameter. This attribute defines the type of variable. Possible values are shown in the following table.

HIGHLOW	The byte order of the variable is in high-low order. The first byte is the most significant.
LOWHIGH	The byte order of the variable is in low-high order. The last byte is the most significant.
CHRDEC	The variable is in character decimal format with no byte order consideration. All characters are in the range 0 to 9. Many Hewlett-Packard printers use this variable type.
CHRHEX	The variable is in character hexadecimal format with no byte order consideration. All characters are in the range 0 to 9, A to F.

**DATA**

A required parameter. This parameter specifies the ASCII control sequence for selecting the paper drawer number. This must be a hexadecimal value.

*Additional paper drawer selection:*

The Additional Drawer Selection (ADDDRWTBL) tag is used to select additional paper source drawers.

This tag is only used if the drawer number is greater than 2. The paper drawer selection tag is used to specify drawer 1 and drawer 2.

You begin the support for additional paper drawer selection with the ADDDRWTBL tag and end the support with the EADDDRWTBL tag. The ADDDRWTBLE tags falls between defining each additional drawer.

If you use the additional paper drawer selection tags, your source could look something like this for a printer that has two additional drawers.

```
:ADDDRWTBL.
  :ADDDRWTBLE...
  :ADDDRWTBLE...
:EADDDRWTBL.
```

*Additional Drawer Selection (ADDDRWTBL) tag:*

The Additional Drawer Selection (ADDDRWTBL) tag defines additional drawer entries.

One or more of these tags must follow the ADDDRWTBLE tag containing the additional drawer. There are no parameters on this tag. The syntax follows:

```
:ADDDRWTBL.
```

*Additional Paper Drawer Selection Entry (ADDDRWTBLE) tag:*

The Additional Paper Drawer Selection Entry (ADDDRWTBLE) tag defines additional drawer entries.

The ADDDRWTBL tag must be followed by one or more ADDDRWTBLE tags. A group of ADDDRWTBLE tags must be followed by the EADDDRWTBLE tag. The syntax follows:

```
:ADDRWTABLE
  NUMBER = 3-255
  DATA = ' 'X.
```

**NUMBER**

This is a required parameter. This parameter defines the drawer.

**DATA**

This is a required parameter. This parameter specifies the ASCII control sequence for the drawer specified. This must be a hexadecimal value.

*End Additional Paper Drawer Selection (EADDRWTBL) tag:*

The End Additional Paper Drawer Selection (EADDRWTBL) tag is used to end a group of additional drawer entries in the customization source.

There are no parameters on this tag. The syntax follows:

```
:EADDRWTBL.
```

*Customizing paper output:*

You can customize these paper output functions.

- Output bin selection
- Jog output tray
- Duplex
- Simplex

*Output bin selection:*

You can define support for multiple output bins.

You begin the syntax for output bin selection with the OUTBINTBL tag and end the syntax with the EOUTBINTBL tag. The OUTBINTBLE tags fall between the start and end tags to define each output bin.

If you use this output bin tag, your source could look something like the following syntax for a printer that has multiple output bins.

```
:OUTBINTBL.
  :OUTBINTBLE...
  :OUTBINTBLE...
:EOUTBINTBL.
```

*Output Bin Selection (OUTBINTBL) tag:*

The Output Bin Selection (OUTBINTBL) tag must be followed by one or more OUTBINTBLE tags that define the output bins.

There are no parameters for this tag. The syntax follows:

```
:OUTBINTBL.
```

*Output Bin Selection Entry (OUTBINTBLE) tag:*

The Output Bin Selection Entry (OUTBINTBLE) tag defines a particular output destination.

This tag must be followed by one or more OUTBINTBLE tags. The End Output Bin Selection (EOUTBINTBLE) tag must follow a group of OUTBINTBLE tags. The syntax follows:

```
:OUTBINTBLE
  NUMBER = 1-65535
  DATA = ' 'X.
```

**NUMBER**

This is a required parameter. This parameter defines the output destination.

**DATA**

This is a required parameter. This parameter specifies the ASCII control sequence for the output destination. This must be a hexadecimal value.

*End Output Bin Selection (EOUTBINTBL) tag:*

The End Output Bin Selection (EOUTBINTBL) tag is used to end a group of output bin table entries in the customizing source.

There are no parameters on this tag. The syntax follows:

```
:EOUTBINTBL.
```

*Jog Output Tray (JOGOUTTRAY) tag:*

The Jog Output Tray (JOGOUTTRAY) tag defines the ASCII control sequence for the jog output tray function for an ASCII printer.

The syntax for this tag follows:

```
:JOGOUTTRAY
  DATA = ASCII control sequence.
```

**DATA**

A required parameter. This parameter specifies the ASCII control sequence for the jog output tray function. This must be a hexadecimal value.

*Duplex:*

You can use these tags to configure the duplex printing functions.

*Set Duplex Printing (DUPXPRT) tag:*

The Set Duplex Printing (DUPXPRT) tag defines the ASCII control sequence for the duplex printing function for an ASCII printer.

The DUPXPRT control prints on both sides of a sheet of paper, from left to right. (Contrast the DUPXPRT control with the TUMDUPXPRT control.) The syntax for this tag follows:

```
:DUPXPRT
  DATA = ASCII control sequence.
```

**DATA**

A required parameter. This parameter specifies the ASCII control sequence for the duplex printing function. This must be a hexadecimal value.

*Select Next Side Printing in Duplex (NXTDUPXPRT) tag:*

The Select Next Side Printing in Duplex (NXTDUPXPRT) tag defines the ASCII control sequence for the select next side printing in duplex function for an ASCII printer.

If you do not define this tag, but the host print transform function requests it, a form feed is substituted for the NXTDUPXPRT tag. The syntax for this tag follows:

```
:NXTDUPXPRT
  DATA = ASCII control sequence.
```

**DATA**

A required parameter. This parameter specifies the ASCII control sequence for the select next side printing in duplex function. This must be a hexadecimal value.

*Set Tumble Duplex Printing (TUMDUPXPRT) tag:*

The Set Tumble Duplex Printing (TUMDUPXPRT) tag defines the ASCII control sequence for the set tumble duplex printing function for an ASCII printer.

The TUMDUPXPRT control prints on both sides of a sheet of paper, from top-to-bottom. (Contrast the TUMDUPXPRT control with the DUPXPRT control.) The syntax for this tag follows:

```
:TUMDUPXPRT
    DATA = ASCII control sequence.
```

**DATA**

A required parameter. This parameter specifies the ASCII control sequence for the set tumble duplex printing function. This must be a hexadecimal value.

*Simplex:*

You can use the Set Simplex Printing (SMPXPRT) tag to set the printer to print on one side of the paper.

**Set Simplex Printing (SMPXPRT) tag**

The SMPXPRT (set simplex printing) tag defines the ASCII control sequence for the set simplex printing function for an ASCII printer. The syntax for this tag follows:

```
:SMPXPRT
    DATA = ASCII control sequence.
```

**DATA**

A required parameter. This parameter specifies the ASCII control sequence for the set simplex printing function. This must be a hexadecimal value.

*Customizing print orientation:*

You can customize these print orientation functions.

- Print angle
- Paper orientation

*Print Angle (PRTANGLE) tag:*

The Print Angle (PRTANGLE) tag defines the control sequence for setting the print direction on the page.

This allows printing in all four orientations on the same page. The syntax of this tag follows:

```
:PRTANGLE
    ANGLE    = 0|90|180|270
    DATA    = ASCII control sequence.
```

**ANGLE**

A required parameter. This parameter specifies the angle at which the output prints. This value must be an integer.

**0** The angle of printing is 0 degrees.

**90** The angle of printing is 90 degrees.

**180**

The angle of printing is 180 degrees.

The angle of printing is 270 degrees.

**DATA**

A required parameter. This parameter specifies the ASCII control sequence for the print angle printer function. This must be a hexadecimal value.

*Paper Orientation (PRTORIENT) tag:*

The Paper Orientation (PRTORIENT) tag defines the control sequence for setting different paper orientations.

The syntax for this tag follows:

```
:PRTORIENT
    ORIENT = PORTRAIT|LANDSCAPE|
            RTT180|RTT270
    DATA = ASCII control sequence.
```

**ORIENT**

A required parameter. The orientation in which a job prints. Possible values follow.

PORTRAIT	The print job prints in an orientation rotated 0 degrees.
LANDSCAPE	The print job prints in an orientation rotated 90 degrees.
RTT180	The print job prints in an orientation rotated 180 degrees.
RTT270	The print job prints in an orientation rotated 270 degrees.

**DATA**

A required parameter. This parameter specifies the ASCII control sequence for setting the paper orientation for the printer. This must be a hexadecimal value.

*Customizing print quality:*

You can use the Print Quality (PRTQLTY) tag to define the print quality for your ASCII printer.

**Print Quality (PRTQLTY) tag**

The Print Quality (PRTQLTY) tag defines the control sequence for selecting the level of print quality (such as draft or letter) that an ASCII printer provides.

The syntax for this tag follows:

```
:PRTQLTY
    QLTYTYPE = DRAFT|LETTER|TEXT
    DATA = ASCII control sequence.
```

**QLTYTYPE**

A required parameter. This parameter defines the quality of print. Possible values follow.

DRAFT	The print quality is draft quality. This is equivalent to the *DRAFT type used in the IBM i printer file commands.
LETTER	The print quality is letter quality. This is equivalent to the *NLQ type used in the IBM i printer file commands.
TEXT	The print quality is text quality. This is equivalent to the *STD type used in the IBM i printer file commands.



## DATA

A required parameter. This parameter specifies the ASCII control sequence for the function. This is a hexadecimal value.

### *Customizing fonts:*

You can change the appearance of characters printed by an ASCII printer by selecting and customizing fonts.

You can choose a range of fonts to be used by the printer. You can also customize individual fonts that are used by the printer.

When customizing fonts, consider the following information:

- Individual font definitions are read before group font definitions.
- The first group definition that satisfies the font request is used.
- If you do not specify font-width data, a system-supplied font width is used.
- You do not have to define font widths for fixed-pitch type styles unless the font identifier you specify is outside the normal range of font identifiers for a particular pitch.

For example, the normal range of font identifiers is from 1 to 65 for a 10-pitch type style. If you specify a font identifier that is greater than 65 for a 10-pitch type style, you must also define a font width for that type style.

- If you specify font-width data, the data values must be either 256 or 512 bytes in length. A maximum width of 255/1440ths of an inch per character can fit in a single byte. If the font widths of all characters can fit in a single byte, the data values are 256 bytes in length. If the font width of any one character exceeds 255/1440ths of an inch in width, use a 2-byte value for each character. The data values are then 512 bytes in length.
- To ensure that your print jobs look similar to print jobs generated from the host, you must specify font-width data equal to, or less than, the font-width data used by the host application.
- For fixed-pitch fonts, character per inch (CPI) definitions are read last.
- Type style requests are automatically generated for IBM page printer data stream level 2, Hewlett-Packard PCL4, and Hewlett-Packard PCL5 tables.
- For proportional and typographic fonts, start proportional mode tags and end proportional mode tags are read last.

### *Font groups:*

You can choose font groups used by an ASCII printer.

If you use the font group tags, your source could look something like this for a printer that supports five separate font groups:

```
:FNTGRP.  
    :FNTGRPE....  
    :FNTGRPE....  
    :FNTGRPE....  
    :FNTGRPE....  
    :FNTGRPE....  
:EFNTGRP.
```

### *Font Group (FNTGRP) tag:*

The Font Group (FNTGRP) tag defines the beginning of one or more font group entry (FNTGRPE) tags. It must be followed by one or more FNTGRPE tags.

The syntax for the FNTGRP tag follows:

:FNTGRP.

*Font Group Entry (FNTGRPE) tag:*

The Font Group Entry (FNTGRPE) tag defines a range of fonts.

The FNTGRPE tag must follow either a font group (FNTGRP) tag or another FNTGRPE tag in your source. A group of one or more of these tags must be followed by an End Font Group (EFNTGRP) tag. The syntax for this tag follows:

```
:FNTGRPE
  MINFID = font identifier (integer)
  MAXFID = font identifier (integer)
  FNTSTR = start font ASCII control
           sequence
  FNTEND = end font ASCII control
           sequence
  FNTWTH = character width data of
           the font.
```

#### **MINFID**

A required parameter. This parameter defines the smallest font identifier in a group. This value must be an integer.

#### **MAXFID**

A required parameter. This parameter defines the largest font identifier in a group. This value must be an integer.

#### **FNTSTR**

A required parameter. This parameter defines the ASCII control sequence to start a font request.

#### **FNTEND**

An optional parameter. This parameter defines the ASCII control sequence to end a font request. If no ASCII control sequence is defined to end a font request, "X must be entered.

#### **FNTWTH**

An optional parameter. This parameter specifies the individual character widths in 1/1440-inch increments for the font group range. If specified, this must be either a 256-byte or 512-byte hexadecimal value. If no font-width data is defined, "X must be entered.

#### **Notes:**

1. The ranges specified for MINFID and MAXFID are not validated for different FNTGRPE tags to ensure that they do not overlap.
2. Validation is not performed to ensure that individual fonts (INDFNTE) are not defined within a range of fonts specified in a FNTGRPE tag.

*End Font Group (EFNTGRP) tag:*

The End Font Group (EFNTGRP) tag ends the font group definition for a transform table.

The syntax for this tag follows:

```
:EFNTGRP.
```

*Individual fonts:*

You can customize individual fonts used by an ASCII printer.

If you use the individual font tags, your source could look something like:

```

:INDFNT.
  :INDFNTE....
  :INDFNTE....
  :INDFNTE....
  :INDFNTE....
  :INDFNTE....
:EINDFNT.

```

*Individual Font (INDFNT) tag:*

The Individual Font (INDFNT) tag defines the beginning of one or more INDFNTE tags.

The INDFNT tag must be followed by one or more individual font entry (INDFNTE) tags. The syntax for this tag follows:

```
:INDFNT.
```

*Individual Font Entry (INDFNTE) tag:*

The Individual Font Entry (INDFNTE) tag defines an individual font.

One or more individual font entry tags can be defined, but these tags must follow the INDFNT tag. The INDFNTE tag or tags must be followed by the End Individual Font (EINDFNT) tag. The syntax for this tag follows:

```

:INDFNTE
  FID = font identifier (integer)
  POINTSIZE = font point size
              (integer)
  FNTSTR = start font ASCII control
           sequence
  FNTEND = end font ASCII control
           sequence
  FNTWTH = character width data of
           the font.

```

**FID**

A required parameter. Identifies the individual font. This value must be an integer.

**POINTSIZE**

This parameter specifies the point size of the individual font in 1/72-inch increments. This value must be an integer. If the font point size is not required (for example, if you are defining a fixed-pitch font), 0 must be entered.

**FNTSTR**

A required parameter. This parameter defines the ASCII control sequence for the start of the font.

**FNTEND**

The ASCII control sequence for the end of the individual font. If there is no end font ASCII control sequence, "X must be specified.

**FNTWTH**

This parameter specifies the individual character widths in 1/1440-inch increments for the individual font. This must be specified as a 256-byte or 512-byte hexadecimal value. If there is no font-width data, "X must be specified.

**Notes:**

1. The last data element within this tag entry must be followed by a period.
2. The FID and POINTSIZE parameter pairs are not validated to ensure that they are not duplicated.
3. The FID parameter is not checked to see if it is already defined in a font group (FNTGRPE) tag.

*End Individual Font (EINDFNT) tag:*

The End Individual Font (EINDFNT) tag defines the end of one or more INDFNTE tags.

The syntax for this tag follows:

```
:EINDFNT.
```

*Font mapping:*

Font global identifiers (FGID) can be mapped to IBM i AFP bitmap fonts.

You begin the syntax for font mapping with the FNTMAPTBL tag and end the syntax with the EFNTMAPTBL tag. The FNTMAPE tags fall between the start and end tags to define each FGID that is mapped to an IBM i bitmap font.

If you use the font mapping tags, your source could look like this:

```
:FNTMAPTBL.  
  :FNTMAPE....  
  :FNTMAPE....  
  :FNTMAPE....  
:EFNTMAPTBL.
```

*Font Mapping (FNTMAPTBL) tag:*

The Font Mapping (FNTMAPTBL) tag defines the mapping of Font Global Identifiers (FGID)-to-IBM i AFP bitmap fonts.

This tag must be followed by one or more FNTMAPE tags that contain the font mapping. There are no parameters to this tag. The syntax is:

```
:FNTMAPTBL.
```

*Font Mapping Entry (FNTMAPE) tag:*

The Font Mapping Entry (FNTMAPE) tag defines the font mapping of Font Global Identifiers (FGID)-to-IBM i AFP bitmap fonts.

One or more of the FNTMAPE tags must follow the FNTMAPTBL tag. A group of FNTMAPE tags must be followed by the End Font Mapping (EFNTMAPTBL) tag. The syntax is:

```
:FNTMAPE  
  FGID = Integer (FGID value)  
  POINTSIZE = font point size  
  Name = 8 character set name  
  LIBRARY = 10 character set library name.
```

#### **FGID**

This is a required parameter. This parameter defines the font type global identifier.

#### **POINTSIZE**

This parameter specifies the point size of the individual font in 1/72-inch increments. This value must be an integer. If the font point size is not required (for example, if you are defining a fixed point font), 0 must be entered.

#### **NAME**

This is a required parameter. This parameter specifies the name of the resident bitmap font character set. This is 8 characters.

#### **LIBRARY**

This is a required parameter. This parameter specifies the library where the character set resides. This is 10 characters.

*End Font Mapping (EFNTMAPTBL) tag:*

The End Font Mapping (EFNTMAPTBL) tag is used to end a group of font map table entries in the customization source.

There are no parameters on this tag. The syntax is:

:EFNTMAPTBL.

*Customizing code page support:*

With the workstation customization functions, you can customize code page support for an ASCII printer.

You can perform the following customization:

- Customize EBCDIC-to-ASCII code page mapping
- Support additional ASCII code pages
- Override the default ASCII code page

When you specify code page information in a customization object, the customization object information takes precedence over information in system-supplied code page tables. If you remove code page tags from a customization object, code page information continues to be read from the system-supplied tables.

*Customizing EBCDIC-to-ASCII code page mapping:*

EBCDIC-to-ASCII mapping tables convert an EBCDIC character specified in an application data stream into an ASCII character code value (for that same character).

The EBCDIC-to-ASCII mapping table used for a given ASCII printer depends on the manufacturer, type, and model configured for the printer. Different types of ASCII printers support different ASCII code pages. The code page in use at any time is determined by a command to select the ASCII code page.

You can customize the EBCDIC-to-ASCII mapping that should be used for an ASCII printer. You can customize the mapping to replace the entire ASCII code page mapped to an EBCDIC code page. For example, the U.S. English EBCDIC code page 037 normally maps to the ASCII code page 437. You could map EBCDIC code page 037 to an ASCII code page other than 437 (such as ASCII code page 850).

The following code page mapping tables are available on the operating system for use by the host print transform function. The default mapping table is the mapping table used if you do not override the default table using a customization object.

EBCDIC code page used	Default ASCII code page	Alternate ASCII code page	Alternate ASCII code page	Alternate ASCII code page	Alternate ASCII code page	Alternate ASCII code page	Alternate ASCII code page
037	437	850	860	863	1051		
273	850	437	1051				
277	850	865	1051				
278	850	437	865	1051			
280	850	437	1051				
282	850	860	1051				
284	850	437	1051				
285	850	437	1051				
297	850	437	1051				
420	864	1051					

EBCDIC code page used	Default ASCII code page	Alternate ASCII code page	Alternate ASCII code page	Alternate ASCII code page	Alternate ASCII code page	Alternate ASCII code page	Alternate ASCII code page
423	851	1051					
424	856	862	1051				
500	850	437	860	861	863	865	1051
838	874	1051					
870	852	1051					
871	850	437	861	1051			
875	869	1051					
880	850	1051					
905	857	1051					
1025	850	1051					
1026	857	1051					

In addition, you can map the EBCDIC symbols code page (code page 259) to an ASCII code page. If you configure a printer to support the ASCII symbols code page (code page 899), the complete mapping from EBCDIC code page 259 to ASCII code page 899 is used. Otherwise, a partial mapping is done from EBCDIC code page 259 to the currently defined ASCII code page. The partial mapping can be done from EBCDIC code page 259 to any of the following ASCII code pages:

437	850	851	852
856	857	860	861
862	863	864	865
869	874	899	1051

You can also customize EBCDIC-to-ASCII mapping for an individual code point within an EBCDIC code page. When you customize the mapping for an individual code point, though, you must indicate data values for every code point within the code page. You cannot just indicate the data value for the code point that you want to customize.

You begin the syntax for various mapping tables with an EBCDIC-to-ASCII Mapping Table (EBCASCTBL) tag, and end the syntax with an End EBCDIC-to-ASCII Mapping Table (EEBCASCTBL) tag. The EBCASCTBLE tags fall between the start and end tags to define the EBCDIC-to-ASCII mapping using the DATA parameter.

If you use the EBCDIC-to-ASCII definition tags, your source could look something like this:

```
:EBCASCTBL.
  :EBCASCTBLE...
  .
  .
  :EBCASCTBLE...
  .
  .
:EBCASCTBL.
```

*EBCDIC-to-ASCII Mapping Table (EBCASCTBL) tag:*

You use the EBCDIC-to-ASCII Mapping Table (EBCASCTBL) tag to begin a group of one or more EBCASCTBLE tags.

This tag must be followed by one or more mapping table entry (EBCASCTBLE) tags. There are no parameters on this tag. The syntax for this tag follows:

```
:EBCASCTBL.
```

*EBCDIC-to-ASCII Mapping Table Entry (EBCASCTBLE) tag:*

You specify the EBCDIC-to-ASCII conversion table for an ASCII printer using the DATA parameter of the EBCDIC-to-ASCII Mapping Table Entry (EBCASCTBLE) tag.

The EBCASCTBLE tag must follow an EBCASCTBL tag. The syntax for this tag is:

```
:EBCASCTBLE
  EBCDICCP = EBCDIC code page
              identifier (integer)
  ASCIIICP = ASCII code page
              identifier (integer)
  DATA = EBCDIC to ASCII table data.
```

#### **EBCDICCP**

A required parameter. This parameter specifies the EBCDIC code page identifier.

*code page ID*

A registered identifier used to specify a particular assignment of code points to graphic characters.

#### **ASCIIICP**

A required parameter. This parameter specifies the ASCII code page identifier.

*code page ID*

A registered identifier used to specify a particular assignment of code points to graphic characters.

#### **DATA**

A required parameter. This parameter specifies the hexadecimal data that is used to map EBCDIC codes from the IBM i system to the ASCII codes needed by the ASCII printer.

When you customize to replace the entire ASCII code page mapped to an EBCDIC code page, you set the DATA value to "X (null). For example, EBCDIC code page 277 maps to ASCII code page 865 by default. If you prefer to map EBCDIC code page 277 to ASCII code page 1051 (Roman 8), enter the following code in your customization source:

```
:EBCASCTBLE
  EBCDICCP = 277
  ASCIIICP = 1051
  DATA = 'X.
```

When you customize for an individual code point within an ASCII code page, you must indicate data values for every code point within the code page. You cannot just indicate the data value for the code point that you want to customize. The data must be hexadecimal, and exactly 192 bytes in length. For example, EBCDIC code page 277 maps to ASCII code page 865 by default. You like this EBCDIC-to-ASCII code page mapping, except for the mapping of 1 character. To change the code page mapping for that character, you must indicate data values for every code point within the code page, not just for the code point you want changed.

*End EBCDIC-to-ASCII Mapping Table (EEBCASCTBL) tag:*

You use the End EBCDIC-to-ASCII Mapping Table (EEBCASCTBL) tag to end the EBCDIC-to-ASCII mapping customization.

The syntax for this tag follows:

```
:EEBCASCTBL.
```

### *Additional ASCII code pages:*

The syntax for each ASCII code page begins with an ASCII Code Page Information (ASCCPINFO) tag and ends with an End ASCII Code Page Information (EASCCPINFO) tag.

The following tags fall between the start and end tags to fully define the support of additional code pages:

- CODEPAGE
- ASCIICTL

If you use the ASCII code page tags, your source could look something like this for a printer that supports five separate code pages:

```
:ASCCPINFO.  
  :CODEPAGE....  
  :ASCIICTL....  
  :ASCIICTL....  
  :CODEPAGE....  
  :ASCIICTL....  
  :CODEPAGE....  
  :CODEPAGE....  
  :CODEPAGE....  
  :ASCIICTL....  
  :ASCIICTL....  
  :ASCIICTL....  
:EASCCPINFO.
```

### *ASCII Code Page Information (ASCCPINFO) tag:*

The ASCII Code Page Information (ASCCPINFO) tag defines the beginning of a group of different ASCII code points within a specified ASCII code page.

The ASCCPINFO tag must immediately precede a CODEPAGE tag in your source. The syntax for this tag follows:

```
:ASCCPINFO.
```

### *Code Page (CODEPAGE) tag:*

The Code Page (CODEPAGE) tag defines the ASCII control sequence for selecting the ASCII code page.

The CODEPAGE tag must follow the ASCCPINFO tag, another CODEPAGE tag, or an ASCIICTL tag in your source. You can use more than one CODEPAGE tag, but these tags must come between the ASCCPINFO and EASCCPINFO tags. The syntax for this tag follows:

```
:CODEPAGE  
  CODEPAGE = ASCII code page (integer)  
  DATA = ASCII control sequence.
```

#### **CODEPAGE**

A required parameter. This parameter specifies the identifier (ID) of the ASCII code page to be selected. Any integer value is accepted.

#### **DATA**

A required parameter. This parameter specifies the ASCII control sequence for selecting the ASCII code page on the ASCII printer. This must be a hexadecimal value. The ASCII control sequence for selecting an ASCII code page is sent to the printer when that code page is requested in the data stream.



*ASCII Control Code Mapping (ASCICTL) tag:*

The ASCII Control Code Mapping (ASCICTL) tag defines the ASCII control sequence for an ASCII control code.

The ASCICTL tag must immediately follow the CODEPAGE tag or another ASCICTL tag in your source. You can use more than one ASCICTL tag, but these tags must follow a CODEPAGE tag. The syntax for this tag follows:

```
:ASCICTL
    ASCII = control code
    DATA = ASCII control sequence.
```

**ASCII**

A required parameter. This parameter specifies an ASCII control code. This must be a hexadecimal value from '01'X to 'FF'X.

**DATA**

A required parameter. This parameter specifies the ASCII control sequence that is substituted for the ASCII control code you want to map. This must be a hexadecimal value.

By specifying the ASCICTL tag, you can convert a single-byte control code to a multiple-byte control sequence. For example, you could use this code to have the printer print characters that are not normally in the range of printable characters (such as a heart or a diamond shape). You could also use this tag to allow the printer to change to a different code page to print a single character. Then you can have the printer change back to the original code page.

*End ASCII Code Page Information (EASCCPINFO) tag:*

The End ASCII Code Page Information (EASCCPINFO) tag ends a group of CODEPAGE and ASCICTL tags that define ASCII code page support for an ASCII printer.

This tag must come after an ASCCPINFO tag, and immediately following either a CODEPAGE tag or an ASCICTL tag in your source. The syntax for this tag is:

```
:EASCCPINFO.
```

*Default ASCII Code Page (DFTASCCP) Tag:*

You can use the Default ASCII Code Page (DFTASCCP) tag to override the default ASCII code page for all EBCDIC code page values.

You might use this tag to replace an IBM-defined code page with the code page defined by another manufacturer for a specific printer. For example, Hewlett-Packard printers have code pages defined that differ from the code pages defined by IBM. With the DFTASCCP tag, you can support a Hewlett-Packard defined code page. The DFTASCCP tag is used only when the system searches the system-supplied tables. The syntax for this tag follows:

```
:DFTASCCP
    ASCIIICP = default ASCII code page
                identifier (integer).
```

**ASCIIICP**

A required parameter. This parameter specifies the ASCII code page identifier that should be used in place of the recommended ASCII code page for all EBCDIC code page values. The identifier must be an integer. The code page identifier is not validated to ensure that you have entered a valid code page.

*Customizing fax support:*

Facsimile Support/400 with a modem connection is supported by host print transform customization.

The tags can only be used when the printer data stream values are these:

- IOCA\_G3MH
- IOCA\_G3MRK2
- IOCA\_G3MRK4

### **Fax size**

You can customize printing for various fax sizes. You begin support for various fax sizes with a Fax Size for Transform (FAXSIZXFM) tag, and end the support with an End Fax Size for Transform (EFAXSIZXFM) tag. The Fax Size Entry (FAXSIZE) tags fall between the start and end tags to define each different fax size.

If you use the fax size tags, your source could look something like this for two different fax sizes:

```
:FAXSIZXFM.  
    :FAXSIZE....  
    :FAXSIZE....  
:EFAXSIZXFM.
```

*Fax Size for Transform (FAXSIZXFM) tag:*

The Fax Size for Transform (FAXSIZXFM) tag defines the image output size for each drawer number.

This tag must be followed by one or more fax size entry (ENVSIZ) tags that contain the fax sizes. The syntax for the FAXSIZXFM tag follows:

```
:FAXSIZXFM.
```

*Fax Size Entry (FAXSIZE) tag:*

The Fax Size Entry (FAXSIZE) tag defines the image output size for a particular drawer number.

One or more of these tags follows an FAXSIZXFM tag. An EFAXSIZXFM tag must follow a group of one or more of these tags. The syntax for the ENVSIZ tag follows:

```
:FAXSIZE  
    DRAWER = 0-255  
    IMGWTH = image width in pels  
    IMGLEN = image length in pels.
```

### **DRAWER**

A required parameter. This parameter defines the drawer number for the output size.

**0** The envelope drawer is specified.

**1-255** The drawer 1-255 is specified.

### **IMGWTH**

A required parameter. This parameter specifies the output image width in pels. This value must be an integer.

### **IMGLEN**

A required parameter. This parameter specifies the output image in length pels. This value must be an integer.

*End Fax Size for Transform (EFAXSIZXFM) tag:*

The End Fax Size for Transform (EFAXSIZXFM) tag ends a group of fax size entries.

The syntax for this tag follows:

:EFAXSIZXFM.

*Customizing Double-Byte Character Set (DBCS) support:*

You can customize DBCS support with these tags.

**Related information:**

“Using the tags” on page 144

Here are the descriptions of the valid printer functions and corresponding tags for a device class of TRANSFORM. These tags must follow the transform table tag (TRNSFRMTBL) in your source.

*CCSID mapping:*

You can customize the mapping of EBCDIC-to-ASCII CCSIDs for double-byte character conversion.

You begin the support for the EBCDIC-to-ASCII CCSID mapping with the EBCDIC-to-ASCII CSSID mapping (EBCASCCSID) tag and end the support with the End EBCDIC-to-ASCII CSSID mapping table entry (EEBCASCCSID) tag. The EBCASCCSIDE tags fall between defining each CCSID mapping.

If you use the EBCDIC-to-ASCII CCSID mapping tags, your source could look something like this for a printer supporting DBCS.

```
:EBCASCCSID.  
  :EEBACASCCSIDE....  
  :EEBACASCCSIDE....  
:EEBCASCCSID.
```

If no tag is specified, the CCSID mapping is made according to the following table.

*Table 6. Default CCSID mapping table*

From CCSID	Default CCSID	Language
5026	932	Japanese
5035	932	Japanese
930	932	Japanese
931	932	Japanese
939	932	Japanese
933	949	Korean
937	950	Traditional Chinese
935	1381	Simplified Chinese

*EBCDIC-to-ASCII CSSID mapping (EBCASCCSID) tag:*

The EBCASCCSID tag is used to begin a group of one or more EBCDIC-to-ASCII CSSID mapping entry (EBCASCCSIDE) tags.

One or more CCSID mapping entries must follow this tag. There are no parameters for this tag. The syntax follows:

:EBASCCSID.

*EBCDIC-to-ASCII CSSID mapping entry (EBCASCCSIDE) tag:*

The EBCACCSIDE tag defines the mapping of double-byte EBCDIC CCSIDs to their ASCII CCSID.

The EBCACCSIDE tags must follow an EBCASCCSID tag. The syntax is:

```
:EBCACCSIDE  
  EBCDICCSID = EBCDIC CCSID (integer)  
  ASCII CCSID = ASCII CCSID (integer).
```

**EBCDICCSID**

This is a required parameter. This parameter defines the EBCDIC CCSID identifier. The CCSID is a registered ASCII identifier that is used to specify the CCSID of the source characters.

**ASCII CCSID**

This is a required parameter. This parameter defines the ASCII CCSID identifier. The CCSID is a registered ASCII identifier that is used to specify the CCSID of the target characters.

*End EBCDIC-to-ASCII CSSID mapping table entry (EEBCASCCSID) tag:*

The EEBCASCCSID tag is used to end the EBCDIC-to-ASCII CCSID mapping customization.

The syntax is:

```
:EEBCASCCSID.
```

*Shift In (SI) tag:*

The Shift In (SI) tag defines the ASCII control sequence to select the single-byte mode for an ASCII printer.

One byte of data represents printable characters. The syntax follows:

```
:SI  
  DATA = ASCII control sequence.
```

**DATA**

A required parameter. This parameter specifies the ASCII control sequence for the printer function. This must be a hexadecimal value.

*Shift Out (SO) tag:*

The Shift Out (SO) tag defines the ASCII control sequence to select the double-byte mode for an ASCII printer.

The syntax follows:

```
:SO  
  DATA = ASCII control sequence.
```

**DATA**

A required parameter. This parameter specifies the ASCII control sequence for the printer function. This must be a hexadecimal value.

*DBCS Space (DBSPACE) tag:*

The DBCS Space (DBSPACE) tag defines the ASCII control sequence for the double-byte space control function for an ASCII printer.

The syntax follows:

```
:DBSPACE  
  DATA = ASCII control sequence.
```

**DATA**

A required parameter. This parameter specifies the ASCII control sequence for the printer function. This must be a hexadecimal value.

*Character Orientation (CHRORIENT) tag:*

The Character Orientation (CHRORIENT) tag defines the control sequence for setting different character orientations.

The syntax follows:

```
:CHRORIENT
  ORIENT = PORTRAIT|LANDSCAPE|RTT180|RTT270
  DATA = ASCII control sequence.
```

**ORIENT**

A required parameter. This parameter specifies the orientation that a character prints.

PORTRAIT	Characters print in an orientation rotated 0 degrees.
LANDSCAPE	Characters print in an orientation rotated 90 degrees.
RTT180	Characters print in an orientation rotated at 180 degrees.
RTT270	Characters print in an orientation rotated at 270 degrees.

**DATA**

A required parameter. This parameter specifies the ASCII control sequence for the printer function. This must be a hexadecimal value.

*Set Character Pitch (SCPITCH) tag:*

The Set Character Pitch (SCPITCH) tag defines the control sequence for setting the number of characters per inch.

The syntax follows:

```
:SCPITCH
  VAROFFSET = variable offset in control sequence
  VARLEN = variable length
  VARTYPE = HIGHLOW|LOWHIGH|CHRDEC|CHRHEX|CHRAN
  CNVNUM = conversion ratio numerator
  CNVDEN = conversion ratio denominator
  DATA = ASCII control sequence.
```

**VAROFFSET**

This is a required parameter. This parameter specifies the offset for the variable portion of the control sequence. This value must be an integer.

**Note:** This offset is relative to the beginning of the control sequence. Therefore, a value of 0 implies that the variable starts in the first byte of the control sequence.

**VARLEN**

This is a required parameter. This parameter specifies the length of the variable portion of the control sequence. This value must be an integer (number of bytes).

**VARTYP**

A required parameter. This parameter specifies the type of variable that is used with the category of printer function tags.

HIGHLOW	The byte order of the variable is in high-low order. The most significant byte is first.
---------	--

LOWHIGH	The byte order of the variable is in low-high order. The most significant byte is last.
CHRDEC	The variable is in character decimal format with no byte order consideration. All characters are in the range of 0 to 9. Many Hewlett-Packard ASCII printers use this type of variable.
CHRHEX	The variable is in character hexadecimal format with no byte order consideration. All characters are in the range of A to F.
CHRAN	The variable is in character alphanumeric format with no byte order consideration. All characters are in the range of 0 to 9, A to Z.

#### **CNVNUM**

A required parameter. This parameter specifies the numerator of the conversion ratio. This value must be an integer. The most commonly used value for CNVNUM is 1.

#### **CNVDEN**

A required parameter. This parameter specifies the denominator of the conversion ratio. This value must be a nonzero integer.

CNVDEN defines the units that are used for the variable portion. For example, if movement is in 1/300ths, CNVNUM should be defined as 1 and CNVDEN should be defined as 300.

#### **DATA**

A required parameter. This parameter specifies the ASCII control sequence for the set top margin (inches) printer function. This must be a hexadecimal value.

*Set Line Pitch (SLPITCH) tag:*

The Set Line Pitch (SLPITCH) tag defines the control sequence for setting the number of lines per inch.

The syntax follows:

```
:SCPITCH
  VAROFFSET = variable offset in control sequence
  VARLEN = variable length
  VARTYPE = HIGHLOW|LOWHIGH|CHRDEC|CHRHEX|CHRAN
  CNVNUM = conversion ratio numerator
  CNVDEN = conversion ratio denominator
  DATA = ASCII control sequence.
```

#### **VAROFFSET**

This is a required parameter. This parameter specifies the offset for the variable portion of the control sequence. This value must be an integer.

**Note:** This offset is relative to the beginning of the control sequence. Therefore, a value of 0 implies that the variable starts in the first byte of the control sequence.

#### **VARLEN**

This is a required parameter. This parameter specifies the length of the variable portion of the control sequence. This value must be an integer (number of bytes).

#### **VARTYP**

A required parameter. This parameter specifies the type of variable that is used with the category of printer function tags.

HIGHLOW	The byte order of the variable is in high-low order. The most significant byte is first.
LOWHIGH	The byte order of the variable is in low-high order. The most significant byte is last.

CHRDEC	The variable is in character decimal format with no byte order consideration. All characters are in the range of 0 to 9. Many Hewlett-Packard ASCII printers use this type of variable.
CHRHEX	The variable is in character hexadecimal format with no byte order consideration. All characters are in the range of A to F.
CHRAN	The variable is in character alphanumeric format with no byte order consideration. All characters are in the range of 0 to 9, A to Z.

**CNVNUM**

A required parameter. This parameter specifies the numerator of the conversion ratio. This value must be an integer. The most commonly used value for CNVNUM is 1.

**CNVDEN**

A required parameter. This parameter specifies the denominator of the conversion ratio. This value must be a nonzero integer.

CNVDEN defines the units that are used for the variable portion. For example, if movement is in 1/300ths, CNVNUM should be defined as 1 and CNVDEN should be defined as 300.

**DATA**

A required parameter. This parameter specifies the ASCII control sequence for the set top margin (inches) printer function. This must be a hexadecimal value.

*Set Font Size Scaling (FONTSCALING) tag:*

The Set Font Size Scaling (FONTSCALING) tag defines the control sequence for setting the font size scaling.

The syntax follows:

```
:FONTSCALING
  VAROFFSET = variable offset in control sequence
  VARLEN = variable length
  VARTYPE = variable type
  CNVNUM = conversion ratio numerator
  CNVDEV = conversion ratio denominator
  DATA = ASCII control sequence.
```

**VAROFFSET**

This is a required parameter. This parameter specifies the offset for the variable portion of the control sequence. This value must be an integer.

**Note:** This offset is relative to the beginning of the control sequence. Therefore, a value of 0 implies that the variable starts in the first byte of the control sequence.

**VARLEN**

This is a required parameter. This parameter specifies the length of the variable portion of the control sequence. This value must be an integer (number of bytes).

**VARTYP**

A required parameter. This parameter specifies the type of variable that is used with the category of printer function tags.

HIGHLOW	The byte order of the variable is in high-low order. The most significant byte is first.
LOWHIGH	The byte order of the variable is in low-high order. The most significant byte is last.

CHRDEC	The variable is in character decimal format with no byte order consideration. All characters are in the range of 0 to 9. Many Hewlett-Packard ASCII printers use this type of variable.
CHRHEX	The variable is in character hexadecimal format with no byte order consideration. All characters are in the range of A to F.
CHRAN	The variable is in character alphanumeric format with no byte order consideration. All characters are in the range of 0 to 9, A to Z.

**CNVNUM**

A required parameter. This parameter specifies the numerator of the conversion ratio. This value must be an integer. The most commonly used value for CNVNUM is 1.

**CNVDEN**

A required parameter. This parameter specifies the denominator of the conversion ratio. This value must be a nonzero integer.

CNVDEN defines the units that are used for the variable portion. For example, if movement is in 1/300ths, CNVNUM should be defined as 1 and CNVDEN should be defined as 300.

**DATA**

A required parameter. This parameter specifies the ASCII control sequence for the set top margin (inches) printer function. This must be a hexadecimal value.

*Set Font Size Scale (FONTSCALE) tag:*

The Set Font Size Scale (FONTSCALE) tag defines the control sequence for setting the font size scale.

The syntax follows:

```
:FONTSCALE
SCALE = 1VX1H | 2VX1H | 1VX2H | 2VX2H
DATA = ASCII control sequence.
```

**SCALE**

A required parameter. This parameter specifies the font size scaling.

1VX1H	Normal vertical, normal horizontal
2VX1H	Double vertical, normal horizontal
1VX2H	Normal vertical, double horizontal
2VX2H	Double vertical, double horizontal

**DATA**

A required parameter. This parameter specifies the ASCII control sequence for the printer function. This must be a hexadecimal value.

*Set Characters per Inch (CPI) tag:*

The Set Characters per Inch (CPI) tag defines the control sequence for setting the number of character per inch.

The syntax follows:

```
:CPI
CPI = 6|67|75|18|
DATA = ASCII control sequence.
```

**CPI**

A required parameter. This parameter specifies the number of characters per inch.



- 6        6 characters per inch
- 67      6.7 characters per inch
- 75      7.5 characters per inch
- 18      18 characters per inch

**DATA**

A required parameter. This parameter specifies the ASCII control sequence for setting the number of characters per inch. This must be a hexadecimal value.

*Set Grid Type (GLTYPE) tag:*

The Set Grid Type (GLTYPE) tag defines the control sequence for setting the grid line type.

The syntax follows:

```
:GLTYPE
VAROFFSET = variable offset in control sequence
VARLEN = variable length
VARTYPE = variable type
DATA = ASCII control sequence.
```

**VAROFFSET**

This is a required parameter. This parameter specifies the offset for the variable portion of the control sequence. This value must be an integer.

**Note:** This offset is relative to the beginning of the control sequence. Therefore, a value of 0 implies that the variable starts in the first byte of the control sequence.

**VARLEN**

This is a required parameter. This parameter specifies the length of the variable portion of the control sequence. This value must be an integer (number of bytes).

**VARTYP**

A required parameter. This parameter specifies the type of variable that is used with the category of printer function tags.

HIGHLOW	The byte order of the variable is in high-low order. The most significant byte is first.
LOWHIGH	The byte order of the variable is in low-high order. The most significant byte is last.
CHRDEC	The variable is in character decimal format with no byte order consideration. All characters are in the range of 0 to 9. Many Hewlett-Packard ASCII printers use this type of variable.
CHRHEX	The variable is in character hexadecimal format with no byte order consideration. All characters are in the range of A to F.
CHRAN	The variable is in character alphanumeric format with no byte order consideration. All characters are in the range of 0 to 9, A to Z.

**DATA**

A required parameter. This parameter specifies the ASCII control sequence for the set top margin (inches) printer function. This must be a hexadecimal value.

*Set Grid Line Width (GLWIDTH) tag:*

The Set Grid Line Width (GLWIDTH) tag defines the control sequence for setting the grid line width.

The syntax follows:

:GLWIDTH  
 VAROFFSET = variable offset in control sequence  
 VARLEN = variable length  
 VARTYPE = variable type  
 DATA = ASCII control sequence.

**VAROFFSET**

This is a required parameter. This parameter specifies the offset for the variable portion of the control sequence. This value must be an integer.

**Note:** This offset is relative to the beginning of the control sequence. Therefore, a value of 0 implies that the variable starts in the first byte of the control sequence.

**VARLEN**

This is a required parameter. This parameter specifies the length of the variable portion of the control sequence. This value must be an integer (number of bytes).

**VARTYP**

A required parameter. This parameter specifies the type of variable that is used with the category of printer function tags.

HIGHLOW	The byte order of the variable is in high-low order. The most significant byte is first.
LOWHIGH	The byte order of the variable is in low-high order. The most significant byte is last.
CHRDEC	The variable is in character decimal format with no byte order consideration. All characters are in the range of 0 to 9. Many Hewlett-Packard ASCII printers use this type of variable.
CHRHEX	The variable is in character hexadecimal format with no byte order consideration. All characters are in the range of A to F.
CHRAN	The variable is in character alphanumeric format with no byte order consideration. All characters are in the range of 0 to 9, A to Z.

**DATA**

A required parameter. This parameter specifies the ASCII control sequence for the set top margin (inches) printer function. This must be a hexadecimal value.

*Draw Grid Line (DRAWLINE) tag:*

The Draw Grid Line (DRAWLINE) tag defines the control sequence for the draw grid line function.

The syntax follows:

:DRAWLINE  
 VAROFFSET = variable offset in control sequence  
 VARLEN = variable length  
 VARTYPE = variable type  
 CNVNUM = conversion ratio numerator  
 CNVNUM = conversion ration denominator  
 DATA = ASCII control sequence.

**VAROFFSET**

This is a required parameter. This parameter specifies the offset for the variable portion of the control sequence. This value must be an integer.

**Note:** This offset is relative to the beginning of the control sequence. Therefore, a value of 0 implies that the variable starts in the first byte of the control sequence.

**VARLEN**

This is a required parameter. This parameter specifies the length of the variable portion of the control sequence. This value must be an integer (number of bytes).

**VARTYP**

A required parameter. This parameter specifies the type of variable that is used with the category of printer function tags.

HIGHLOW	The byte order of the variable is in high-low order. The most significant byte is first.
LOWHIGH	The byte order of the variable is in low-high order. The most significant byte is last.
CHRDEC	The variable is in character decimal format with no byte order consideration. All characters are in the range of 0 to 9. Many Hewlett-Packard ASCII printers use this type of variable.
CHRHEX	The variable is in character hexadecimal format with no byte order consideration. All characters are in the range of A to F.
CHRAN	The variable is in character alphanumeric format with no byte order consideration. All characters are in the range of 0 to 9, A to Z.

**CNVNUM**

A required parameter. This parameter specifies the numerator of the conversion ratio. This value must be an integer. The most commonly used value for CNVNUM is 1.

**CNVDEN**

A required parameter. This parameter specifies the denominator of the conversion ratio. This value must be a nonzero integer.

CNVDEN defines the units that are used for the variable portion. For example, if movement is in 1/300ths, CNVNUM should be defined as 1 and CNVDEN should be defined as 300.

**DATA**

A required parameter. This parameter specifies the ASCII control sequence for the set top margin (inches) printer function. This must be a hexadecimal value.

*Printer Data Stream (PRTDTASTRM) tag:*

The Printer Data Stream (PRTDTASTRM) tag identifies the printer data stream that is supported by the ASCII printer.

The following additional values support DBCS printing. The syntax follows:

```
:PRTDTASTRM
  DATASTREAM = IBMNONPAGES|IBMPAGES|ESC/P|LIPS2+|LIPS3|201PL.
```

Value	Description	Raster Mode Supported
ESC/P	The Epson DBCS ESC/P data stream is supported.	No
IBMNONPAGES	The IBM DBCS Nonpages (dot matrix printers) data stream is supported.	No
IBMPAGES	The IBM DBCS Pages data stream is supported.	Yes
LIPS2+	The Cannon DBCS LIPS2+ data stream is supported.	No
LIPS3	The Cannon DBCS LIPS3 data stream is supported.	No
201PL	The NEC DBCS data stream is supported.	No

*Customizing file extension:*

You can use the File Extension (FILEEXT) tag to define the file extension of system-generated file names.

### **File Extension (FILEEXT) tag**

The File Extension (FILEEXT) tag identifies the file extension to be used for system-generated file names when the output data is directed to an integrated file system file. This is a sequence of hexadecimal values that are interpreted as UTF-8 characters and concatenated to the end of the file name. It is limited to 16 bytes in length.

The syntax follows:

```
:FILEEXT  
    DATA = file extension in hexadecimal
```

#### **DATA**

This is a required parameter. It specifies the hexadecimal value for the file extension.

If this tag is not specified, a file extension of .prn is used for system-generated file names.

### **Example**

The following tag causes any system-generated output file names to end with .tif.

```
:FILEEXT  
    DATA = '746966'X.
```

### **Customizing WSCST object of \*CTXFORM transform type:**

You can modify the way the PDF transform maps fonts by customizing your WSCST object of \*CTXFORM transform type.

As with the previous WSCST object classes, you can retrieve the source from shipped WSCST objects using the **Retrieve WSCST source (RTVWSCST)** command, and generate customized versions with the **Create WSCST (CRTWSCST)** command. Because of the need to support open character strings for many parameters, the source physical file used for WSCST source of this class is encoded in UTF-8 (CCSID 1208). Attempts to use source physical files of a different encoding with either command will fail, and message CPF5D33 or CPF5D34 will be issued.

*WSCST Device Class (WSCST ) tag*

WSCST objects of this device class are identified by specifying the value CTXFORM in the WSCST tag. This must be the first tag specified in the WSCST source member. The syntax is:

```
:WSCST DEVCLASS = CTXFORM.
```

This value indicates that the WSCST object defines configuration information for a transform from the IBM Transform Services for i licensed program (5770-TS1). This product and the appropriate product option must be installed to use WSCST objects of this type.

*CTT Transform (CTXFRMTBL) tag:*

The CTXFRMTBL tag defines the beginning of the customization table for WSCST objects of this device class.

The syntax is:

```
:CTXFRMTBL.
```

*Output Data Format (OUTDTAFMT) tag:*

The OUTDTAFMT tag defines the desired data format to be generated by the transform.

The syntax for this tag follows:

```
: OUTDTAFMT
    FORMAT = PDF.
```

#### **FORMAT**

This is a required parameter. The FORMAT tag specifies the format of the output data.

Support for the specified output format requires the installation of at least one option of the IBM Transform Services for i5/OS licensed program. Refer to the following table to determine which option is required for a specified output format.

*Table 7. Required option for specified output format*

Specified output format	Required option
PDF	Option1: AFP to PDF Transform

*File Extension (FILEEXT) Tag:*

The FILEEXT tag identifies the file extension to be used for system-generated file names when the output data is directed to an integrated file system file.

The syntax is:

```
:FILEEXT
    STRING = 'file extension'.
```

#### **STRING**

This is a required parameter. It is a null-terminated character string that is concatenated to the end of the file name. It is limited to 3 characters in length. The characters specified in this parameter must be limited to those in the ISO-8859-1 standard encoding.

If this tag is not specified, a file extension of .prn is used for system-generated file names.

#### **Example**

The following tag causes any system-generated output file names to end with .pdf.

```
:FILEEXT
    STRING = 'pdf'.
```

*Font Mapping by FGID:*

Font global identifiers (FGID) can be mapped to font references recognized by the output data format.

You begin the syntax for font mapping with the Font Mapping by FGID (FGIDMAPTBL) tag and end the syntax with the End Font Mapping (EFGIDMAPTBL) tag. The Font Mapping Entry (FGIDMAPE) tags in between define mappings of a specified FGID to a font reference. Using the font mapping tags, your source could look like the following example.

```
:FGIDMAPTBL.
    :FGIDMAPE....
    :FGIDMAPE....
    :FGIDMAPE....
:EFGIDMAPTBL.
```

*Font Mapping by FGID (FGIDMAPTBL) tag:*

The FGIDMAPTBL tag defines the mapping of font Global Identifiers (FGID) to font resources to be used in the output document.

This tag must be followed by one or more FGIDMAPE tags that contain the font mapping. There are no parameters to this tag. The syntax follows:

:FGIDMAPTBL.

*Font Mapping Entry (FGIDMAPE) tag:*

The FGIDMAPE tag defines the font mapping of individual font global identifiers (FGID) to font resources.

One or more of these tags must follow the FGIDMAPTBL tag. A group of FGIDMAPE tags must be followed by the EFGIDMAPTBL tag. The syntax is:

```
:FGIDMAPE
  FGID = Integer (FGID value)
  TECH= TYPE1|CIDFONT0
  NAME = font reference name

  WEIGHT = LIGHT|MED|BOLD
  ITALIC = YES|NO
  FILENAME = name of font file to embed
```

**Note:** The last parameter value must end with a period (.).

#### **FGID**

This is a required parameter. This parameter defines the font type global identifier.

#### **TECH**

This is a required parameter. This parameter specifies the technology of the font resource to use for the font global identifier. The possible values include:

##### **TYPE1**

This value indicates that the font reference uses the Adobe Type 1 font format. This font technology is supported by the PDF output data format.

##### **CIDFONT0**

This value indicates that the font reference is a CID-Keyed font based on the Adobe Type 1 font format. These fonts are used for large character sets, such as DBCS and Unicode. This font technology is supported by the PDF output data format.

#### **NAME**

This is a required parameter. This is a case-sensitive, null-terminated character string that specifies the name by which the font is to be referenced. It is limited to 64 characters in length. The characters specified in this parameter must be limited to those in the ISO-8859-1 standard encoding.

For a given font, the name used to reference can be dependent on the output data format, the application used to view or print the output, and the underlying operating system on which it is run.

For the PDF output format, the 14 standard PDF fonts listed as follows can generally be used as TYPE1 font references without embedding the font file.

- Courier
- Courier-Oblique
- Courier-Bold
- Courier-BoldOblique
- Helvetica
- Helvetica-Oblique

- Helvetica-Bold
- Helvetica-BoldOblique
- Times-Roman
- Times-Italic
- Times-Bold
- Times-BoldItalic
- Symbol
- ZapfDingbats

The following additional PostScript Type 1 fonts can also be specified without embedding the font file.

- AvantGarde-Book
- AvantGarde-BookOblique
- AvantGarde-Demi
- AvantGarde-DemiOblique
- Bookman-Demi
- Bookman-DemiItalic
- Bookman-Light
- Bookman-LightItalic
- Helvetica-Narrow
- Helvetica-Narrow-Oblique
- Helvetica-Narrow-Bold
- Helvetica-Narrow-BoldOblique
- NewCenturySchlbk-Roman
- NewCenturySchlbk-Italic
- NewCenturySchlbk-Bold
- NewCenturySchlbk-BoldItalic
- Palatino-Roman
- Palatino-Italic
- Palatino-Bold
- Palatino-BoldItalic
- Helvetica-Condensed
- Helvetica-Condensed-Bold
- Helvetica-Condensed-Oblique
- Helvetica-Condensed-BoldObl
- ZapfChancery-MediumItalic

To use other Type 1 fonts, the font files must be provided and specified with the FILENAME parameter.

To support languages with large character sets, such as Japanese Chinese, and Korean, PDF defines registered CMaps and character collections, which can be referenced with the following names.

*Table 8. Registered CMaps and character collections*

Name	Character collection	CIDFont
JpnSys	Adobe-Japan1	KozGoPro-Medium
JpnSys2	Adobe-Japan1	KozMinPro-Regular
ChsSys	Adobe-GB1	STSongStd-Light

Table 8. Registered CMaps and character collections (continued)

Name	Character collection	CIDFont
ChsSys2	Adobe-GB1	AdobeSongStd-Light
ChtSys	Adobe-CNS1	AdobeMingStd-Light
ChtSys2	Adobe-CNS1	MSungStd-Light
KorSys	Adobe-Korea1	AdobeMyungjoStd-Medium

#### **WEIGHT**

This is an optional parameter. It defines the degree of boldness of the typeface caused by the thickness of the strokes that form a graphic character. This parameter is used to guide the selection of a substitute font if the named reference is not available to the application or device that renders the output. The valid values are:

- LIGHT
- MEDIUM (default)
- BOLD

This parameter is ignored if the optional FILENAME parameter is specified.

#### **ITALIC**

This is an optional parameter. It indicates whether the characters slant to the right. This parameter is used to guide the selection of a substitute font if the named reference is not available to the application or the device that renders the output. The valid values are:

- NO (default)
- YES

This parameter is ignored if the optional FILENAME parameter is specified.

#### **FILENAME**

This is an optional parameter that specifies the name of the font file, or files, that contain the referenced font. This parameter is permitted only when the font technology parameter is TYPE1. This is a case-sensitive, null-terminated character string. It is limited to 64 characters in length. The characters specified in this parameter must be limited to those in the ISO-8859-1 standard encoding.

For the PDF output data format, this parameter instructs the transform to embed the font resource within the output document.

For TYPE1 fonts, the FILENAME parameter must contain the name of the font program files without their extension. Files of this name with the file extensions .pfb and .pfm must reside in the \QIBM\UserData\OS400\Fonts\Type1 directory or the directory specified in the optional font directory tag.

If the specified files are not found, message CPDCE03 is logged and the font is not embedded.

#### **Example 1**

The following example maps FGID 11 to the Helvetica-Oblique standard PDF font.

```
:FGIDMAPE
  FGID = 11
  TECH = TYPE1
  NAME = 'Helvetica-Oblique'.
```

#### **Example 2**

The following example maps all references to FGID 3412 to a font referenced by the name ZuzusPetals contained in the files BedfordFalls.pfb and BedfordFalls.pfm.



```
:FGIDMAPE
  FGID = 3412
  TECH = TYPE1
  NAME = 'ZuzusPetals'
  FILENAME = 'BedfordFalls'.
```

### Example 3

The following example maps FGID 304 to the JpnSys2 virtual DBCS font. The Adobe-Japan1 registered character collection is referenced.

```
:FGIDMAPE
  FGID = 304
  TECH = CIDFONT0
  NAME = 'JpnSys2'.
```

*End Font Mapping (EFGIDMAPTBL) tag:*

The EFGIDMAPTBL tag is used to end a group of font map table entries in the customization source.

There are no parameters on this tag. The syntax is:

```
:EFGIDMAPTBL.
```

*TrueType font mapping:*

TrueType and OpenType fonts can be mapped to font references recognized by the output data format.

You begin support for font mapping with the TTFMAPTBL tag and end the support with the ETTFMAPTBL tag. The TTFMAPE tags fall between defining each full font name (FFN) that is to be mapped to a font reference. Using the font mapping tags, your source could look like the following example:

```
:TTFMAPTBL.
  :TTFMAPE....
  :TTFMAPE....
  :TTFMAPE....
:ETTFMAPTBL.
```

For the PDF output data format, TrueType font references that are not mapped are embedded in the output document.

*TrueType Font Mapping (TTFMAPTBL) tag:*

The TTFMAPTBL tag defines the mapping of TrueType font references to font resources to be used in the output document.

This tag must be followed by one or more TTFMAPE tags that contain the font mapping. There are no parameters to this tag. The syntax follows:

```
:TTFMAPTBL.
```

*TrueType Font Mapping Entry (TTFMAPE) tag:*

The TTFMAPE tag defines the font mapping of individual TrueType font references to another font resource.

One or more of these tags must follow the TTFMAPTBL tag. A group of TTFMAPE tags must be followed by the ETTFMAPTBL tag. The syntax is:

#### :TTFMAPE

TTFONT = Full Font Name of the TrueType font  
TECH= TYPE1|CIDFONT0  
NAME = font reference name  
WEIGHT = LIGHT|MED|BOLD  
ITALIC = YES|NO

#### Notes:

- The last parameter value must end with a period (.).
- Validation is not performed to ensure that specified values identify a predefined character collection or CMap name.

#### TTFONT

This is a required parameter. This is a case-sensitive, null-terminated character string that identifies the TrueType font reference to be mapped. It is limited to 64 characters in length. The characters specified in this parameter must be limited to those in the ISO-8859-1 standard encoding. In AFP, TrueType fonts are referenced by their Full Font Name (FFN).

#### TECH

This is a required parameter. This parameter specifies the technology of the font resource to use. The possible values include:

##### TYPE1

This value indicates that the font reference uses the Adobe Type1 font format. This font technology is supported by the PDF output data format.

##### CIDFONT0

This value indicates that the font reference is a CID-Keyed font based on the Adobe Type 1 font format. These fonts are used for large character sets, such as DBCS and Unicode. This font technology is supported by the PDF output data format.

#### NAME

This is a required parameter. This is a case-sensitive, null-terminated character string that specifies the name by which the font is to be referenced. It is limited to 64 characters in length. The characters specified in this parameter must be limited to those in the ISO-8859-1 standard encoding. For a given font, the name used to reference can be dependent on the output data format, the application used to view or print the output, and the underlying operating system on which it is run.

For the PDF output format, the 14 standard PDF fonts listed as follows can generally be used as Type1 font references.

- Courier
- Courier-Oblique
- Courier-Bold
- Courier-BoldOblique
- Helvetica
- Helvetica-Oblique
- Helvetica-Bold
- Helvetica-BoldOblique
- Times-Roman
- Times-Italic
- Times-Bold
- Times-BoldItalic
- Symbol
- ZapfDingbats

The following additional PostScript Type 1 fonts can also be used.

- AvantGarde-Book
- AvantGarde-BookOblique
- AvantGarde-Demi
- AvantGarde-DemiOblique
- Bookman-Demi
- Bookman-DemiItalic
- Bookman-Light
- Bookman-LightItalic
- Helvetica-Narrow
- Helvetica-Narrow-Oblique
- Helvetica-Narrow-Bold
- Helvetica-Narrow-BoldOblique
- NewCenturySchlbk-Roman
- NewCenturySchlbk-Italic
- NewCenturySchlbk-Bold
- NewCenturySchlbk-BoldItalic
- Palatino-Roman
- Palatino-Italic
- Palatino-Bold
- Palatino-BoldItalic
- Helvetica-Condensed
- Helvetica-Condensed-Bold
- Helvetica-Condensed-Oblique
- Helvetica-Condensed-BoldObl
- ZapfChancery-MediumItalic

To support languages with large character sets, such as Japanese, Chinese, and Korean, PDF defines registered CMap and character collections, which can be referenced with the following names.

*Table 9. Registered CMaps and character collections*

Name	Character collection	CIDFont
JpnSys	Adobe-Japan1	KozGoPro-Medium
JpnSys2	Adobe-Japan1	KozMinPro-Regular
ChsSys	Adobe-GB1	STSongStd-Light
ChsSys2	Adobe-GB1	AdobeSongStd-Light
ChtSys	Adobe-CNS1	AdobeMingStd-Light
ChtSys2	Adobe-CNS1	MSungStd-Light
KorSys	Adobe-Korea1	AdobeMyungjoStd-Medium

## **WEIGHT**

This optional parameter defines the degree of boldness of the typeface caused by the thickness of the strokes that form a graphic character. It is used to guide the selection of a substitute font if the named reference is not available to the application or the device that renders the output. The valid values are as follows:

- LIGHT
- MEDIUM (default)
- BOLD

## ITALIC

This optional parameter indicates whether the characters slant to the right. This parameter is used to guide the selection of a substitute font if the named reference is not available to the application or the device that renders the output. The valid values are as follows:

- NO (default)
- YES

### Example 1

The following example maps the Monotype Sans WT font to the Helvetica-Oblique standard PDF font.

```
:TTFMAPE
  TTFONT = 'Monotype Sans WT'
  TECH = TYPE1
  NAME = 'Helvetica-Oblique'.
```

### Example 2

The following example maps the Monotype Sans Duospace WT SC font to the ChsSys2 virtual DBCS font. The Adobe-GB1 registered character collection is referenced.

```
:TTFMAPE
  TTFONT = 'Monotype Sans Duospace WT SC'
  TECH = CIDFONT0
  NAME = 'ChsSys2'.
```

### Example 3

The following example maps all references to the Thornsedale WT J font to the JpnSys virtual DBCS font. The desired attributes of a substitute typeface are proportionally spaced serif, bold, and italic.

```
:TTFMAPE
  TTFONT = 'Thornsedale WT J'
  TECH = CIDFONT0
  NAME = 'JpnSys'
  WEIGHT = BOLD
  ITALIC = YES.
```

*End TrueType Font Mapping (ETTFMAPTBL) tag:*

The ETTFMAPTBL tag is used to end a group of font map table entries in the customization source.

There are no parameters on this tag. The syntax follows:

```
:ETTFMAPTBL.
```

*Font directory:*

You can specify the integrated file system directory in which to search for font resources of a given technology.

You begin the syntax for font directories with the Font Directory Table (FNTDIRTBL) tag and end the syntax with the End Font Directory (EFNTDIRTBL) tag. The Font Directory Entry (FNTDIRE) tags between them list the directory that is to be searched for fonts of a given technology. Using the font directory tags, your source could look like the following example:

```
:FNTDIRTBL.
  :FNTDIRE....
  :FNTDIRE....
:EFNTDIRTBL.
```

## Font Directory Table (FNTDIRTBL) tag

The FNTDIRTBL tag defines the beginning of the font directory list. This tag must be followed by one or more FNTDIRE tags, which identify the directory where fonts of a given technology can be accessed. Only one FNTDIRTBL tag is allowed for each font technology. There are no parameters for this tag. The syntax follows:

```
:FNTDIRTBL.
```

## Font Directory Entry (FNTDIRE) tag

The FNTDIRE tag defines the font resource to be embedded. One or more of these tags must follow the FNTDIRTBL tag. A set of FNTDIRE tags must be followed by the EFNTDIRTBL tag. The syntax follows:

```
:FNTDIRE
    TECH= TYPE1
    PATH = font directory path
```

**Note:** The last parameter value must end with a period (.).

### TECH

This is a required parameter. This parameter specifies the technology of the font resources contain in the directory. The possible value is as follows.

### TYPE1

This value indicates that this font directory entry should be used for Adobe Type1 fonts. The default directory for fonts of this technology is `\QIBM\UserData\OS400\Fonts\Type1 .`

### PATH

This is a required parameter. This is a case-sensitive, null-terminated character string that specifies the path name that identifies the font directory. It is limited to 240 characters in length. The characters specified in this parameter must be limited to those in the ISO-8859-1 standard encoding.

## End Font Directory (EFNTDIRTBL) tag

The EFNTDIRTBL tag is used to end a group of font directory entries in the customization source. There are no parameters on this tag. The syntax follows:

```
:EFNTDIRTBL.
```

## Example

The following tags instruct the transform to search an alternate directory for Adobe Type 1 fonts.

```
:FNTDIRTBL.
:FNTDIRE
    TECH = TYPE1
    PATH = '/home/myfontdirectory'.
:EFNTDIRTBL.
```

If the files associated with any specified fonts are not found in this directory, message CPDCE03 is logged and the font is not embedded.

## Changing an existing printer device description

You can change the settings in an existing printer device description to enable host print transform.

**Prerequisites:** Before you change a printer device description, it is recommended that you:

- End the printer writer (ENDWTR command)
- Vary off the printer device (WRKCFGSTS command)

When working with existing printer device descriptions, you can enable the host print transform by changing certain parameter values in the printer device description.

To enable the host print transform, type the Change Device Description (Printer) (CHGDEVPRT) command and press F4 (Prompt).

### **Displaying the printer device description**

To verify your host print transform parameters, type the Display Device Description (DSPDEVD) command to display the printer device description.

If the default value \*MFRTYPMDL was specified for the PPRSRC1, PPRSRC2, and ENVELOPE parameters, the system-supplied values are shown when the device description is displayed.

Use the Change Device Description (Printer) (CHGDEVPRT) command to change any parameter values that are not correct for your printer.

### **IBM IBM i Access for Windows PC5250 configuration recommendations**

Configuring a PC5250 printer session to use the host print transform causes the conversion of the SCS printer data stream to the printer data stream on the IBM i operating system. This is especially beneficial if several different printer emulation packages are being used (for example, PC5250, workstation function (WSF), and others) because all of the output would look similar. This also provides control over how the output is printed by enabling you to retrieve, change, and create your own Workstation Customization Table.

When the printer session is started, these parameters are passed from the personal computer to the IBM i operating system. The printer device description is created or changed to reflect the values passed from the PC5250 Emulation program. Any changes must be made by changing the printer's emulation session on the personal computer instead of using the CHGDEVPRT command.

### **Maintaining printer customization**

Before you do the work of creating a workstation customization object for a printer, try printing jobs using the host print transform support. You might not need to create a customization object for the printer due to the extended support provided by the host print transform.

**Related information:**



Workstation Customization Programming PDF

### **3486, 3487, and 3488 InfoWindow display station configuration recommendations**

You can configure printers that are supported by the InfoWindow display as recommended in the InfoWindow display's configuration menus. You can configure any printer that is not included in the list of supported printers as a 4201/4202 printer.

When you leave the InfoWindow display's configuration menu, the printer device description is automatically created if both the following are true:

- The printer is powered on
- Automatic configuration is on

After the printer device description is created on i5/OS, enable the host print transform using the CHGDEVPRT command. The display's printer configuration is not used after the host print transform is enabled. The data stream generated for the printer is based on the MFRTYPMDL parameter value specified in the printer device description on i5/OS.

**Note:** Do not change the display's printer configuration after your device description has been created on the IBM i operating system. Doing so can cause the i5/OS printer device description to be

replaced. In this case, the host print transform is no longer enabled. The CHGDEVPRT command can be used to enable the host print transform again.

## Maintaining printer customization

Before you do the work of creating a workstation customization object for a printer, try printing jobs using the host print transform support. You might not need to create a customization object for the printer due to the extended support provided by the host print transform.

The 348x displays can use a workstation customization object (located in the device description of the display) to define the printer attached to it. Conversely, the host print transform uses a workstation customization object located in the device description of the printer. If you have customized some printer features in the display's workstation customization object, use of the host print transform overrides the customization.

However, if you need to customize your printer while using the host print transform, you should:

1. Make sure the host print transform is enabled.  
The TRANSFORM parameter value for the printer device description must be \*YES.
2. Use a workstation customization object that is compatible with the host print transform.  
Customize the printer features in this object to match the customized features that you used in the display's workstation customization object.
3. Use the workstation customization object name that is compatible with the host print transform as the value for the WSCST parameter in the printer device description.

**Remember:** The location of the WSCST object name (in the printer device description, not in the display device description) is important when using customization with the host print transform.

### Related information:



Workstation Customization Programming PDF

## 3477 InfoWindow display station configuration recommendations

You can configure printers that are supported by the InfoWindow display as recommended in Appendix A of the *IBM 3477 InfoWindow User's Guide*.

You can configure printers that are not supported by the 3477 with the following values:

- Type A for the **Printer Character Set** field
- 5256 for the **Printer Emulation** field

When you leave the display's configuration menu, the printer device description is automatically created if both the following are true:

- The printer is powered on
- Automatic configuration is on

After the printer device description is created on the IBM i operating system, enable the host print transform using the Change Device Description (Printer) (CHGDEVPRT) command. The display's printer configuration is not used after the host print transform is enabled. The data stream generated for the printer is based on the MFRTYPMDL parameter value specified in the printer device description on the IBM i operating system.

**Note:** Do not change the display's printer configuration after your device description has been created on the IBM i operating system. Doing so can cause the i5/OS printer device description to be replaced. In this case, the host print transform is no longer enabled. The CHGDEVPRT command can be used to enable the host print transform again.

## Understanding the 3477 power-on initialization sequence

The 3477 sends a power-on initialization sequence to the attached printer every time the printer or display is powered on. This initialization is designed for IBM printers. If a printer that does not support the IBM data stream is attached, some unrecognizable characters might be printed. After this initialization sequence, the data stream generated is based on the MFRTYPMDL parameter in the printer device description on the IBM i operating system. No more unrecognizable characters are printed.

If your 3477 is a Model H, you can turn off the power-on initialization sequence by doing the following:

1. From the 3477's setup menu, select *Test Workstation*.
2. Press the Alt, Shift, and Setup keys together.
3. The F6 key is now set to switch between power-on initialization active and power-on initialization not active.

Pressing F6 displays +6 or \*6 on the bottom of the screen. If +6 is displayed, power-on initialization is not active. If \*6 is displayed, power-on initialization is active.

4. Press the Reset key to return to the 3477 setup menu.

## Maintaining printer customization

Before you do the work of creating a workstation customization object for a printer, try printing jobs using the host print transform support. You might not need to create a customization object for the printer due to the extended support provided by the host print transform.

The 3477 Model H is the only 3477 InfoWindow display that supports customization. The 3477 Model H display can use a workstation customization object (located in the device description of the display) to define the printer attached to it. Conversely, the host print transform uses a workstation customization object located in the device description of the printer. If you have customized some printer features in the display's workstation object, use of the host print transform overrides the customization.

However, if you need to customize your printer while using the host print transform, you should:

1. Make sure the host print transform is enabled.  
The TRANSFORM parameter value for the printer device description must be \*YES.
2. Use a workstation customization object that is compatible with the host print transform.  
Customize the printer features in this object to match the customized features that you used in the display's workstation customization object.
3. Use the workstation customization object name that is compatible with the host print transform as the value for the WSCST parameter in the printer device description.

**Remember:** The location of the WSCST object name (in the printer device description, not in the display device description) is important when using customization with the host print transform.

### Related information:



Workstation Customization Programming PDF

## 3197 display station configuration recommendations

You can refer to the *IBM 3197 Display Stations User's Guide* to set up your display for a printer session.

Select *Display-Printer Mode* with a *Printer Device ID* of 5256. Specify 850 as the *Printer Character Set Selection*.

When you leave the InfoWindow display's configuration menu, the printer device description is automatically created if both the following are true:



- The printer is powered on
- Automatic configuration is on

After the printer device description is created on the IBM i operating system, enable the host print transform using the Change Device Description (Printer) (CHGDEVPRT) command. The display's printer configuration is not used after the host print transform is enabled. The data stream generated for the printer is based on the MFRTYPMDL parameter value specified in the printer device description on i5/OS.

**Notes:**

1. The 3197 sends a power-on initialization sequence to the attached printer every time the printer or display is powered on. This initialization is designed for IBM printers. If a printer is attached that does not support the IBM data stream, some unrecognizable characters might be printed. After this initialization sequence, the data stream generated is based on the MFRTYPMDL parameter in the printer device description on i5/OS. No more unrecognizable characters are printed.
2. Do not change the display's printer configuration after your device description has been created on the IBM i operating system. Doing so can cause the i5/OS printer device description to be replaced. In this case, the host print transform is no longer enabled. The CHGDEVPRT command can be used to enable the host print transform again.

The order in which the 3197 display and its attached printer are powered on can affect the system's ability to recognize the printer. This order is dependent on the model of the 3197. Use the Work with Configuration Status (WRKCFGSTS) command to check the status of a printer device. If the status of the printer is **vary on pending** or if the printer is not automatically configured, reverse the order in which the display and printer were powered on.

**ASCII workstation controller configuration recommendations**

There is no automatic configuration capability for printers attached to the ASCII workstation controller. The Create Device Description (Printer) (CRTDEVPRT) command must be used to create the configuration descriptions. For existing printers, use the Change Device Description (Printer) (CHGDEVPRT) command.

Configure printers that are not supported by the ASCII workstation controller with these values: device type as 4019, model as Model 1, and emulated printer as 3812. Many of the other parameters depend on the attached printer (line speed, data bits, and so on). To avoid i5/OS parameter checking, specify \*YES as the value on the emulating ASCII parameter in the printer device description. For example, this allows a communications line speed of 38 400 bps.

After the host print transform is enabled, the device type and model parameters in the device description have no effect on the data stream sent to the printer. The data stream generated for the printer is based on the MFRTYPMDL parameter in the i5/OS printer device description.

**Maintaining printer customization**

Before you do the work of creating a workstation customization object for a printer, try printing jobs using the host print transform support. You might not need to create a customization object for the printer due to the extended support provided by the host print transform.

The ASCII workstation controller can use a workstation customization object in the printer device description to define the printer's characteristics. That workstation customization object can be customized to use the special features of a printer that the ASCII workstation controller does not support.

If the host print transform is enabled, the customization object specified for the ASCII workstation controller printer device description must be removed or replaced. The customization objects created for the ASCII workstation controller are not compatible with the host print transform.

However, if you need to customize your printer while using the host print transform, you should:

1. Make sure the host print transform is enabled.  
The TRANSFORM parameter value for the printer device description must be \*YES.
2. Use a workstation customization object that is compatible with the host print transform.  
Customize the printer features in this object to match the customized features that you initially used in the workstation customization object specified in the ASCII workstation controller printer device description.
3. Use the workstation customization object name that is compatible with the host print transform as the new value for the ASCII workstation controller parameter in the printer device description.

**Remember:** The ASCII workstation controller and the host print transform both use the workstation customization object parameter in the printer device description. That object must contain the customized features and be compatible with the host print transform.

#### Related information:



Workstation Customization Programming PDF

### IBM remote 5250 emulation program configuration recommendations

You can see the *Remote 5250 Emulation Program User's Guide*, G570-2203, for information about configuring printer sessions.

For example, use the IBM5204.PDT file and select 5219 Printer emulation for the IBM 5204 Quickwriter. Configure printers that were not previously supported by the emulation program using the TBLPRT.PDT and selecting 5219 printer emulation.

When the printer session is started, the system automatically creates the printer device description (if automatic configuration is turned on).

On the IBM i operating system, use the Change Device Description (Printer) (CHGDEVPRT) command to enable the host print transform.

### Maintaining printer customization

Before you do the work of creating a workstation customization object for a printer, try printing jobs using the host print transform support. You might not have to create a customization object for the printer due to the extended support provided by the host print transform.

The Remote 5250 Emulation Program supports printers through a printer description table (PDT). If you have customized a printer by changing the supplied PDT, use of the host print transform overrides this customization.

However, if you need to customize your printer while using the host print transform, you should:

1. Make sure the host print transform is enabled.  
The TRANSFORM parameter value for the printer device description must be \*YES.
2. Make a note of the features you customized in the PDT.
3. Use a workstation customization object that is compatible with the host print transform.  
Update the features in this object to reflect the customized features you used in the PDT.
4. Use the workstation customization object name that is compatible with the host print transform as the value for the WSCST parameter in the printer device description.

See Workstation Customization Programming PDF for detailed information on customizing printers.

**Related information:**



Workstation Customization Programming PDF

## Configuring image print transform

The image print transform function works with ASCII printers, as well as IPDS printers that have AFP(\*YES) specified in the configuration.

### Printing to an ASCII printer with image print transform

You can follow these steps to enable the image print transform function when printing to an ASCII printer.

- Ensure that the spooled file is a \*USERASCII spooled file.
- Verify that the printer device description has the TRANSFORM field set to \*YES.
- Verify that the printer device description has the IMGCFG field set to a valid value other than \*NONE.

The TRANSFORM field and the IMGCFG field can be set when the device description is created with the Create Device Desc (Printer) (CRTDEVPRT) command, or changed after the device description was created with the Change Device Desc (Printer) (CHGDEVPRT) command.

### Printing to an IPDS printer with image print transform

You can follow these steps to enable the image print transform function when printing to an IPDS printer that has AFP(\*YES) specified in the configuration.

- Ensure that the spooled file is a \*USERASCII spooled file.
- Verify that the printer device description has the IMGCFG field set to a valid value other than \*NONE.

The IMGCFG field can be set either when the device description is created with the Create Device Desc (Printer) (CRTDEVPRT) command, or changed after the device description was created with Change Device Desc (printer) (CHGDEVPRT) command.

### Printing with remote output queues and image print transform

You can follow these steps to enable the image print transform function when printing or sending spooled files to a remote output queue.

- Ensure that the spooled file is a \*USERASCII spooled file.
- Verify that the output queue has the TRANSFORM field set to \*YES.
- Verify that the output queue has the IMGCFG field set to a valid value other than \*NONE.

You can set the TRANSFORM field and the IMGCFG field when you create the output queue with the Create Output Queue (CRTOUTQ) command. You can change both fields after creating the output queue with the Change Output Queue (CHGOUTQ) command.

---

## Managing printing

Administering your company's printing solution after the initial setup and configuration involves dealing with changing user and application requirements and handling any upgrades or performance considerations as your system grows.

The following tasks can help you manage your printing environment:

- Manage spooled files
- Printer file related tasks
- Manage printers
- Manage printer writer

- Manage remote writer
- Manage IPP server
- Miscellaneous printing tasks

See Example: Print font samples for information about how to print font samples.

**Related reference:**

“Example: Print font samples” on page 236

These examples contain instructions and source code that allow you to print a font global identifier (FGID), font character set, or coded font and to see what it looks like.

## Managing spooled files

Spooled files management includes tasks such as holding a spooled file, releasing a spooled file, and moving a spooled file.

### Displaying a list of spooled files

To display a list of spooled files (printer output), use either of these methods.

#### System i Navigator

1. Expand **Basic Operations**.
2. Click **Printer Output**.

The default setting is to display all printer output associated with the current user. You can display other printer output by right-clicking **Printer Output** and then clicking **Customize this view > Include**.

#### Character-based interface

Use the Work with Spooled Files (WRKSPLF) command.

**Related information:**

Work with Spooled Files (WRKSPLF) command

### Displaying the contents of a spooled file

To display the contents of a spooled file (printer output), use either of these methods.

#### System i Navigator

1. Expand **Basic Operations**.
2. Click **Printer Output**.
3. Right-click the printer output file that you want to display.
4. Click **Open**.

#### Character-based interface

Use the Work with Spooled Files (WRKSPLF) command, and then use option 5 (Display).

**Notes:**

1. The System i Navigator interface has the additional capability to display ASCII spooled files.
2. The character-based interface has the additional capability of being able to display \*LINE and \*IPDS spooled files.

**Related information:**

Work with Spooled Files (WRKSPLF) command

## Displaying messages associated with a spooled file

To display messages associated with a spooled file (printer output), use either of these methods.

### System i Navigator

1. Expand **Basic Operations**.
2. Click **Printer Output**.
3. Right-click the printer output file that has a message.
4. Click **Reply**.

### Character-based interface

Use the Work with Spooled Files (WRKSPLF) command, and then use option 7 (Message).

#### Related information:

Work with Spooled Files (WRKSPLF) command

## Holding a spooled file

To temporarily prevent the spooled file (printer output) that you selected from printing, use either of these methods.

### System i Navigator

1. Expand **Basic Operations**.
2. Click **Printer Output**.
3. Right-click the printer output file that you want to hold.
4. Click **Hold**.
5. Specify the hold options and click **OK**.

### Character-based interface

Use the Work with Spooled Files (WRKSPLF) command, and then use option 3 (Hold).

**Note:** The character-based interface has the additional capability of being able to hold all spooled files with the same user, print device, form type, user data, or ASP with one action.

#### Related information:

Work with Spooled Files (WRKSPLF) command

## Releasing a spooled file

To release a spooled file (printer output) that is being held, use either of these methods.

### System i Navigator

1. Expand **Basic Operations**.
2. Click **Printer Output**.
3. Right-click the printer output file that you want to hold.
4. Click **Release**.

### Character-based interface

Use the Work with Spooled Files (WRKSPLF) command, and then use option 6 (Release).

**Note:** The character-based interface has the additional capability of being able to release all spooled files with the same user, print device, form type, user data, or ASP with one action.

#### Related information:

Work with Spooled Files (WRKSPLF) command

## Moving a spooled file

To move the spooled file (printer output) from one output queue to another output queue, use either of these methods.

### System i Navigator

1. Expand **Basic Operations**.
2. Click **Printer Output**.
3. Right-click the printer output file that you want to move.
4. Click **Move**.
5. Specify the name of the printer or output queue where you want to move the printer output and click **OK**.

### Character-based interface

Use the Work with Spooled Files (WRKSPLF) command, and then use option 2 (Change).

**Note:** The character-based interface has the additional capability of being able to change an attribute for all spooled files with the same user, print device, form type, user data, or ASP with one action.

### Related concepts:

“Redirecting output considerations” on page 110

Spooled or nonspooled output intended for a printer can be redirected to another printer. However, each file is checked to ensure that the file attributes (device type, number of lines per inch, number of characters per inch, page length, and page width) and any advanced functions used by the file (such as variable LPI, variable font, or defined characters) are valid on the new printer.

### Related information:

Work with Spooled Files (WRKSPLF) command

## Deleting spooled files

To delete a spooled file (printer output), use either of these methods.

### System i Navigator

1. Expand **Basic Operations**.
2. Click **Printer Output**.
3. Right-click the printer output file that you want to delete.
4. Click **Delete**.
5. Click **Delete** to confirm.

### Character-based interface

Use the Work with Spooled Files (WRKSPLF) command, and then use option 4 (Delete).

**Note:** The character-based interface has the additional capability of being able to delete all spooled files with the same user, print device, form type, user data, or ASP with one action.

### Related information:

Work with Spooled Files (WRKSPLF) command

## Converting a spooled file to PDF

- 1 To convert a spooled file to a PDF file, follow these steps. An IBM Infoprint Server for iSeries license is required.

## | **System i Navigator**

1. Right-click the printer output file that you want to convert.
2. Click **Convert to PDF**.
3. The PDF Convert wizard opens.
4. Follow the instructions in the wizard.

## | **IBM Navigator for i**

1. Select **Basic Operations** from the IBM Navigator for i window.
2. Select **Printer Output**.
3. Select the printer output file that you want to convert.
4. From the **Select Action** menu, select **Export as PDF to output queue**, **Export as PDF to IFS**, or **Export as PDF to e-mail** and click **Go**.
5. The PDF convert wizard opens. Follow the instructions in the wizard.

### **Related tasks:**

“Exporting a spooled file to your PC”

To export a spooled file (printer output) to your PC file system where you can work with it using PC tools, follow these steps.

## | **Exporting a spooled file to your PC**

To export a spooled file (printer output) to your PC file system where you can work with it using PC tools, follow these steps.

## | **IBM Navigator for i**

1. Select **Basic Operations** from the IBM Navigator for i window.
2. Select **Printer Output**.
3. Select the printer output file that you want to copy to the PC.
4. From the **Select Action** menu, select **Export as ASCII Text** to convert the spooled file to an ASCII text file. Select **Export as PDF to client file system** to convert the spooled file to a PDF. Select **Export as Raw** to copy the spooled file to the PC with no conversion.
5. Specify whether you want to open or save the file. If you want to save the file, specify the file name and where you want to store it. Then click **Save**.

### **Related tasks:**

“Converting a spooled file to PDF” on page 214

To convert a spooled file to a PDF file, follow these steps. An IBM Infoprint Server for iSeries license is required.

## **Copying a spooled file to a physical file**

To copy a spooled file to a physical file, use the Copy Spooled Files (CPYSPLF) command.

The original spooled file is not affected by the copy operation and can still be printed by the printer writer program. When copying a spooled file to a physical file, many device attributes cannot be copied.

### **Related information:**

Copy Spooled Files (CPYSPLF) command

## **Sending a spooled file to another user or system**

To send a spooled file (printer output) to a remote system that is running TCP/IP or to another user on the Systems Network Architecture distribution services (SNADS) network, use either of these methods.

## **System i Navigator**

1. Expand **Basic Operations**.
2. Click **Printer Output**.

3. Right-click the printer output file that you want to send.
4. Click **Send**.
5. Click **Send via TCP/IP** to send the printer output to a remote system that is running TCP/IP or click **Send via SNA** to send the printer output to another user on the SNADS network.
6. Specify the send options and click **OK**.

#### Character-based interface

Use the **Work with Spooled Files (WRKSPLF)** command, and then use option 1 (Send) to send the printer output to another user on the SNADS network. To send a spooled file to another user using TCP/IP, use the **Send TCP/IP Spooled File (SNDTCPSPLF)** command.

#### Related information:

Work with Spooled Files (WRKSPLF) command

Send TCP/IP Spooled File (SNDTCPSPLF) command

### Changing attributes of a spooled file

To change a spooled file's (printer output's) attributes, use either of these methods.

#### System i Navigator

1. Expand **Basic Operations**.
2. Click **Printer Output**.
3. Right-click the printer output file that you want to change.
4. Click **Properties**.
5. Specify the attributes or properties that you want to change and click **OK**.

#### Character-based interface

Use the Work with Spooled Files (WRKSPLF) command, and then use option 2 (Change).

**Note:** The character-based interface has the additional capability of being able to change an attribute for all spooled files with the same user, print device, form type, user data, or ASP with one action.

#### Related information:

Work with Spooled Files (WRKSPLF) command

### Restarting the printing of a spooled file

To restart printing a spooled file on a particular page, follow these steps.

#### System i Navigator

1. Expand **Basic Operations > Printer Output**.
2. Right-click the printer output file for which you want to restart printing and select **Properties**.
3. On the **General** tab of the Properties window, click **Restart Printing**.
4. On the window that displays next, specify the page information and click **OK**.

### Suspending one spooled file and printing another

To temporarily stop the printing of one spooled file (printer output) and start the immediate printing of another spooled file (printer output), follow these steps.

#### System i Navigator

1. Expand **Basic Operations**.
2. Click **Printer Output**.
3. Right-click the printer output file that you want to begin printing next.



4. Click **Print next**. This printer output is moved to the top of the output queue.
5. Right-click the printer output file that is currently printing.
6. Click **Hold**.
7. Specify to hold **At end of page** and click **OK**. This printer output stops printing at the end of the current page. The next printer output in the output queue starts printing.
8. Right-click the printer output file that is stopped.
9. Click **Print next**. This printer output is moved to the top of the output queue and will resume printing with the page following the last page printed.

### **Enabling spooled file notification message**

To be notified when a spooled file (printer output) completes printing or is held by the printer writer program, you need to enable the spooled file notification function.

#### **System i Navigator**

1. Expand **Users and groups**.
2. Click **All users**.
3. Double-click the user name that you want to change.
4. Click **Jobs**.
5. Click **Display Session**.
6. Select **Send message to spooled file owner**.

### **Controlling the number of spooled files**

The number of spooled files on your system should be limited. When a job is completed, spooled files and internal job control information are kept until the spooled files are printed or canceled. The number of jobs on the system and the number of spooled files known to the system increase the amount of time needed to perform IPL and internal searches, and increase the amount of temporary storage required.

Periodically identify spooled files that are no longer needed and delete them. For more information about how to display a list of spooled files, see [Display a list of spooled files](#).

You can control the number of generated job logs with the LOG and LOGOUTPUT parameters on the Create Job Description (CRTJOB) or Change Job (CHGJOB) commands, or with the QLOGOUTPUT system value. For more information, see [Job log pending](#).

You can specify the maximum number of spooled files generated by a job with the QMAXSPLF system value.

- | You can also control the number of days that joblogs and other system output stays on the system by making use of the system cleanup function. For more information, see the [Change Cleanup](#) command.

#### **Related tasks:**

“[Displaying a list of spooled files](#)” on page 212

To display a list of spooled files (printer output), use either of these methods.

#### **Related information:**

Create Job Description (CRTJOB) command

Change Job (CHGJOB) command

Job log pending

Change Cleanup (CHGCLNUP) command

## Deleting expired spooled files

You can use the EXPDATE or DAYS parameter on the Change Printer File (CHGPRTF), Create Printer File (CRTPRTF), Change Spooled File Attributes (CHGSPLFA), or Override with Printer File (OVRPRTF) command to make a spooled file eligible for deletion using the Delete Expired Spooled files (DLTEXPSPLF) command.

For example, the following command creates a job schedule entry which causes the DLTEXPSPLF command to delete all expired spooled files on your system every day:

```
ADDJOBSCDE JOB(DLTEXPSPLF) CMD(DLTEXPSPLF ASPDEV(*ALL)) FRQ(*WEEKLY) SCDDATE(*NONE) SCDDAY(*ALL)
SCDTIME(010000) JOBQ(QSYS/QSYSNOMAX) TEXT('DELETE EXPIRED SPOOLED FILES SCHEDULE ENTRY')
```

### Related information:

Change Printer File (CHGPRTF) command

Create Printer File (CRTPRTF) command

Change Spooled File Attributes (CHGSPLFA) command

Override with Printer File (OVRPRTF) command

Delete Expired Spooled files (DLTEXPSPLF) command

## Reclaiming spooled file storage

You can use the Reclaim Spool Storage (RCLSPLSTG) command or the Automatically clean up unused printer output storage (QRCLSPLSTG) system value to reclaim spooled file storage. These are the only allowable ways to remove spooled database members from the QSPL or QSPLxxxx libraries. Any other way can cause severe problems.

For more information about spooled file storage, see Spooled file library.

### Automatically clean up unused printer output storage (QRCLSPLSTG) system value

Use the Automatically clean up unused printer output storage (QRCLSPLSTG) system value to adjust the desired balance between spool performance and auxiliary storage. This system value can be used to clean up unused printer output storage on system auxiliary storage pools (ASPs), basic user ASPs and independent ASPs. For more information see, Storage system values: Automatically clean up unused printer output storage in the System values topic.

**Note:** System performance is degraded if Automatically clean up unused printer output storage (QRCLSPLSTG) is set to 0 days.

Assume that one of your application programs had an error and it produced thousands of spooled files that were of no value to you. When this happened those spooled files used lots of storage space on your system. To reclaim the spool storage, complete the following tasks:

1. Change the Automatically clean up unused printer output storage (QRCLSPLSTG) system value to 1.
2. Delete all the unwanted spooled files that the application program created. Note the time of day you deleted all the unwanted spooled files.
3. After 24 hours, provided the empty spooled file members are not reused, the system reclaims the auxiliary storage that was being used by the empty spooled files.
4. Change the Automatically clean up unused printer output storage (QRCLSPLSTG) system value back to its former value.

### Reclaim Spool Storage (RCLSPLSTG) command

Alternatively, you can immediately reclaim all empty spooled file members by using the Reclaim Spool Storage (RCLSPLSTG) command with the DAYS parameter set to \*NONE. This command can be used to clean up unused printer output storage on system auxiliary storage pools (ASPs), basic user ASPs, and independent ASPs.

**Notes:**

1. Any unused database members are deleted immediately when running the RCPLSPLSTG command with the Days parameter set to \*NONE. That means that there is no pool of unused members that can be used when creating spooled files.
2. Lock contention can occur on output queues or spool database files, resulting in bottlenecks and severe performance problems.

Assume that one of your application programs had an error and it produced thousands of spooled files that were of no value to you. When this happened those spooled files used lots of storage space on your system. To reclaim the spool storage, complete the following tasks:

1. Delete all the unwanted spooled files that the application program created.
2. Run the RCLSPLSTG command with the DAYS parameter set to \*NONE. The system immediately reclaims all auxiliary spool storage that was being used by the unwanted spooled files.

**System ASP storage**

You can reduce the amount of storage taken up by spooled files by moving or creating spooled files directly into a user auxiliary storage pool (ASP) or independent ASP. You can accomplish this by specifying \*OUTQASP on the SPLFASP parameter when creating an output queue in a library that is located in the user ASP or independent ASP.

All spooled files that you place in this output queue have the spooled file data stored in the user ASP or independent ASP in a library QSPLxxxx, where xxxx is the user ASP or independent ASP number.

**Note:** For files on a user ASP, the links to the job still reside on the system ASP. If the system ASP is lost, all spooled files in the user ASPs are lost. If a user ASP is lost, only spooled files in that user ASP are lost.

**Related concepts:**

“Spooled file library” on page 23

The spooled file library (QSPL or QSPLxxxx, where xxxx is the number of the basic user ASP or primary independent ASP) contains database files that are used to store data for inline data files and spooled files. Each file in library QSPL or QSPLxxxx can have several members. Each member contains all the data for an inline data file or spooled file.

**Related information:**

Storage system values: Automatically clean up unused printer output storage

Reclaim Spool Storage (RCLSPLSTG) command

Reclaim spooled file storage experience report

**Saving and restoring spooled files**

You can use the SPLFDTA parameter on the Save Library (SAVLIB), Save Object (SAVOBJ), Restore Library (RSTLIB), and Restore Object (RSTOBJ) CL commands to save and restore spooled files without losing the print fidelity, attributes, or identity of the spooled files.

To maintain 100% print fidelity when you restore saved spooled files, you must:

- Save and restore all external resources within the libraries that they existed in when the spooled file was created.
- Make sure that the user profile (spooled file owner) exists and has the appropriate authorities to all the external resources required for the spooled file to print, including integrated file system directories,

true type font integrated file system files, font resource objects, overlays, form definitions, page definitions, page segments, and embedded object integrated file system files.

- Make sure that the Resource Allocation Table (RAT) has the same fonts and linked fonts as when the spooled file was created. This is only necessary for those spooled files that use true type linked fonts.
- Make sure that all of the true type fonts that are used by the spooled files are in the appropriate directories on your system.
- Make sure that the environment variable QIBM\_AFP\_RESOURCES\_PATH is set to the appropriate path for those spooled files that use embedded objects but were not directory qualified.
- Restore all embedded objects used by spooled files into the same directories that they were in when the files were created.

When a spooled file is in the process of being saved or restored, another restore or save operation will be blocked. The blocked operation might fail to save or restore that spooled file. A diagnostic message will be issued when this occurs.

When a spooled file is restored, it is reattached to the original job if the original job still exists on the system. If the original job does not exist, then the spooled file is restored in a detached state. If the spooled file is restored in a detached state, it is possible to have more than one spooled file with the same fully qualified job name, spooled file name, and spooled file number. Under these conditions, access to the restored spooled file requires the user or application to include the job system name and spooled file creation date. This enables the operating system to select the correct spooled file.

The order of spooled files that you restore is determined mainly by attributes such as the status of the file and is not necessarily the order in which you restore them. For more information, see Order of spooled files on an output queue.

For information about the procedures that were used to save and restore spooled files before V5R4 of i5/OS, see Save spooled files in the Systems management topic collection.

**Related concepts:**

“Order of spooled files on an output queue” on page 12

The order of spooled files on an output queue is mainly determined by the status of the spooled file.

**Related information:**

Save Library (SAVLIB) command

Save Object (SAVOBJ) command

Restore Library (RSTLIB) command

Restore Object (RSTOBJ) command

Save spooled files

## Controlling printing by spooled file size

You can use the MAXPAGES parameter on the Create Output Queue (CRTOUTQ) or Change Output Queue (CHGOUTQ) command to control the printing of spooled files by size.

For example, assume that you want to restrict spooled files with more than 40 pages from printing between 8 a.m. and 4 p.m. on output queue MYOUTQ. Between noon and 1 p.m. you want to allow spooled files with 10 pages or less to print. The following command implements these restrictions:

```
CHGOUTQ OUTQ(MYOUTQ) MAXPAGES((40 0800 1600) (10 1200 1300))
```

**Related information:**

Create Output Queue (CRTOUTQ) command

Change Output Queue (CHGOUTQ) command

## Repairing output queues and spooled files

The **Start Spool Reclaim (STRSPLRCL)** command repairs output queues and spooled files that are left in unrecoverable states.

If a writer job ends abnormally or an incomplete update has occurred, the output queue or spooled files associated with the writer job could remain in various statuses (such as WTR, PRT, PND, and MSGW). The STRSPLRCL command does not wait for the spool reclaim function to complete. You can use this command to fix output queues and spooled files in SYSDATABASE, in an independent auxiliary storage pool (IASP), or in both.

If an asterisk (\*) is specified for the **ASP group (ASPGRP)** parameter and the current thread has an ASP group in its name space, two separate requests to reclaim output queues could be sent. If a matching output queue is found in \*SYSDATABASE, a request will be sent to the spool maintenance job in \*SYSDATABASE. If a matching output queue is found in the specified ASP group, a request is sent to the spooled maintenance job for that ASP group. When the reclaim function has completed, message CPC3309 is sent to the QHST and QSYSOPR message queues by each spooled maintenance job.

### Related information:

Start Spool Reclaim (STRSPLRCL) command

## Printer-file-related tasks

Tasks related to printer files include tasks such as changing printer files, overriding printer files, and deleting printer file overrides.

### Changing printer files

You can change the parameters of a printer file by using the Change Printer File (CHGPRTF) CL command.

Using the CHGPRTF command makes the specified changes permanent for the current active session and for all future sessions.

Assume that you have a mailing label program that uses a printer file named LABELPR3. You now want to have two sets of mailing labels every time the mailing label program is run and you want the spooled file to go to the output queue named LABELS. The CL command looks like this:

```
CHGPRTF FILE(LABELPR3) COPIES(2) OUTQ(LABELS)
```

When you enter this CHGPRTF command, it becomes effective immediately.

When any application program that uses printer file LABELPR3 is run, the spooled file goes to output queue LABELS and two copies of the mailing labels are printed.

### Related information:

Change Printer File (CHGPRTF) command

### Overriding printer files

You can temporarily specify a different printer file or printer file attribute by using the Override Printer File (OVRPRTF) CL command.

### Related concepts:

“Printer file overrides” on page 6

Overriding files (printer files, display files, diskette files, database files, and tape files) can be done by commands, from CL programs, or from high-level language programs. Overrides can be called from different levels (programs calling another program). This article discusses overriding printer files.

### Related information:

Override Printer File (OVRPRTF) command

## Overriding file attributes:

The simplest form of overriding a file is to override some attributes of the file.

For example, assume that you create a printer file named OUTPUT with these attributes:

- Page size: 66 by 132
- Lines per inch: 6
- Copies of printed output: 2
- Pages for file separators: 2
- Overflow line number: 55

The Create Printer File (CRTPRTF) command used to create this file looks like this:

```
CRTPRTF FILE(QGPL/OUTPUT) SPOOL(*YES)
PAGESIZE(66 132) LPI(6)
COPIES(2) FILESEP(2) OVRFLW(55)
```

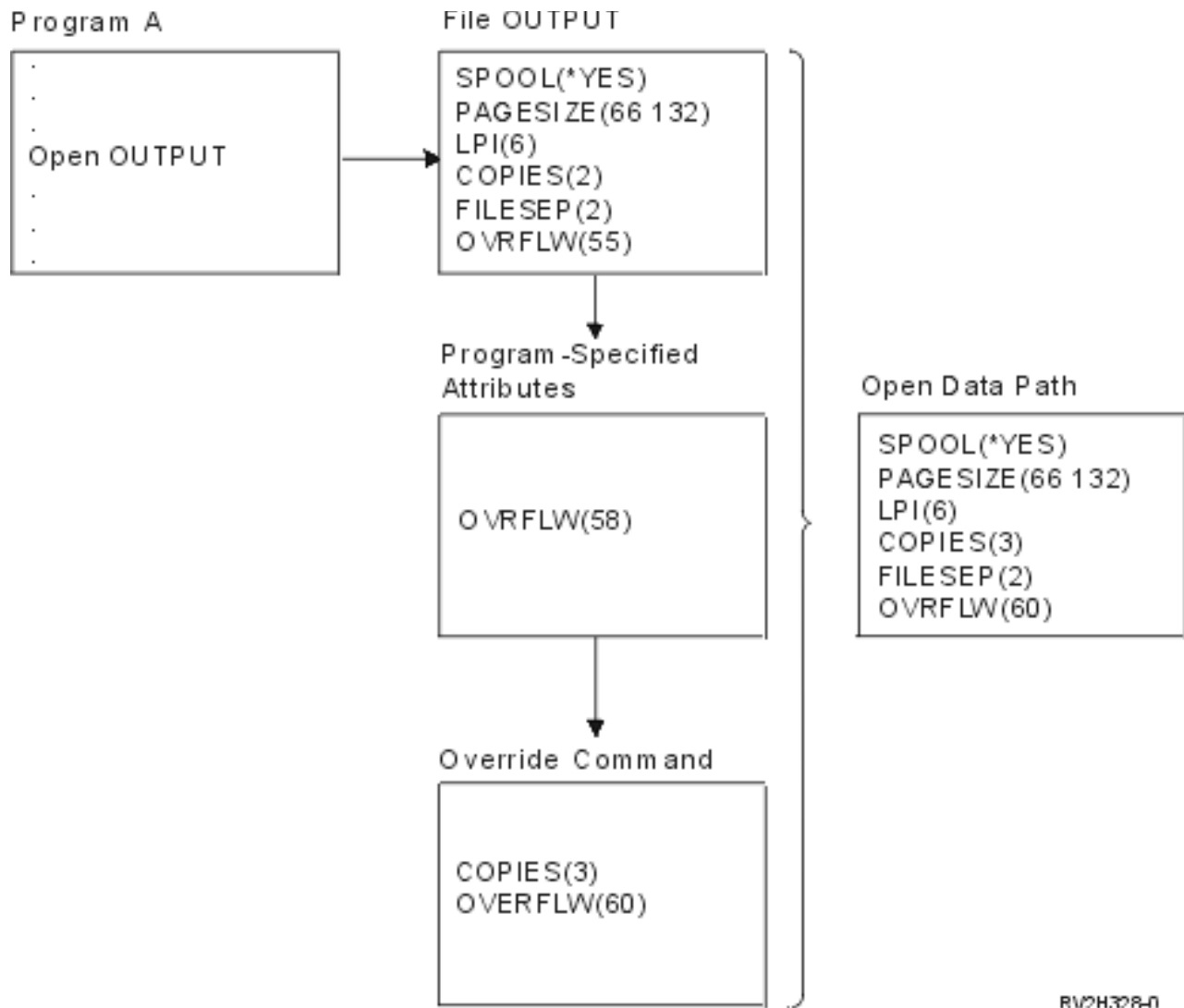
The printer file named OUTPUT is specified in your application program with an overflow line number of 58. However, before you run the application program, you want to change the number of copies of printed output to 3 and the overflow line to 60. The override command looks like this:

```
OVRPRTF FILE(OUTPUT) COPIES(3) OVRFLW(60)
```

When you call the application program, three copies of the output are printed.

When the application program opens the file, the file overrides, program-specified attributes, and file attributes are merged to form the open data path (ODP), which is used to manage the file during the running of the program. File overrides have precedence over program-specified attributes. Program-specified attributes have precedence over file-specified attributes. In this example, when the file is opened and output operations are performed, spooled output is produced with a page size of 66 by 132, six lines per inch, three copies, two file separator pages, and overflow at 60 lines.

The following chart explains this example.



R12H328-0

### Overriding file names or types:

Another simple form of overriding a file is to change which file is used by the program. This might be useful for files that have been moved or renamed after the program has been compiled.

For example, you want the output from your application program to be printed using the printer file named REPORTS instead of the printer file named OUTPUT (OUTPUT is specified in the application program).

Before you run the program, enter the following command:

```
OVRPRTF FILE(OUTPUT) TOFILE(REPORTS)
```

The file REPORTS must have been created (CRTPRTF command) before it can be used.

If you want to override to a different type of file, you use the override command for the new type of file. For example, if you are overriding a diskette file with a printer file, use the Override with Printer File (OVRPRTF) command.

## Overriding file names or types and file attributes of the new file:

This form of overriding files is a combination of overriding file attributes and overriding file names or types. With this form of override, you can override the file that is to be used in a program and you can also override the attributes of the overriding file.

For example, you want the output from your application program to be printed using printer file REPORTS instead of printer file OUTPUT (OUTPUT is specified in the application program). In addition to having the application program use printer file REPORTS, you want to override the number of printed copies to three. Assume that printer file REPORTS was created with the following command:

```
CRTPRTF FILE(REPORTS) SPOOL(*YES)
PAGESIZE (68 132) LPI(8)
OVRFLW(60) COPIES(2) FILESEP(1)
```

Before you run the program, enter the following command:

```
OVRPRTF FILE(OUTPUT) TOFILE(REPORTS) COPIES(3)
```

Then call the application program, and three copies of the output are printed using the REPORTS printer file.

Note that this is not equal to the following two override commands:

### Override 1

```
OVRPRTF FILE(OUTPUT) TOFILE(REPORTS)
```

### Override 2

```
OVRPRTF FILE(REPORTS) COPIES(3)
```

Only one override is applied for each call level for an open of a particular file, so if you want to override the file that is used by the program and also override the attributes of the overriding file from one call level, you must use a single command. If two overrides are used, the first override causes the output to be printed using the REPORTS printer file, but the second override is ignored.

## Generic override for printer files:

The OVRPRTF command enables you to have one override for all the printer files in your job with the same set of values. Without the generic override, you must do a separate override for each of the printer files.

By specifying \*PRTF as the file name on the OVRPRTF command, you can apply one override to all printer files.

The override specified on the OVRPRTF command with \*PRTF is applied if there is no other override for the printer file name at the same call level. The following example shows how \*PRTF works:

### Override 1

```
OVRPRTF FILE(OUTPUT) COPIES(6) LPI(6)
```

### Override 2

```
OVRPRTF FILE(*PRTF) COPIES(1) LPI(8)
```

```
CALL PGM(X)
```

When program X opens the file named OUTPUT, the opened file has the following attributes:

### COPIES(6)

From Override 1

LPI(6) From Override 1



When program X opens the file named PRTOUT, the opened file has the following attributes:

**COPIES(1)**

From Override 2

**LPI(8)** From Override 2

### Deleting printer file overrides

If you want to delete an override before the program has completed running, you can use the Delete Override (DLTOVR) command. This command deletes only overrides that are active in the call level in which the command is entered. To identify an override, use the printer file name specified on the FILE parameter of the override command.

The following example shows an override of a printer file (PRTF1) with a different printer file (PRTF2). On the second line of the example is the Delete Override (DLTOVR) command that is used to delete the override. The printer file named PRTF1 is used to process the output from the application program.

```
OVRPRTF FILE(PRTF1) TOFILE(PRTF2)
DLTOVR FILE(PRTF1)
```

You can delete all printer file overrides at this call level by specifying \*ALL for the FILE parameter.

The following example shows two printer file overrides, PRTC and PRT3. The override is changing the COPIES parameter value in both cases. The third line of the example shows the Delete Override (DLTOVR) command with the FILE parameter value set to \*ALL. This means that the overrides to PRTC and PRT3 are deleted.

```
OVRPRTF FILE(PRTC) COPIES(2)
OVRPRTF FILE(PRT3) COPIES(4)
DLTOVR FILE(*ALL)
```

**Related information:**

Delete Override (DLTOVR) command

### Displaying printer file overrides

You can use the Display Override (DSPOVR) command to display printer file overrides. You can display all printer file overrides or overrides for a specific printer file.

To display any overrides to the printer file PRTF1, type:

```
DSPOVR FILE(PRTF1)
```

To display all printer file overrides, type:

```
DSPOVR FILE(*ALL)
```

However, when the FILE parameter has a value of \*ALL, the system presents a display with all the active overrides to all the file types supported on the system. Using FILE(\*ALL) on the DSPOVR command is an indirect way to learn what overrides are active for any printer files; a more direct way is to issue a DSPOVR command for each printer file.

**Related information:**

Display Override (DSPOVR) command

## Managing printers

Printer management tasks include checking the status of a printer, making printer available, and making printer unavailable.

### Checking the status of a printer

To check the status of a printer, follow these steps.

1. Expand **Basic Operations**.

2. Click **Printers**. The printers and their status are displayed.
3. Right-click the printer for a menu of actions.

### **Making a printer available**

To make the printer available (vary on the printer), follow these steps.

1. Expand **Basic Operations**.
2. Click **Printers**.
3. Right-click the printer that you want to make available.
4. Click **Make Available**.

### **Making a printer unavailable**

To make the printer unavailable (vary off the printer), follow these steps.

1. Expand **Basic Operations**.
2. Click **Printers**.
3. Right-click the printer that you want to make unavailable.
4. Click **Make Unavailable**.

## **Managing printer writers**

Printer writer management tasks include starting a printer writer, stopping a printer writer, and checking the status of a printer writer.

### **Starting a printer writer**

To start the printer writer, follow these steps.

1. Ensure that the status of the printer is **Available**.
  - a. See Check the status of a printer to determine how to check the status.
  - b. If the printer is not available, make the printer available. See Make printer available.
2. Expand **Basic Operations**.
3. Click **Printers**.
4. Right-click the printer for the writer that you want to start.
5. Click **Start**.

#### **Related tasks:**

“Checking the status of a printer” on page 225

To check the status of a printer, follow these steps.

“Making a printer available”

To make the printer available (vary on the printer), follow these steps.

### **Stopping a printer writer**

To stop the printer writer, follow these steps.

1. Expand **Basic Operations**.
2. Click **Printers**.
3. Right-click the printer for the writer that you want to stop.
4. Click **Stop**.

### **Checking the status of a printer writer**

You can use the Work with Writer (WRKWTR) command to check the status of a printer writer.

For example, enter the following command to check the status of PRT01:

```
WRKWTR *DEV PRT01
```

To find a printer writer's job log, whether it is currently active or it has ended, use the Work with Job (WRKJOB) command. For example, enter the following command to display the jobs for PRT01:

```
WRKJOB PRT01
```

If there is more than one job on the system that corresponds to the printer writer, select the job that you want and press Enter. Once you have selected a job use Option 10 to display the job log of a currently running job or Option 4 to work with the spooled files of a job that has ended.

**Related information:**

Work with Writer (WRKWTR) command

## Managing remote writers

Remote writer management tasks include displaying remote writer status, starting remote writer, and stopping remote writer.

### Displaying remote writer status

To display the writers, use the Work with All Writers (WRKWTR) command.

For example, enter the following command:

```
WRKWTR WTR(*ALL)
```

**Note:** Do not use option 1 (Start) to start a remote writer. Use the Start Remote Writer (STRRMTWTR) command to start the remote writer. For example, enter the following command to start RMTWTR01:

```
STRRMTWTR OUTQ(RMTWTR01)
```

**Related information:**

Work with All Writers (WRKWTR) command

Start Remote Writer (STRRMTWTR) command

### Starting a remote writer

To start the remote writer, use the Start Remote Writer (STRRMTWTR) command to start the remote writer.

For example, enter the following to start RMTWTR01:

```
STRRMTWTR OUTQ(RMTWTR01)
```

**Related information:**

Start Remote Writer (STRRMTWTR) command

### Stopping a remote writer

To stop the remote writer, use the End Writer (ENDWTR) command.

For example, enter the following command to stop RMTWTR01:

```
ENDWTR OUTQ(RMTWTR01)
```

**Related information:**

End Writer (ENDWTR) command

## Managing IPP server

You can administer the Internet Printing Protocol (IPP) server using the IBM IPP Server Administrator for iSeries, a browser-driven graphical user interface. Before using the interface, you must configure it.

Using the Manage IBM IPP Server form, you can use the start, stop, and restart buttons to perform the given action for the IPP server. The current server status, along with the ports the server is listening on, is displayed in the table. At most, the IPP server can listen on two ports at any one time. Port 631 is for

nonsecure data and secure data provided by a connection upgrade to TLS. The other port is for secure data provided by an SSL connection. The last time the table information was updated is also displayed. The table can be refreshed by clicking the refresh button.

To access the IBM IPP Server Administration interface, follow these steps:

1. Use the following URL to access the iSeries Tasks page:

`http://system:2001`

where *system* is the name of the system you are working with.

2. On the iSeries Tasks page, click the icon for the IBM IPP Server.

The IBM IPP Server Administrator interface is shown. To use the administrator interface, you must have \*IOSYSCFG authority, and you must have read and write access to the following property files:

- QIBM/UserData/OS400/Ipp/conf/qippsvr-cust.conf
- QIBM/UserData/OS400/Ipp/conf/printer.properties

In order to change preferences, you must have read and write access to the preference property file, QIBM/UserData/OS400/Ipp/conf/preferences.properties, if it exists.

#### Related concepts:

“Setting up the IPP server” on page 127

You can administer and configure the Internet Printing Protocol (IPP) server using the IBM IPP Server Administrator for iSeries, a browser-driven graphical user interface.

### Starting the IPP server

To start the IPP server, follow these steps.

1. Click the **Administration** tab.
2. Click **Manage IBM IPP Server**.
3. Click **Start**.

### Stopping the IPP server

To stop the IPP server, follow these steps.

1. Click the **Administration** tab.
2. Click **Manage IBM IPP Server**.
3. Click **Stop**.

### Changing an IPP printer configuration

To change an IPP printer configuration, follow these steps.

1. Click the **Configuration** tab.
2. Expand the **Internet Printers** menu item.
3. Click **Change configuration** to start the change task wizard.
4. Follow the task steps to change an IPP printer configuration.
5. Click the **Finished** button on the confirmation panel to change the configuration.

### Displaying an IPP printer configuration

To view all of the IPP printers configured on the system, follow these steps.

1. Click the **Configuration** tab to load the configuration navigational menu.
2. Expand the **Internet Printers** menu item.
3. Click **Display Configurations**.

### Deleting an IPP printer configuration

To delete an IPP printer configuration, follow these steps.

1. Click the **Configuration** tab.

2. Expand the **Internet Printers** menu item.
3. Click **Delete configuration** to display a list of configured IPP printers.
4. Select the IPP printer to delete.
5. Click the **Delete** button.

## Miscellaneous printing tasks

Here are some miscellaneous tasks for printing.

### Allowing direct printing

Direct printing is when the \*SPOOL parameter in the printer file is set to \*NO and output data is written directly to the printer.

A printer can be shared between a printer writer and direct print jobs. To allow direct print jobs as well as spooled files for a printer, specify \*YES on the ALWDRTPT parameter of the Start Printer Writer (STRPRTWTR) command. This causes the printer writer to release the printer for direct print jobs when either of the following conditions is true:

- The printer writer is held.
- The output queue associated with the printer contains no spooled files waiting to print.

When using direct printing, you must also be concerned with the WAITFILE parameter in the printer file. The WAITFILE parameter specifies how long a direct print job should wait if the printer is busy before being canceled. The default value for the WAITFILE parameter is \*IMMED. It is recommended that the WAITFILE parameter be set to 2 or 3 minutes. If you set the value of the WAITFILE too large, you can tie up your interactive session while waiting for the job to print.

The STRPRTWTR command can be modified to allow all printers being started to accept direct print jobs. Use the Change Command Default (CHGCMDDFT) command to set the value of the value of the ALWDRTPT parameter to \*YES. Thereafter, when the STRPRTWTR command is used, direct printing is enabled.

The Change Writer (CHGWTR) command cannot be used to enable direct printing.

Direct printing is not supported on printers configured for Advanced Function Presentation (AFP).

#### Related concepts:

“Direct print considerations” on page 101

When output data is written directly to the printer instead of the output queue, the job is called a direct print job. When you print a direct print job on an SCS printer, the IBM i operating system changes the file device type to \*SCS. When you print on an IPDS printer that is configured with AFP(\*NO), the operating system changes the file device type to \*IPDS. If the print job has the file device type \*AFPDS, \*USERASCII, \*LINE, or \*AFPDSLIN, the direct print job is not supported.

#### Related information:

Start Printer Writer (STRPRTWTR) command

Change Command Default (CHGCMDDFT) command

### Specifying separator pages

Separator pages are pages that print at the beginning of the printed output.

Separator pages for print jobs (job separators) are specified by the SEPPAGE parameter on the output queue. For each job having spooled files on the output queue, the specified number of separator pages print at the beginning of the printed output for each job.

Separator pages for spooled files (file separators) are specified in the printer file or the printer writer. The specified number of separator pages print before each spooled file.

Customized separator pages can be configured through the use of an exit program. You can specify an exit program on the SEPPGM parameter of the printer device description. A sample exit program (sourced in C and RPG) can be found in the QUSRTOOL library. For information about this sample exit program, see member TBSINFO in file QATTINFO in library QUSRTOOL.

## Specifying print text

A line of text that is printed at the bottom of every page is called *print text*. Print text is set by with the PRTTXT parameter on the Create Printer File (CRTPRTF), Change Printer File (CHGPRTF), or Override Printer File (OVRPRTF) command.

Up to 30 characters are allowed in the line of print text. The 30 characters are centered at the bottom of the page, 2 lines below the overflow line. If the user already has data to print on the line that the print text goes on, the print text is bumped down to the next blank line on the page. If no lines are blank, the print text is printed on the last line of the page.

### Notes:

1. For externally described printer files with DEVTYPE(\*AFPDS) using the DDS POSITION keyword, the print text is positioned by ignoring the location of any data placed on the page by records using the POSITION keyword. If all the data on the page is positioned using the DDS POSITION keyword, the print text is located on the overflow line.
2. If a host resident font is specified on the printer file, 10 characters per inch is used to calculate the location of the text specified on the PRTTXT parameter.

A system value, Printed page footer (QPRTTXT), can be used to specify the print text so that the same text can appear on all files printed on the system. Also, the print text can be taken from the job description so that all files created from a particular job can have the same print text.

Print text is useful for printing a security classification on each page. It can also be used to print a company name or slogan on each page.

## Replacing unprintable characters

You can replace unprintable characters in your data before the data is written to the printer or to a spooled file by specifying RPLUNPRT(\*YES) on the Create Printer File (CRTPRTF), Change Printer File (CHGPRTF), or Override Printer File (OVRPRTF) command. The replacement of an unprintable character depends on the printer being used and the hexadecimal value of the unprintable character.

The RPLUNPRT value must be selected before the spooled file is created. After a spooled file is in an output queue, changing the RPLUNPRT value has no effect on that particular spooled file.

- When RPLUNPRT(\*YES) is specified, any characters in the range hex 00 through hex 3F, and hex FF, are replaced. The default replacement character is a blank. The character that a printer cannot print varies depending on the type of printer.
- When RPLUNPRT(\*NO) is specified, no translation of the data stream is made. Any characters in the range hex 00 through hex 3F, or hex FF, might cause undesirable results. These characters are in the range used by printer control characters.

For most characters in this range, the printer signals an unrecoverable error and the spooled file either is held in the output queue or is not processed. Some characters in this range control forms action and character representation on the printer and, as a result, additional skipping or spacing might occur. If control characters are placed in the data, system functions, such as displaying or copying spooled files and restarting or backing up a printer, might produce results that cannot be predicted.

If the hexadecimal value of the unprintable character is hex 40 through hex FE, a message is sent to the message queue associated with the printer. The message gives you the option to end the writer, hold the spooled file, ignore the error and continue printing, or select a page number where printing should be restarted. If the ignore option is taken, then unprintable characters continue to be reported. If the option to start again (specify page number) is taken, all unprintable characters are replaced with blanks and you receive no more notification for unprintable characters.

## Considerations for the 4245, 5262, 6252, and 6262 printers

The printer translates lowercase characters to uppercase characters when using a print band that does not contain lowercase characters. If your print job contains other characters that are not on the print band, they can be translated to blanks by specifying RPLUNPRT(\*YES) for the printer file.

A print band is selected by switches on the 5262 operator's panel. The operator must select both a language ID and a band image using these switches.

The 4245, 6252, and 6262 printers detect the print band the printer is using.

When a print band changes for a print job, no inquiry message is sent to the message queue associated with the printer writer. You can specify a different form type for that job, such as blank payroll forms or blank invoices. A message to change the form type is sent to the printer writer message queue to notify the operator of the change to the print band.

## Printing a graphic with other output

The command word #`$$@INCLGRPH` allows a user to include a graphic anywhere in the data printed by high-level languages.

To include graphics with other program output, a special control record is used. The format for this control record is:

```
#$$@INCLGRPH filename,x,y,w,l
```

### Notes:

- There must be only one space between the command word and the parameters.
- All five parameters must be specified. A default value (the default values are included in the list below) for parameters *x*, *y*, *w*, and *l* can be obtained by omitting any value. For example, a valid control record with defaults for parameters *x*, *y*, and *l* looks like:  

```
#$$@INCLGRPH filename,,9.5,
```
- The #`$$@INCLGRPH` control record should be in a print record by itself, because any other data along with it might be considered as parameters.
- Parameters should immediately follow one another, separated by commas, using no blanks.
- The characters INCLGRPH must be all uppercase.
- The #`$$@INCLGRPH` control record must begin in the first column.
- The #`$$@INCLGRPH` control word is used with characters from code page 500. For example, in code page 500, @ is hex '7B', \$ is hex '5B', and @ is hex '7C'. Other code pages might use other characters in the #`$$@INCLGRPH` control record. You will need to change the characters depending on the code page being used.

The parameters are defined as follows. Parameters *x*, *y*, *w*, and *l* define the area on the page where the graphics file is printed. Parameters *x* and *y* define the upper-left corner of the graphics area, and parameters *w* and *l* define the size of the graphics area.

### filename

The name of the graph object file to be included. If the file has more than one member, the last member is used. The library containing the file must be in your library list.

- x** The distance, in inches, from the left edge of the page to the left edge of the graphics area on the page. The default is 0.
- y** The distance, in inches, from the top of the page to the top edge of the graphics area on the page. The default is 0.
- w** The width of the graphics area, in inches. The default is the width of the current page you are using.

- 1 The length of the graphics area, in inches. The default is the length of the current page you are using.

Parameters  $x$ ,  $y$ ,  $w$ , and  $l$  can be specified in decimal form in any combination of  $xx.xx$ , where  $x$  is any number from 0 through 9. The specified value cannot be more than 45.50 and, if a 0 value is specified, the result is the default value for that parameter.

If there are any errors found in the control record or an error occurs while processing the graphics file, the control record is printed as normal text data. The graphics file to be used must be in a format acceptable to the printer. For IPDS devices, this format is level DR/2 of the Graphic Object Content Architecture (GOCA). See the manual *Graphic Object Content Architecture*, SC31-6804, for more information about GOCA.

## Displaying graphic symbol sets

IPDS printers allow the selection of graphic symbol sets as fonts when using the DDS font keyword.

To find which graphic symbol sets are available on your system, enter the CL command:

```
DSPOBJD OBJTYPE(*GSS) OBJ(QGDDM/*ALL)
```

## Changing remote output queue description

To change a remote output queue description, follow these steps.

1. Use the End Writer (ENDWTR) command to stop the remote writer. For example, enter the following to stop RMTWTR01:

```
ENDWTR OUTQ(RMTWTR01) OPTION(*IMMED)
```

2. Use the Change Output Queue (CHGOUTQ) command to change the output queue. For example, enter the following to change RMTWTR01:

```
CHGOUTQ OUTQ(RMTWTR01)
```

3. Press F9 to display all of the parameters.

4. Make the changes and press Enter.

5. Use the Start Remote Writer (STRRMTWTR) command to start the remote writer. For example, enter the following to start RMTWTR01:

```
STRRMTWTR OUTQ(RMTWTR01)
```

### Related information:

End Writer (ENDWTR) command

Change Output Queue (CHGOUTQ) command

Start Remote Writer (STRRMTWTR) command

---

## Examples: Using DDS and printer files

Here are some examples of using data description specifications and printer files.

## Examples: Using DDS with RPG or COBOL

These examples provide instructions and example code to illustrate how to use data description specifications (DDS) with RPG or COBOL.

**Note:** By using the code examples, you agree to the terms of the “Code license and disclaimer information” on page 399.

### Example: DDS row and column positioning source code

The row and column method of positioning means specifying where the data starts printing (how many rows down and how many columns in).



**Note:** By using the code examples, you agree to the terms of the “Code license and disclaimer information” on page 399.

The following data description specifications source coding uses the row and column method of positioning.

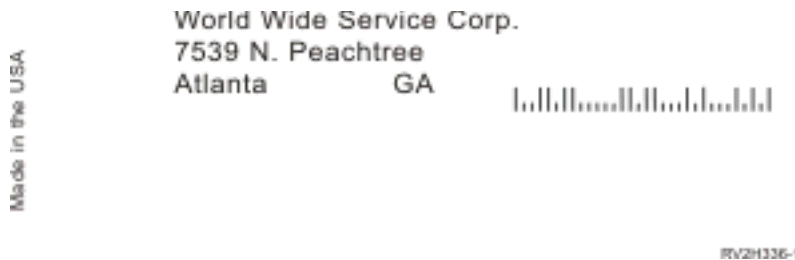
```

000100911101          R LABEL
000200911101          NAME          25A 0 8 10
000300911101          ADDR1         25A 0 9 10
000400911101          CITY          15A 0 10 10
000500911101          STATE         2A 0 10 27
000600911101          ZIPCD         5S 00 10 30BARCODE(POSTNET *HRITOP)
000700911101                               13 5'Made in the USA'
000800911101                               TXTRTT(270)
000900911101                               FONT(5687 (*POINTSIZ 6))

```

For example, at row 10 column 30, a bar code is printed.

The following figure shows the output achieved using the row and column positioning method.



### Example: DDS absolute positioning source code

Absolute positioning means being able to start printing at any point on a piece of paper by specifying that point.

**Note:** By using the code examples, you agree to the terms of the “Code license and disclaimer information” on page 399.

Use of absolute positioning requires the Advanced Function Presentation data stream. This is obtained on i5/OS by specifying \*AFPDS on the device type (DEVTYPE) parameter of the printer file.

Inches or centimeters are the measurement methods available. You choose the measurement method by specifying \*INCH or \*CENT on the unit of measure (UOM) parameter of the printer file.

The following data description specifications coding uses the absolute method of positioning.

```

000100911101          R LABEL          PAGSEG(LOGO 1.5 1)
000200911101          BOX(1 1 5 4 *MEDIUM)
000300911101          BOX(1.5 4 2 4.5 *NARROW)
000400911101          NAME          25A 0    POSITION(1.3 1.6)
000500911101          ADDR1         25A 0    POSITION(1.5 1.6)
000600911101          CITY          15A 0    POSITION(1.7 1.6)
000700911101          STATE         2A 0    POSITION(1.7 2.7)
000800911101          ZIPCD         5S 00    POSITION(1.7 3)
000900911101          BARCODE(POSTNET *HRITOP)
f01000911101          TEXT          20A 0    TXTRTT(270)
000800911101          POSITION(1.9 .25)
001100911101          FONT(5687 (*POINTSIZ 6))

```

In this example, a page segment called LOGO is specified to start printing at 1.5 units down and 1 unit in. The TEXT record (Made in the USA) is supplied by the application program. The following figure shows the output achieved using absolute positioning.



### Example: COBOL positioning source code

Here is an example of positioning source code in COBOL.

**Note:** By using the code examples, you agree to the terms of the “Code license and disclaimer information” on page 399.

```

STMT SEQNBR -A 1 B..+...2....+...3....+...4....+...5....+...6....+...7..
IDENTFCN  S  COPYNAME                                     CHG DATE
 1 000100 IDENTIFICATION DIVISION.                       10/20/91
 2 000200 PROGRAM-ID.  CBLLBL.                            10/20/91
 3 000300 ENVIRONMENT DIVISION.                          10/20/91
 4 000400 INPUT-OUTPUT SECTION.                          10/20/91
 5 000500 FILE-CONTROL.                                  10/20/91
 6 000600     SELECT PRINTER-FILE                        10/20/91
 7 000700         ASSIGN TO FORMATFILE-LABELS.          10/23/91
 8 000800     SELECT VENDOR-FILE                        10/20/91
 9 000900         ASSIGN TO DATABASE-VENDORS.           10/20/91
10 001000 DATA DIVISION.                                10/20/91
11 001100 FILE SECTION.                                  10/20/91
12 001200 FD  PRINTER-FILE                               10/20/91
13 001300     DATA RECORD IS PRINT-REC.                 10/23/91
14 001400 01  PRINT-REC.                                 10/23/91
15 001500     COPY DDS-ALL-FORMATS-0 OF LABELS.          10/23/91
16 +000001     05  LABELS-RECORD PIC X(72).              <-ALL-FMTS
    +000002*  OUTPUT FORMAT;LABEL  FROM FILE LABELS    OF LIBRARY SGAFP <-ALL-FMTS
    +000003*                                     <-ALL-FMTS
17 +000004     05  LABEL-0      REDEFINES LABELS-RECORD. <-ALL-FMTS
18 +000005         06  NAME          PIC X(25).          <-ALL-FMTS
19 +000006         06  ADDR1        PIC X(25).          <-ALL-FMTS
20 +000007         06  CITY         PIC X(15).          <-ALL-FMTS
21 +000008         06  STATE        PIC X(2).           <-ALL-FMTS
22 +000009         06  ZIPCD        PIC S9(5).          <-ALL-FMTS
23 001600 FD  VENDOR-FILE                               10/23/91
24 001700     DATA RECORD IS VENDOR-REC.                10/23/91
25 001800 01  VENDOR-REC.                                10/23/91
26 001900     COPY DDS-ALL-FORMATS-I OF VENDORS.         10/23/91
27 +000001     05  VENDORS-RECORD PIC X(82).            <-ALL-FMTS
    +000002*  I-O FORMAT;VNDMSTR  FROM FILE VENDORS  OF LIBRARY SGAFP <-ALL-FMTS
    +000003*                                     VENDMAST DB FORMAT <-ALL-FMTS
28 +000004     05  VNDMSTR      REDEFINES VENDORS-RECORD. <-ALL-FMTS
29 +000005         06  VNDNBR        PIC S9(5)          COMP-3. <-ALL-FMTS
    +000006*                                     VENDOR NUMBER <-ALL-FMTS

```

```

30 +000007      06 NAME          PIC X(25).          <-ALL-FMTS
+000008*      NAME          <-ALL-FMTS
31 +000009      06 ADDR1        PIC X(25).          <-ALL-FMTS
+000010*      ADDRESS LINE 1    <-ALL-FMTS
32 +000011      06 CITY          PIC X(15).          <-ALL-FMTS
+000012*      CITY          <-ALL-FMTS
33 +000013      06 STATE        PIC X(2).           <-ALL-FMTS
+000014*      STATE         <-ALL-FMTS
34 +000015      06 ZIPCD        PIC S9(5)          COMP-3.    <-ALL-FMTS
+000016*      ZIP CODE       <-ALL-FMTS
35 +000017      06 VNDCLS       PIC S9(2)          COMP-3.    <-ALL-FMTS
+000018*      VENDOR CLASS   <-ALL-FMTS
36 +000019      06 VNDSTS       PIC X(1).           <-ALL-FMTS
+000020*      A=ACTIVE, D=DELETE, S=SUSPEND <-ALL-FMTS
37 +000021      06 BALOWE       PIC S9(7)V9(2)     COMP-3.    <-ALL-FMTS
+000022*      BALANCE OWED   <-ALL-FMTS
38 +000023      06 SRVRTG       PIC X(1).           <-ALL-FMTS
+000024*      G=GOOD, A=AVERAGE, B=BAD, P=PREFERRED<-ALL-FMTS
39 002000 WORKING-STORAGE SECTION.                10/20/91
40 002100 77 EOF-FLAG          PIC X.              10/23/91
41 002200 88 NOT-END-OF-FILE   VALUE " ".         10/23/91
5738CBI V2R1M0 910524
AS/400 COBOL Source SGAFP/CBLLBL RCHASA12 10/24/91 10:18:16 Page 3
STMT SEQNBR -A 1 B.+....2....+....3....+....4....+....5....+....6....+....7..
IDENTFCN S COPYNAME          CHG DATE
42 002300 88 END-OF-FILE      VALUE "1".         10/23/91
002400          10/23/91
43 002500 PROCEDURE DIVISION. 10/23/91
002600 MAIN-PARA.           10/23/91
44 002700 OPEN INPUT VENDOR-FILE 10/23/91
002800 OUTPUT PRINTER-FILE. 10/23/91
45 002900 PERFORM PRINT-LABELS UNTIL END-OF-FILE. 10/23/91
46 003000 CLOSE VENDOR-FILE, PRINTER-FILE. 10/23/91
003100 GOBACK.             10/23/91
003200          10/23/91
47 003300 PRINT-LABELS.       10/23/91
48 003400 READ VENDOR-FILE     10/23/91
49 003500 AT END SET END-OF-FILE TO TRUE. 10/23/91
50 003600 IF NOT-END-OF-FILE   10/23/91
51 003700 MOVE CORRESPONDING VNDMSTR TO LABEL-0 10/23/91
* ** CORRESPONDING items for statement 51:
* ** NAME
* ** ADDR1
* ** CITY
* ** STATE
* ** ZIPCD
* ** End of CORRESPONDING items for statement 51
52 003800 WRITE PRINT-REC FORMAT IS "LABEL". 10/23/91
* * * * * E N D O F S O U R C E * * * * *

```

### Example: RPG positioning source code

Here is an example of positioning source code in RPG.

**Note:** By using the code examples, you agree to the terms of the "Code license and disclaimer information" on page 399.

```

SEQUENCE      IND  DO  LAST  PAGE  PROGRAM
NUMBER *...1....+....2....+....3....+....4....+....5....+....6....+....7....
USE  NUM  UPDATE  LINE  ID
          S o u r c e   L i s t i n g
H          *****
100 FVENDORS IF E          K          DISK          10/24/91
    RECORD FORMAT(S): LIBRARY SGAFP FILE VENDORS.
    EXTERNAL FORMAT VNDMSTR RPG NAME VNDMSTR
200 FLABELS 0 E          PRINTER          08/25/91
    RECORD FORMAT(S): LIBRARY SGAFP FILE LABELS.

```

```

                EXTERNAL FORMAT LABEL RPG NAME LABEL
A000000 INPUT FIELDS FOR RECORD VNDMSTR FILE VENDORS FORMAT VNDMSTR.
A000000 VENDMAST DB FORMAT
A000001          P 1 30VNDNBR  VENDOR NUMBER
A000002          4 28 NAME     NAME
A000003          29 53 ADDR1   ADDRESS LINE 1
A000004          54 68 CITY    CITY
A000005          69 70 STATE   STATE
A000006          P 71 730ZIPCD  ZIP CODE
A000007          P 74 750VNDCLS  VENDOR CLASS
A000008          76 76 VNDSTS  A=ACTIVE, D=DELETE, S=SUSPEND
A000009          P 77 812BALOWE  BALANCE OWED
A000010          82 82 SRVRTG   G=GOOD, A=AVERAGE, B=BAD, P=PREFERRED
    300 C          READ VENDORS          50 3 10/24/91
    400 C          *IN50  DOWNE*ON          B001 08/25/91
    500 C          WRITELABEL          001 08/25/91
    600 C          READ VENDORS          50 3 10/24/91
    700 C          ENDDO          E001 08/25/91
    800 C          SETON          LR 1 08/25/91
B000000 OUTPUT FIELDS FOR RECORD LABEL FILE LABELS FORMAT LABEL.
B000001          NAME 25 CHAR 25
B000002          ADDR1 50 CHAR 25
B000003          CITY 65 CHAR 15
B000004          STATE 67 CHAR 2
B000005          ZIPCD 72 ZONE 5,0
* * * * * E N D O F S O U R C E * * * * *

```

## Example: Print font samples

These examples contain instructions and source code that allow you to print a font global identifier (FGID), font character set, or coded font and to see what it looks like.

The source code provided is data description specifications (DDS), which can be used with the following high-level languages:

- C
- COBOL
- RPG

### Notes:

1. By using the code examples, you agree to the terms of the “Code license and disclaimer information” on page 399.
2. These sample programs work only with printers configured as AFP(\*YES).
3. See AFP compatibility fonts and font substitution for the names of FGIDs, font character sets, and coded fonts.

Use the following instructions to print a font global identifier (FGID), font character set, or coded font and see what it looks like. If you need assistance with any of the CL commands, use the F4 (Prompt) key and then press the Help key on any of the parameters.

1. Create a library to contain the objects needed to print the font samples. In this example, the library is named FONTSAMPLE.  
CRTLIB FONTSAMPLE
2. Add FONTSAMPLE to your library list.  
ADDLIB FONTSAMPLE
3. Create a source physical file in FONTSAMPLE to contain your source code. In this example the source file is named SOURCE.  
CRTSRCPF FONTSAMPLE/SOURCE
4. Add a member named FONT to that physical file. This member is used for entering the source code for the printer file.

ADDPFM FILE(FONTSAMPLE/SOURCE) MBR(FONT)

5. Edit the member FONT with the source entry utility (SEU).

STRSEU SRCFILE(FONTSAMPLE/SOURCE) SRCMBR(FONT) TYPE(PRTF)

Type the DDS source code (shown in “DDS source code” on page 238) for the printer file. Edit the DDS source and insert the correct font identifier or name. When you are done, press F3 to exit.

6. Create the printer file from the DDS source you just typed in.

CRTPRTF FILE(FONTSAMPLE/FONT) SRCFILE(FONTSAMPLE/SOURCE) SRCMBR(FONT)  
DEVTYPE(\*AFPDS)

7. Choose the high-level language you will be using to produce printed output. The compiler for the high-level language you choose must be installed on your system. Samples are given for C, RPG, and COBOL. Add a member to the physical file SOURCE. Use one of the following names depending on which language you choose:

- CCODE for the C language
- COBOLCODE for the COBOL language
- RPGCODE for the RPG language

ADDPFM FILE(FONTSAMPLE/SOURCE) MBR(CCODE, COBOLCODE, or RPGCODE)

8. Edit the member (CCODE, COBOLCODE, or RPGCODE) with source entry utility (SEU). Type the command that corresponds to the high-level language you are using.

**C:** STRSEU SRCFILE(FONTSAMPLE/SOURCE) SRCMBR(CCODE) TYPE(C)

**RPG:** STRSEU SRCFILE(FONTSAMPLE/SOURCE) SRCMBR(RPGCODE) TYPE(RPG)

**COBOL:**

STRSEU SRCFILE(FONTSAMPLE/SOURCE) SRCMBR(COBOLCODE) TYPE(CBL)

9. Type the program source for one of these languages:

- C source code
- RPG source code
- COBOL source code

When you are done, press F3 to exit.

10. Create the program, using the command below that corresponds to the language you chose:

**C:** CRTCPGM PGM(FONTSAMPLE/CPGM) SRCFILE(FONTSAMPLE/SOURCE) SRCMBR(CCODE)

**RPG:** CRTRPGPM PGM(FONTSAMPLE/RPGPM) SRCFILE(FONTSAMPLE/SOURCE) SRCMBR(RPGCODE)

**COBOL:**

CRTCBLPGM PGM(FONTSAMPLE/CBLPGM) SRCFILE(FONTSAMPLE/SOURCE) SRCMBR(COBOLCODE)

11. Call the program that corresponds to the language you chose:

**C:** Call CPGM

**RPG:** Call RPGPM

**COBOL:**

Call CBLPGM

The output from the program is sent to a spooled file named FONT. You cannot view the spooled file until it prints.

#### **Related reference:**

“AFP compatibility fonts and font substitution” on page 306

This section contains information about Advanced Function Presentation (AFP) compatibility fonts, character identifiers and other printing characteristics as well as several tables to help you understand font substitution.

## DDS source code

Here is the data description specifications (DDS) source code mentioned Example: Print font samples.

**Note:** By using the code examples, you agree to the terms of the “Code license and disclaimer information” on page 399.

```
5738PW1 V2R2M0 920615          SEU SOURCE LISTING
02/09/93 13:56:16                PAGE      1
SOURCE FILE . . . . . FONTSAMPLE/SOURCE
MEMBER . . . . . FONT
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7
...+... 8 ...+... 9 ...+... 0
100                                R REC1
200                                6 10FONT(5)
300                                'Rhetoric Orator FGID 5'
400                                8 10FNTCHRSET(QFNT01/C0T055B0 +
500                                QFNT01/T1V10037)
600                                'Sonoran Serif 12 Pt +
700                                Font Char Set C0T055B0'
800                                10 10CDEFNT(QFNTCPL/X0BIR1)
900                                'Book Italic 10 Pt +
1000                               CODED FONT X0BIR1'
1100                               * Above is the source for DDS and the fontsample program
*** * E N D O F S O U R C E * * * *
```

## C source code

Here is the C source code mentioned in Example: Print font samples.

**Note:** By using the code examples, you agree to the terms of the “Code license and disclaimer information” on page 399.

```
5738PW1 V2R2M0 920615          SEU SOURCE LISTING
SOURCE FILE . . . . . FONTSAMPLE/SOURCE
MEMBER . . . . . CCODE
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7
...+... 8 ...+... 9 ...+... 0
100 #include <stdio.h>
200 #include <xxasio.h>
300 main()
400 {
500 FILE          *outfile;
600
700 outfile = fopen("font","wb type=record");
800 QXXFORMAT(outfile, "REC1      ");
900 fwrite("",0,0,outfile);
1000 }
*** * E N D O F S O U R C E * * * *
```

## COBOL source code

Here is the COBOL source code mentioned in Example: Print font samples.

**Note:** By using the code examples, you agree to the terms of the “Code license and disclaimer information” on page 399.

```
5738PW1 V2R2M0 920615          SEU SOURCE LISTING
SOURCE FILE . . . . . FONTSAMPLE/SOURCE
MEMBER . . . . . COBOLCODE
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7
...+... 8 ...+... 9 ...+... 0
100 100010 IDENTIFICATION DIVISION.
200
300 100020 PROGRAM-ID. PRINTLBL.
400
500 100030 ENVIRONMENT DIVISION.
600
700 100040 INPUT-OUTPUT SECTION.
```

```

800
900 100050 FILE-CONTROL.
1000
1100 100060     SELECT PRINTER-FILE
1200
1300 100070     ASSIGN TO FORMATFILE-FONT.
1400
1500 100080 DATA DIVISION.
1600
1700 100090 FILE SECTION.
1800
1900 100100 FD PRINTER-FILE
2000
2100 100110     DATA RECORD IS REC1.
2200
2300 100120 01 REC1.
2400
2500 100130     COPY DDS-ALL-FORMATS OF FONT.
2600
2700 100140 PROCEDURE DIVISION.
2800
2900 100150 MAIN-PARA.
3000
3100 100160     OPEN OUTPUT PRINTER-FILE.
3200
3300 100170     WRITE REC1 FORMAT IS "REC1".
3400
3500 100180     CLOSE PRINTER-FILE.
3600
* * * * E N D O F S O U R C E * * * *

```

## RPG source code

Here is the RPG source code mentioned in Example: Print font samples.

**Note:** By using the code examples, you agree to the terms of the “Code license and disclaimer information” on page 399.

```

5738PW1 V2R2M0 920615          SEU SOURCE LISTING
SOURCE FILE . . . . . FONTSAMPLE/SOURCE
MEMBER . . . . . RPGCODE
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7
...+... 8 ...+... 9 ...+... 0
100     FFONT    0   E          PRINTER
200     C          WRITEREC1
300     C          SETON          LR
* * * * E N D O F S O U R C E * * * *

```

## Example: Using a program-described printer file with an application program

This example shows how an application program calls and uses a program-described printer file to control how your printed output looks.

This example consists of:

- An RPG-coded application program that produces mailing labels.
- A printer file that is opened by the application program when the application program runs.
- A detailed listing of which parameters from the printer file are used by the application program and at what point during the processing of the application program they are used.

This is the RPG-coded application program. The numbers within the program correspond to the list on the following pages. That list explains how this program works and specifically how the program opens and uses the printer file.

- Part (1) opens the printer file.
- Part (2) sends the output to the designated output queue.
- Part (3) closes the opened files.
- Part (4) processes the data.
- Part (5) provides the code to control how the printed output will look.

**Note:** By using the code examples, you agree to the terms of the “Code license and disclaimer information” on page 399.

```
*****
          FADDRESS IF E      K      DISK
(1)  FLABELPRTO  F    132    OF    PRINTER
*****
```

```
          CTAR      20  1
          CSAR      30  1
          STAR      2   1
          ZPAR      8   1
```

```
          READ ADDRESS                      10
```

```
*IN10  DOWEQ '0'
```

```
ADD2   IFEQ *BLANKS
        MOVE '1'      *IN55
        ELSE
        MOVE '0'      *IN55
        END
```

```
          EXSR CKCITY
```

```
*****
(2)          EXCPTPRINT
*****
          READ ADDRESS                      10

          END
```

```
*****
(3)          MOVE '1'      *INLR
*****
```

```
          CKCITY  BEGSR
```

```
          MOVEA*BLANKS  CTAR
          MOVEA*BLANKS  STAR
          MOVEA*BLANKS  ZPAR
          MOVEA*BLANKS  CSAR
```

```
          MOVEACITY     CTAR
          MOVEAST       STAR
          MOVEAZIP      ZPAR
```

```
          Z-ADD1        X      20
          Z-ADD1        Y      20
```

```
          EXSR LOOKBL                      1ST WORD.
          ADD 1                                X
          CTAR,X  IFGT *BLANKS                2ND WORD
          MOVE ' '      CSAR,Y
          ADD 1                                Y
          EXSR LOOKBL
```

```
(4)          ADD 1                                X
          CTAR,X  IFGT *BLANKS                3RD WORD
          MOVE ' '      CSAR,Y
```



```

ADD 1      Y      EXSR LOOKBL
END
END
MOVE ', '  CSAR,Y
ADD 1      Y
MOVE ' '   CSAR,Y
ADD 1      Y
MOVE STAR,1 CSAR,Y
ADD 1      Y

MOVE STAR,2 CSAR,Y
ADD 1      Y
MOVE ' '   CSAR,Y
ADD 1      Y
MOVE ' '   CSAR,Y
ADD 1      Y
Z-ADD1     X
X          DOWLT9
MOVE ZPAR,X CSAR,Y
ADD 1      Y
ADD 1      X
END
MOVEAC SAR CTSTZP 30
ENDSR

LOOKBL     BEGSR
CTAR,X     DOWGT*BLANKS
MOVE CTAR,X CSAR,Y
ADD 1      X
ADD 1      Y
END
ENDSR
*****
RTE 1 2     PRINT
NAME 25
E 1        PRINT
ADD1 25
(5) E 1 N55  PRINT
ADD2 25
E 1        PRINT
CTSTZP 30
E 1 55     PRINT
*****

```

## Open processing

Part **(1)** of the application program opens files that are called by the application program.

When a program opens files, an object of type \*FILE is connected to the program for processing. Among the files opened in this example, and of particular interest at this point, is the printer file whose name is LABELPRT. You can find the printer file name LABELPRT next to the **(1)** in the program listing.

A printer file is opened to prepare the system so that the application can put data into a spooled file or print it out directly to a printer. Information from the high-level language application program, the printer file, and any printer file overrides is combined.

The printer file open operation is controlled by parameters specified in the printer file, the high-level language program, and in printer file overrides (through the Override with Printer File (OVRPRTF) command). See "Printer file overrides" on page 6 for more information about overrides.

As an example, if the printer file specified lines per inch (LPI) of 8, and an OVRPRTF command specified an LPI of 6, the LPI of 6 is used since the override value specified by the OVRPRTF command takes precedence over the LPI value specified in the printer file.

The following list contains parameters from the printer file LABELPRT. These are the parameters that the application program accesses or looks at when it opens the printer file. They are the majority of the parameters in the printer file, but not all of them. When the application looks at each parameter, it finds a value specified for each parameter. For a description of each parameter, see the Create Printer File (CRTPRTF) CL command.

FILE  
DEV  
DEVTYPE  
CVTLINDTA  
PAGESIZE  
LPI  
UOM  
CPI  
OVRFLW  
RPLUNPRT  
FIDELITY  
CTLCHAR  
PRTQLTY  
FORMFEED  
DRAWER  
OUTBIN  
FONT  
CHRID  
DECfmt  
FNTCHRSET  
CDEFNT  
PAGDFN  
FORMDF  
AFPCHARS  
TBLREFCHR  
PAGRIT  
PRTTXX  
JUSTIFY  
DUPLEX  
IPDSPASTHR  
USRRSCLIBL  
CORNERSTPL  
EDGESTITCH  
SADLSTITCH  
FNTRSL  
SPOOL  
SCHEDULE  
USRDTA  
SPLFOWN  
USRDFNOPT  
USRDFNDA  
USRDFNOBJ  
IGCDTA  
IGEXNCR  
IGCCHRTT  
IGCCPI  
IGCSOSI  
IGCCDEFNT  
WAITFILE  
SHARE  
LVLCHK  
AUT  
TEXT

## Output processing

Part (2) of the application program performs the operations of reading, compiling, and sending the output to the output queue specified in the OUTQ parameter of the CRTPRTF command or to the printer specified in the DEV parameter of the CRTPRTF command. In this example, the SPOOL parameter has a value of (\*YES). This means the output will become a spooled file in the designated output queue.

The following printer file parameters are the CRTPRTF parameters looked at by the system program and application program during the output processing portion of the application program. For a description of each parameter, see the CRTPRTF CL command.

This example does not use a DDS source file. Thus, when the application program calls the printer file LABELPRT and looks at the SRCFILE parameter, the value will be \*NONE. Since no DDS will be used, the high-level language must provide the programming to control how the printed output will look. This example contains the high-level language code in part (5) of the example program listing.

```
SRCFILE
SRCMBR
FOLD
ALIGN
CHLVAL
PRTTXT
REDUCE
MULTIUP
FRONTMGN
BACKMGN
FRONTOVL
BACKOVL
MAXRCDS
DFRWRT
```

### Close processing

Part (3) of the application program performs the close operations of the application program.

When the application program has finished the output processing part of the application program, it performs a close operation on all the files it opened during the open processing portion of the application program.

The following printer file parameters from the CRTPRTF command are the parameters looked at by the system program and the application program during the close portion of the application program. For a description of each parameter, see the CRTPRTF CL command.

```
SCHEDULE
```

### Output from the example

```
Ann White
Box 123
RR 1
Anytown, IA 12345
```

```
Tom Smith
123 Main St.
Somewhere, IN 54321
```

### Related information:

Create Printer File (CRTPRTF) command

## Example: Using an externally described printer file with an application program

This example shows how an application program calls and uses an externally described printer file to control how your printed output looks.

This example consists of:

- An RPG-coded application program that produces mailing labels.
- A printer file that is opened by the application program when the application program runs.
- A detailed listing of which parameters from the printer file are used by the application program and at what point during the processing of the application program they are used.
- A detailed listing of the DDS keywords and an explanation of which DDS keywords are used by the mailing label application program.

This is the RPG-coded application program. The numbers within the program correspond to the list on the following pages. That list explains how this program works and specifically how the program opens and uses the printer file.

- Part (1) opens the printer file.
- Part (2) sends the output to the designated output queue.
- Part (3) closes the opened files.
- Part (4) processes the data.

**Note:** By using the code examples, you agree to the terms of the “Code license and disclaimer information” on page 399.

```
*****
                FADDRESS IF E          K          DISK
(1)  FLABELPR30  E                      PRINTER
*****

                CTAR      20  1
                CSAR      30  1
                STAR      2  1
                ZPAR      8  1

                READ ADDRESS                      10
*****
                WRITEHEADNG

                *IN10   DOWEQ'0'

                EXSR CKCITY

(2)                WRITEDetail1

                ADD2    IFNE *BLANKS
                    WRITEDetail3
                    END

                WRITEDetail4
                READ ADDRESS                      10

                END
*****

*****
(3)                MOVE '1'          *INLR
*****
                CKCITY   BEGSR

                MOVEA*BLANKS  CTAR
```

```

                MOVEA*BLANKS STAR
                MOVEA*BLANKS ZPAR
                MOVEA*BLANKS CSAR
                MOVEACITY CTAR
                MOVEAST STAR
                MOVEAZIP ZPAR
                Z-ADD1 X 20
                Z-ADD1 Y 20

                EXSR LOOKBL 1ST WORD
                ADD 1 X
CTAR,X IFGT *BLANKS 2ND WORD
                MOVE ' ' CSAR,Y
                ADD 1 Y
                EXSR LOOKBL
                ADD 1 X
CTAR,X IFGT *BLANKS 3RD WORD
                MOVE ' ' CSAR,Y
                ADD 1 Y
(4) EXSR LOOKBL
                END
                END
                MOVE ', ' CSAR,Y
                ADD 1 Y
                MOVE ' ' CSAR,Y
                ADD 1 Y
                MOVE STAR,1 CSAR,Y
                ADD 1 Y
                MOVE STAR,2 CSAR,Y
                ADD 1 Y
                MOVE ' ' CSAR,Y
                ADD 1 Y
                MOVE ' ' CSAR,Y
                ADD 1 Y
                Z-ADD1 X
X DOWLT9
                MOVE ZPAR,X CSAR,Y
                ADD 1 Y
                ADD 1 X
                END
                MOVEAC SAR CTSTZP 30
                MOVEAC SAR CTSTZ2 30
                ENDSR

LOOKBL BEGSR
CTAR,X DOWGT*BLANKS
                MOVE CTAR,X CSAR,Y
                ADD 1 X
                ADD 1 Y
                END
                ENDSR

```

\*\*\*\*\*

## Open processing

Part (1) of the application program opens files that are called by the application program. Among those, and of particular interest at this point, is the printer file whose name is LABELPR3. You can locate LABELPR3 next to (1) in the program listing.

A printer file is opened to prepare the system so that the application can put data into a spooled file or print it out directly to a printer. Information from the high-level language application program, the printer file, and any printer file overrides is combined.

The printer file open operation is controlled by parameters specified in the printer file, in the high-level language program, and in printer file overrides (through the OVRPRTF command). See "Printer file overrides" on page 6 for more information about overrides.

As an example, if the printer file specified lines per inch (LPI) of 8, and an Override with Printer File (OVRPRTF) command specified an LPI of 6, the LPI of 6 is used since the override value specified by the OVRPRTF command takes precedence over the LPI value specified in the printer file.

The following list contains parameters from the printer file LABELPRT. These are the parameters that the application program accesses or looks at when it opens the printer file. They are the majority of the parameters in the printer file, but not all of them. When the application looks at each parameter, it finds a value specified for each parameter. For a description of each parameter, see the Create Printer File (CRTPRTF) CL command.

FILE  
DEV  
DEVTYPE  
CVTLINDTA  
PAGESIZE  
LPI  
UOM  
CPI  
OVRFLW  
RPLUNPRT  
FIDELITY  
CTLCHAR  
PRTQLTY  
FORMFEED  
DRAWER  
OUTBIN  
FONT  
CHRID  
DECFMT  
FNTCHRSET  
CDEFNT  
PAGDFN  
FORMDF  
AFPCHARS  
TBLREFCHR  
PAGRIT  
PRTTXX  
JUSTIFY  
DUPLEX  
IPDSPASTHR  
USRRSCLIBL  
CORNERSTPL  
EDGESTITCH  
SADLSTITCH  
FNTRSL  
SPOOL  
SCHEDULE  
USRDTA  
SPLFOWN  
USRDFNOPT  
USRDFNDA  
USRDFNOBJ  
IGCDTA  
IGCEXNCR  
IGCCHRTT  
IGCCPI  
IGCSOSI  
IGCCDEFNT  
WAITFILE

SHARE  
LVLCHK  
AUT  
TEXT

## Output processing

Part (2) of the application program performs the operations of reading, compiling, and sending the output to the output queue specified in the OUTQ parameter of the CRTPRTF command, or to the printer specified in the DEV parameter of the CRTPRTF command. In this example, the SPOOL parameter has a value of (\*YES), which means the output will become a spooled file in the designated output queue.

The following printer file parameters from the CRTPRTF command are parameters that are looked at by the application program during the output processing. The DDS is compiled before the application program runs. The application program never looks at the DDS file and member, only at the compiled results.

Since this example uses DDS, look at (1) in the program listing and see that the name of the printer file is LABELPR3. LABELPR3 was compiled using the source from the member and file that are listed here.

SRCFILE  
SRCMBR  
FOLD  
ALIGN  
CHLVAL  
PRTTXT  
REDUCE  
MULTIUP  
FRONTMGN  
BACKMGN  
FRONTOVL  
BACKOVL  
MAXRCDS  
DFRVRT  
OPTION  
GENLVL

## Data description specifications

Below is the example of the compiled DDS used by the RPG program. You can update the DDS; however, you must then recompile it.

```
000100900115          R HEADNG
000200900115                    3  2'MAILING LABELS'
000300900115
000400900115          R DETAIL1
000500900115          NAME          25      2  2UNDERLINE
000600900115          ADD1          25      3  2
000700900115          R DETAIL3
000800900115          ADD2          25          2SPACEB(1)
000900900115          R DETAIL4
001000900115          CTSTZP        30          2HIGHLIGHT SPACEB(1)
```

This example uses three DDS keywords: SPACEB, UNDERLINE, and HIGHLIGHT.

DDS and its associated keywords can only be used if the SRCFILE parameter contains the name of the file and the SRCMBR parameter contains the name of the member that the DDS source resides in.

For a description of DDS keywords for printer files and more detailed information about DDS source files, see DDS Reference: Printer files in the Programming category.

### Close processing

Part (3) of the application program performs the close operations of the application program.

When the application program has finished the output processing part of the application program, it performs a close operation on all the files it opened during the open processing of the application program.

The application program looks at the SCHEDULE parameter from the CRTPRTF command during the close portion of the application program.

### Output from the example

```
Ann White  
Box 123  
RR 1  
Anytown, IA 12345
```

```
Tom Smith  
123 Main St.  
Somewhere, IN 54321
```

---

## Troubleshooting printing

If your printer is not printing, check this list of common reasons.

- The spooled files in the output queue are not in a ready status. For information about how to display a list of spooled files, including the status of the spooled files, see “Displaying a list of spooled files” on page 212.
- The printer might be varied off or might need you to reply to a message. For information about how to display a list of printers, including the status of the printers, see “Checking the status of a printer” on page 225.
- The printer writer is not started, has been held, or has ended. For information about how to display the status of the printer writer, see “Checking the status of a printer writer” on page 226.

For help in troubleshooting the IBM IPP Server, see Troubleshooting the IPP server.

For help in troubleshooting the image print transform, see Troubleshooting the image print transform.

For additional help in troubleshooting printing problems, see InfoPrint Solutions Company : Support & Maintain  ([www.infoprintsolutionscompany.com/internet/wwsites.nsf/vwWebPublished/sm\\_home\\_us](http://www.infoprintsolutionscompany.com/internet/wwsites.nsf/vwWebPublished/sm_home_us)) and Support for IBM System i  ([www.ibm.com/support/](http://www.ibm.com/support/)) Web sites.

### Troubleshooting the IPP server

If an error occurs using the IBM IPP Server Administrator, the message frame box will typically contain the error data and provide information about how to correct the problem.

For additional information, internal tracing can be used. Tracing provides a mechanism for retrieving state information when the IPP Server Administrator is running. To activate tracing, specify a valid error log file name using the **Error Logs** form. Then, select a logging level below **Critical**. Selecting a logging level of **Debug** provides the maximum amount of tracing information. The trace log containing detailed information about the operation of the administrator is located in file /QIBM/UserData/OS400/Ipp/Logs/qippcfg.log.



The trace log that contains detailed information about the operation of the IPP server is located in file /QIBM/UserData/OS400/Ipp/Logs/qippsvr.log.

To stop internal tracing for the administrator and the IPP server, select a logging level above Error, such as Critical, Alert, or Emergency, on the **Error Logs** form and click **Apply** .

**Note:** The trace files /QIBM/UserData/OS400/Ipp/Logs/qippcfg.log and qippsvr.log are deleted every time tracing/error logging is started. If you want to trace, it is recommended that at regular intervals you stop tracing, archive the trace file, and start the tracing again.

Use the following table to help troubleshoot other problems that you might encounter when working with the IBM IPP Server Administrator or the IBM IPP Server.

Symptom	Action
You do not have the correct authority to a file.	To change any configuration files, IBM IPP Server Administrator browser interface users require at least *IOSYSCFG authority. Verify that you have the correct authority.
File does not exist or is corrupted.	Always make sure the following files exist on your system: /QIBM/UserData/OS400/Ipp/conf/qippsvr.conf /QIBM/UserData/OS400/Ipp/conf/preferences.properties /QIBM/UserData/OS400/Ipp/conf/printer.properties  You should not manually edit these files if you are not familiar with this task. If you change one file, you made also need to change additional files. If you change these files manually, the server might fail to initialize.
The log file name is not valid.	Click <b>Help</b> for the <b>Log file name</b> field for more information about valid log file names.
You made changes to the IPP server configuration, but they do not appear to be recognized.	Stop the IPP server and start it again. Whenever a change is made to the server's configuration, you need to restart the IPP server for the change to become effective.
The browser interface is not being displayed in the language specified.	The language specified in the browser is not supported by the IBM IPP Server Administrator browser interface application.
You cannot find help for a field.	Click <b>Help</b> for information. You can also activate verbose for the form, which provides a more detailed overview of the entire form.
An internal error occurred using the browser interface.	Turn on tracing and examine the trace file to help locate the problem.

Symptom	Action
The IPP server does not start or stay running after you attempt to start it.	<p>An error has probably occurred during initialization. Whenever the IPP server is started, an i5/OS fully-qualified job name is displayed in the message frame, in the form: JobNumber/JobUser/JobName</p> <p>If the IPP server fails to start, follow these steps:</p> <ol style="list-style-type: none"> <li>1. Log on to the IBM i operating system.</li> <li>2. Enter WRKJOB JOB (<i>JobNumber/JobUser/JobName</i>) .</li> <li>3. Select option 4, (Work with Spooled Files).</li> <li>4. Using display option 5, view any spooled files to help locate the cause of the problem.</li> </ol> <p>If you have manually edited the IPP server configuration file, a newly specified directive might be causing the IPP server to fail initialization. If this is the case, edit the configuration file and remove the directive in error.</p>
The IPP server does not stop after you attempt to stop it.	<p>An error has occurred on the IBM i operating system. To stop the IPP server manually:</p> <ol style="list-style-type: none"> <li>1. Log on to the IBM i operating system.</li> <li>2. Enter WRKACTJOB SBS(QHTTSPVR).</li> <li>3. Find the QIPPSVR jobs.</li> <li>4. Use option 4 and end the jobs immediately.</li> </ol>
When SSL is enabled, the IPP server does not stay running.	<p>The IPP server might not have a valid certificate associated with it for SSL connections. Use Digital Certificate Manager for i5/OS to create a certificate and associate it with the IPP server. (The IPP server itself should be registered with DCM automatically.) Associate the certificate with the application QIBM_IPP_QIPPSVR.</p>

## Troubleshooting the image print transform

These are answers to questions that might arise when you use the image print transform function or the Convert Image API.

- Why does it take longer to process PostScript data streams?

One reason why PostScript data streams take a long time to process is the amount of information that needs to be transformed. Color documents especially require large amounts of memory and many data conversions, which means longer processing times.

**Note:** If the photometricity of the converted data stream is not requested, it is assumed by default to be RGB, or color. However, if you do not want RGB or the input data stream is not color, specify an image configuration object that only supports black and white output. This can greatly increase the throughput of the image print transform function and speed up PostScript processing.

- Why is the converted data stream positioned incorrectly on or off the page? Why is it not centered?

The resolution specified in the image configuration object is probably not supported by the printer that the object is configured with. When this happens, an incorrect no print border is retrieved from the image configuration object and the data is consequently positioned incorrectly on the output page. The printer might also be set up to automatically add a no-print border, which can cause the output generated by the image print transform function to be shifted on the page. Verify that the correct image configuration object is being used with the printer, that the printer has been set up properly, and that the printer has been physically calibrated.

- Why didn't the PostScript data stream generate a new data stream?

The PostScript data stream probably did not contain any printable data. To verify this, start the image print transform function and check the job log of the writer. Look for a message that indicates that no printable data was found. If no message exists, an error might have occurred while processing the file. Refer to the job log of the corresponding QIMGSERV job.

- Why is the printed image three times the original size when converted from color or gray scale to black and white?

When a color image or gray scale image is converted to black and white, a dithering process takes place. In this process, a single color or gray scale pixel is transformed into a 3x3 matrix of pixels. Each pixel within this matrix is either black or white, depending on the color being rendered.

## Reference information

Here are additional reference materials related to printing.

## CL commands

These groups of CL commands can be used to configure and manage printing.

### Jobs

These commands can be used to work with jobs.

CL command	Description
ADDJOBSCDE	Schedule batch jobs by adding an entry to the job schedule.
CHGJOB	Change some of the attributes of a job such as the default printer device or output queue.
CHGJOB	Change the values of a job description. If you have the correct level of authority, you can change other user's job description values.
CRTJOB	Create job descriptions. When a user signs on, the values in the job description determine which output queue and printer device are used.
HLDJOB	Makes a job ineligible for processing by the system.
RLSJOB	Makes a job eligible for processing after that job is held from processing.
WRKJOB	Allows you to work with or change information concerning a user job.
WRKJOB	Change the attributes of a user's job description. You must have special authority to change the values of a job description other than your own.

## Output queues

These commands can be used to work with output queues.

CL command	Description
CHGOUTQ	Change certain attributes of an output queue, such as the sequence of the spooled files in the output queue.
CLROUTQ	Remove all spooled files from an output queue.
CRTOUTQ	Create a new output queue.
DLTOUTQ	Delete an output queue from the system.
HLDOUTQ	Prevent all spooled files from being processed by the printer writer.

CL command	Description
RLSOUTQ	Release a previously held output queue for processing by the printer writer.
STRSPLRCL	Repairs output queues and spooled files that are left in unrecoverable states.
WRKOUTQ	Show the overall status of all output queues or the detailed status of a specific output queue.
WRKOUTQD	Show descriptive information for an output queue.

## Printer devices

These commands can be used to work with printer devices.

CL command	Description
CHGDEVPRT	Change a device description for a printer device.
CRTDEVPRT	Create a device description for a printer device.
WRKGFSTS	Display a list of devices and work with their status.

## Printer files

These commands can be used to work with printer files.

CL command	Description
CHGPRTF	Change attributes of the printer file.
CRTPRTF	Create a printer file.
DLTOVR	Delete printer file overrides.
DSPOVR	Display printer file overrides.
OVRPRTF	Override (replace) the file named in the program, override certain parameters of a file that are used by the program, or override the file named in the program and override certain parameters of the file processed

## Printer writers

These commands can be used to work with printer writers.

CL command	Description
CHGWTR	Change some printer writer attributes such as form type, number of file separator pages, or output queue attributes.
ENDWTR	End a printer writer and make its associated printer device available to the system.
HLDWTR	Stops the printer writer at the end of a record, at the end of a file, or at the end of a page.
RLSWTR	Release a previously held printer writer.
STRPRTWTR	Start a printer writer to a specified printer device in order to print spooled files on that device.
STRRMWTR	Start a remote writer to a specified output queue. The spooled files in that queue are sent to a specified remote system.

CL command	Description
WRKWTR	Display all the printers configured to the system by specifying WTR(*ALL) and the output queue (OUTQ) parameter.

## Spooled files

These commands can be used to work with spooled files (also known as printer output).

CL command	Description	System i Navigator instructions
CHGSPLFA	Change some attributes of a spooled file such as the output queue name or number of copies.	"Changing attributes of a spooled file" on page 216
CPYSPLF	Copy a spooled file to a specified database file.	Not applicable.
DLTEXPSPLF	Delete expired spooled files.	"Controlling the number of spooled files" on page 217
DLTSPLF	Delete a spooled file from the output queue.	"Deleting spooled files" on page 214
DSPSPLF	Display data records of a spooled file.	"Displaying the contents of a spooled file" on page 212
HLDSPFL	Stop the processing of an output file by a printer writer.	"Holding a spooled file" on page 213
RCLSPLSTG	Reclaim empty spooled file members.	Not applicable.
RLSSPLF	Release a previously held spooled output file for processing by the printer writer.	"Releasing a spooled file" on page 213
SNDNETSPLF	Send a spooled file to another system using SNADS.	"Sending a spooled file to another user or system" on page 215
SNDTCPSPLF	Send a spooled file to another system using TCP/IP.	"Sending a spooled file to another user or system" on page 215
STRSPLRCL	Repairs output queues and spooled files that are left in unrecoverable states.	Not applicable.
WRKSPLF	Display a list of spooled files and then be able to perform a variety of actions on those files.	"Displaying a list of spooled files" on page 212
WRKSPLFA	Show the current attributes of a spooled file.	Not applicable.

## User profiles

These commands can be used to work with user profiles.

CL command	Description
CHGUSRPRF	The Change User Profile (CHGUSRPRF) command changes the values specified in a user profile.
CRTUSRPRF	The Create User Profile (CRTUSRPRF) command identifies a user to the system and allows you to customize the way the system appears.
DSPUSRPRF	The Display User Profile (DSPUSRPRF) command displays the contents of a user profile.

CL command	Description
DLTUSRPRF	The Delete User Profile (DLTUSRPRF) command deletes a user profile from the system.
WRKUSRPRF	The Work with User Profiles (WRKUSRPRF) command allows you to specify the name of one or more user profiles you want shown.

## Printer file parameter considerations

These printer file parameters are discussed in more detail to help you better understand how to use them.

### Align (ALIGN) parameter

The align parameter on the Start Printer Writer (STRPRTWTR) and Create Printer File (CRTPRTF) commands influences how the IBM i operating system issues messages to check forms alignment in printers before they start to print.

If the value on the STRPRTWTR align parameter is \*WTR, the printer writer keeps track of spooled files that are to be printed and issues a forms alignment message whenever it determines that forms alignment is needed.

If this value is \*FIRST, the forms alignment message is issued only for the first spooled file printed.

The following events cause the printer writer to issue a forms alignment message:

- Delete or hold spooled file command (option \*IMMED) issued against a spooled file with a status of WTR
- Printer writer or spooled file restarted
- Previous spooled file was created with ASCII data (virtual print) and current spooled file was not
- Forms length of previous spooled file is different from current spooled file
- File being printed is the first file to print after a printer writer has been started
- Forms type is changed (after a G reply to a CPA3394 or CPA3395 message)

**Note:** A B reply to these messages allows you to skip or not receive the alignment message.

- Incorrect control characters have been detected on a workstation printer in the previous file
- The Cancel key on a workstation printer has been pressed while the previous file is being printed
- A C (Cancel) reply has been given to the inquiry message for unprintable characters detected on a workstation printer in a previous file
- An H (Hold) reply has been given to some inquiry messages on a workstation printer
- A printer writer has recovered from a communications failure on a remote workstation printer

If you do not want the printer writer to control the issuing of forms alignment messages, you can specify \*FILE as the value for the align parameter of the STRPRTWTR command and \*YES on the align parameter of the printer file used with the application that produces the spooled file.

If the value on the STRPRTWTR align parameter is \*FILE, the printer writer looks at the attributes of the spooled file created by the application program. Specifically, the printer writer looks at the align parameter value of the spooled file attributes to determine if it should send a forms alignment check.

If the align parameter value of the spooled file attributes is \*YES, the printer writer sends a message to check forms alignment on the target printer.

If the align parameter value of the spooled file attributes is \*NO, the printer writer does not send any messages to check forms alignment on the target printer.

If you want to avoid any forms alignment messages, you can specify \*FILE for the align parameter on the Start Printer Writer (STRPRTWTR) command and \*NO as the align parameter value of the printer file. With this combination of values, the system does not send any messages to check alignment.

### **Coded font (CDEFNT) parameter**

This parameter is specified only for printer files with DEVTYPE (\*AFPDS) when printing on IPDS-capable printers configured with AFP(\*YES).

A coded font is the combination of a font character set and a code page. This combination is assigned a name and called a coded font.

**Note:** The coded font contains only the names of the font character set and code page. It does not contain the font and code page data.

A point size can be specified for outline fonts. It will be ignored for raster fonts.

To find out which coded fonts come with the IBM i operating system, you can use the Work with Font Resources (WRKFNTRSC) command and specify QFNTCPL for the library and \*CDEFNT as the object attribute.

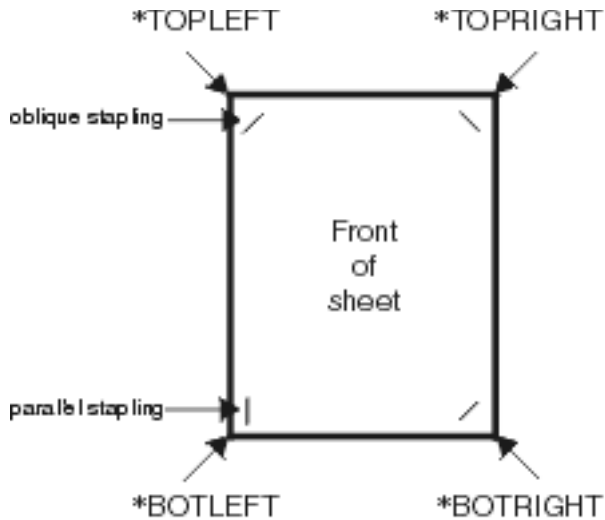
**Note:** If you have obtained coded fonts from other sources but have them in a different library, the WRKFNTRSC command can display the coded fonts located in that library.

You can specify a library for the coded font specified on the printer file. However, if the font character set and code page that make up the coded font are not in a library that is defined to your library list, the coded font is not found.

### **Corner staple (CORNERSTPL) parameter**

This parameter specifies the corner of the media to be used for stapling.

The following diagram indicates the corners that you can specify for a printer. The corners that you can specify are device dependent. Refer to your printer's documentation for this information. Note that page rotation does not affect the staple placement.



**Note:** The offset and angle of the staple from the selected corner are device dependent.

RMW 1974

### DBCS coded font (IGCCDEFNT) parameter

The DBCS coded font is the font that the system uses for DBCS printing on IPDS printers. This parameter is specified only for printer files with DEVTYPE (\*SCS) or (\*AFPDS) when printing on IPDS-capable printers configured with AFP(\*YES). A point size can be specified for outline fonts. It will be ignored for raster fonts.

### Transforming SCS DBCS data to Advanced Function data stream data

When transforming the SCS data stream to Advanced Function Presentation data stream, the IGCCDEFNT parameter is used to print the DBCS data. When the spooled file is created as SCS, it contains SO/SI (shift out / shift in) characters to identify the double-byte data. When the printer writer is printing a spooled file, a font change to the DBCS coded font replaces the SO in the AFP data stream. When the SI is encountered in the data stream, a font change is placed in the data stream to change the font back to the previous SBCS font.

### Generating Advanced Function Presentation data stream data to spool

The IGCCDEFNT parameter is used when generating Advanced Function Presentation data stream. If the data stream contains double-byte data (IGCDTA(\*YES)), a font substitution to the DBCS coded font takes place.

In an externally described file (DDS), the user can use the IGCCDEFNT DDS keyword to specify a DBCS font. This font will be used to print any DBCS data encountered in that field or record. If the user has a DBCS-graphic field specified, the printer file IGCCDEFNT parameter is used unless the IGCCDEFNT DDS keyword has been specified at the record or field level.

### Device type (DEVTYPE) parameter

The device type (DEVTYPE) parameter specifies the type of data stream created for a printer file. This parameter indicates whether the resulting data stream should be an Intelligent Printer Data Stream (\*IPDS), an SNA character stream (\*SCS), an ASCII data stream (\*USERASCII), an Advanced Function Presentation Data Stream (\*AFPDS), line data (\*LINE), or mixed data (\*AFPDSLIN).

An AFP data stream can be created from several sources:

- System z platform



- PrintManager™
- AFP Utilities for i5/OS
- Using the virtual print function with the System i Access for Windows licensed program

However, to use the functions provided by many of the printer file parameters that are supported for AFP data stream, the spooled file must be created with a printer file that specifies a device type of \*AFPDS. These parameters include FRONTMGN, BACKMGN, FRONTOVL, BACKOVL, FNTCHRSET, CDEFNT, and IGCCDEFNT. In cases where the IBM i operating system does not actually generate the AFP data stream (see the previous list), either a printer file is not used or the DEVTYPE parameter for the printer file is ignored.

Throughout this topic, AFP data stream output that is created on the IBM i operating system with a printer file specifying DEVTYPE(\*AFPDS) is referred to as \*AFPDS created on the system.

If DEVTYPE(\*USERASCII) is specified, the user is responsible for the content of the entire data stream (such as PPDS for the 4019 printer). The system does not send any formatting commands that correspond to the spooled file attributes. For example, the system sends initial formatting commands to the printer that set up the page size, lines per inch, characters per inch, and font for spooled files with DEVTYPE(\*SCS) or DEVTYPE(\*IPDS). These commands are not sent for spooled files with DEVTYPE(\*USERASCII). Instead, the system sends the contents of the spooled file without adding any formatting commands.

This spooled file can then be sent to an ASCII printer attached to a System i product. It is suggested that you not use file or job separators when you specify DEVTYPE(\*USERASCII).

**Note:** Do not put hex 03 transparency commands in the data stream. Use only ASCII commands that the target printer can understand. The system inserts the hex 03 commands with the correct lengths; thus, no EBCDIC-to-ASCII translation occurs.

### Edge stitch (EDGESTITCH) parameter

This parameter specifies where staples are to be placed along the finishing margin of the media. You need to check your printer documentation to determine which elements, and values of elements, are supported. Where your printer does not support any of the values for a specific element, specify a value of \*DEVVD for the element.

**Note:** The finishing margin is an invisible line along which finishing operations, such as edge stitching, are done. You specify the position of the finishing margin, relative to the physical edge, in the reference edge offset element of the parameter.

#### Element 1: Reference Edge

Specifies which edge to use for finishing. Possible values are:

**\*DEVVD**

The default that is used by the device.

**\*BOTTOM**

The reference edge is the bottom edge.

**\*LEFT** The reference edge is the left edge.

**\*RIGHT**

The reference edge is the right edge.

**\*TOP** The reference edge is the top edge.

#### Element 2: Reference Edge Offset

Specifies the offset from the reference edge to place the edge stitching. Possible values are:

**\*DEVVD**

The default that is used by the device.

**reference-edge offset**

This element that is specified in centimeters (range 0 through 57.79) or inches (range 0 through 22.57).

**Element 3: Number of Staples**

Specifies the number of staples to use for edge stitching. Possible values are:

**\*DEVD**

The default value for the device. This is the value that is used if \*DEVD is also specified for the staple offsets value of this parameter. The system uses the default number of staples for the device when you specify \*DEVD for this and for the staple offset value.

**number of staples**

The valid value range is from 1 to 122 staples. The number of staples is the same as the number of staple offsets specified.

**Element 4: Staple Offsets**

Specifies the distance between staples that are used in the edge stitching. If the staple-placement will be on the left or right edge of the paper, the first staple offset is determined by measuring from the intersection of the finishing margin and the bottom edge of the paper to where the center of the staple will be. Subsequent staple offsets are measured from the same point (not from the previous staple). If the staple-placement will be at the top or bottom edge of the paper, the first staple offset is determined by measuring from the intersection of the finishing margin and the left edge of the paper to where the center of the staple will be. Subsequent staple offsets are measured from the same point (not from the previous staple). Possible values are:

**\*DEVD**

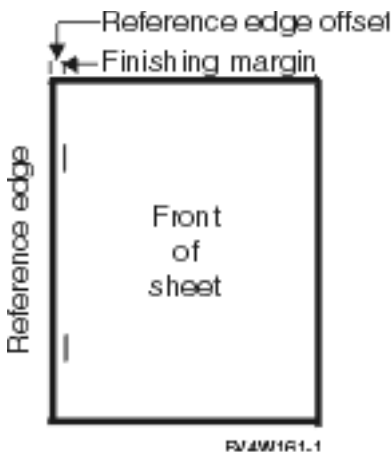
The default staple offset for the device. If you specify a value for the Number of staples, the printer will automatically calculate the position of each staple.

**staple offset**

The valid value range is from 1 to 122 staple offsets. If you specify a staple offset, the Number of staples must be \*DEVD. This measurement is in centimeters (range 0 through 57.79) or inches (range 0 through 22.57)

The following diagram shows the reference edge offset from the left edge with two staples. It uses the following values:

- Element 1: Reference edge—\*LEFT
- Element 2: Reference edge offset—\*DEVD
- Element 3: Number of staples—\*DEVD
- Element 4: Staple offsets—\*DEVD



## Fidelity (FIDELITY) parameter

For AFP functions that are not supported, the FIDELITY parameter allows the user to determine if a spooled file with advanced printing functions should continue to print on an IPDS printer configured AFP(\*YES) or stop printing when an advanced printing function is encountered that is not supported.

For example, if a spooled file containing bar code commands is to be printed on a 3820 printer and the FIDELITY parameter value is:

- \*ABSOLUTE, the spooled file does not print because the 3820 does not support bar codes.
- \*CONTENT, the spooled file prints without the bar codes.

**Note:** If the device description of the printer has the value for the print while converting (PRTCVT) parameter set to \*YES, the spooled file can print the pages up to the point where the bar code was encountered no matter what the fidelity parameter value is.

## Fidelity and other printer file parameters

If FIDELITY(\*ABSOLUTE) is specified, the following list of printer file parameters must have a valid value specified for the selected printer. Otherwise, the spooled file is held.

- DRAWER
- FONT
- DUPLEX
- MULTIUP
- OUTBIN
- PAGRTT
- FRONTOVL
- BACKOVL

If FIDELITY (\*CONTENT) is specified, a default value is used. For example, if DUPLEX (\*YES) is requested, but the printer does not support duplex printing, the spooled file is printed on one side of the paper.

## Font character set (FNTCHRSET) parameter

This parameter is specified only for printer files with DEVTYPE (\*AFPDS) when printing on IPDS-capable printers configured with AFP(\*YES).

When using a font character set, a character set and a code page must be specified on the font character set (FNTCHRSET) parameter of the printer file being used. A point size can be specified for outline fonts. It will be ignored for raster fonts.

You cannot specify a coded font if you use the FNTCHRSET parameter on the printer file.

To find out which font character sets and code pages come with the IBM i operating system, you can use the Work with Font Resources (WRKFNTRSC) command and specify QFNTCPL for the library and \*FNTCHRSET or \*CDEPAG as the object attribute.

Font character sets and code pages are downloaded from the System i platform to an IPDS printer when the spooled file is printed. They are supported on all IPDS printers except the 4224, 4230, 4234, 4247, and 64xx. The use of font character sets provides greater consistency in the printed appearance of output from different printers.

**Note:** When a printer file is created and a character set and code page are specified for the font character set (FNTCHRSET) parameter, column spacing is done using this printer file level parameter. Any

fonts or code pages specified in the DDS FNTCHRSET keyword are ignored. The font and code page specified in the printer file parameter FNTCHRSET are used.

### **Margin (FRONTMGN and BACKMGN) parameters**

To use the margin parameters, the device type (DEVTYPE) parameter on the printer file must be \*AFPDS. For device types other than \*AFPDS, the system calculates the margins.

Margins define the starting point of printed output on a piece of paper. The FRONTMGN parameter specifies the starting point on the front side of the paper; the BACKMGN parameter specifies the starting point on the back side of the paper.

There are two types of margins: front and back. Offset values, down and across, are used to fix the position of the margin. Across is defined as left to right. Down is defined as top to bottom.

Margins are measured in either inches or centimeters. The type of measurement is specified in the unit-of-measure (UOM) parameter on the printer file.

### **Using \*DEVD and 0 as margin parameter values**

If you have existing application programs that specify how far across and how far down to start printing, you should specify 0 (zero) or \*DEVD for the margin parameter offset values.

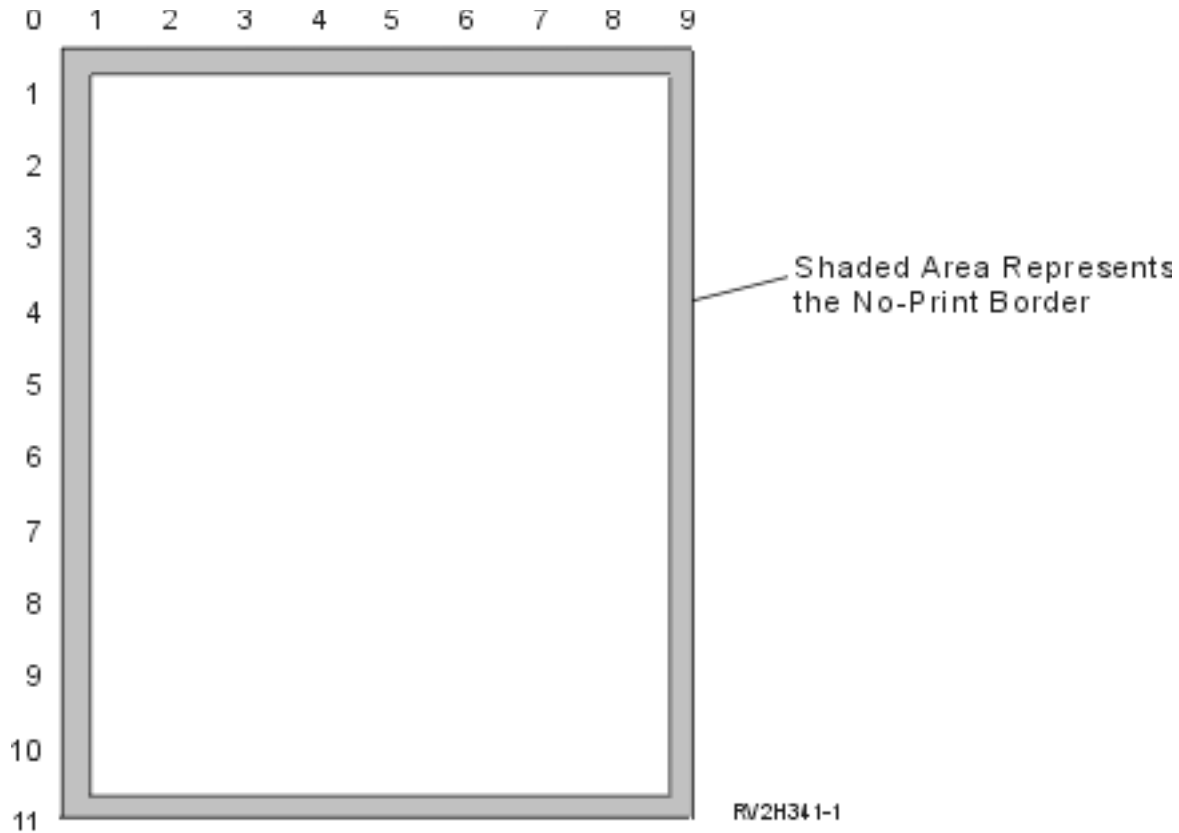
#### **\*DEVD**

When \*DEVD is specified, the no-print border (*no-print border* is a small area around the whole page where data will not print) is used to determine the starting printing point.

For printers configured as AFP(\*NO), 0 is used to determine the starting point when \*DEVD is specified.

**0** When 0 is specified, the top left corner of the page is used to determine the starting printing point.

The following diagram shows a no-print border. The size of the no-print border can vary from printer to printer.



### Restrictions with the margin parameters

The margin parameters (both front and back) are ignored for spooled files that have \*COR specified as the value for the page rotation (PAGRTT) printer file parameter. This is because \*COR assumes a half-inch margin. In addition, the margin parameters are ignored for spooled files that have a pages per side (MULTIUP) value of 2, 3, or 4.

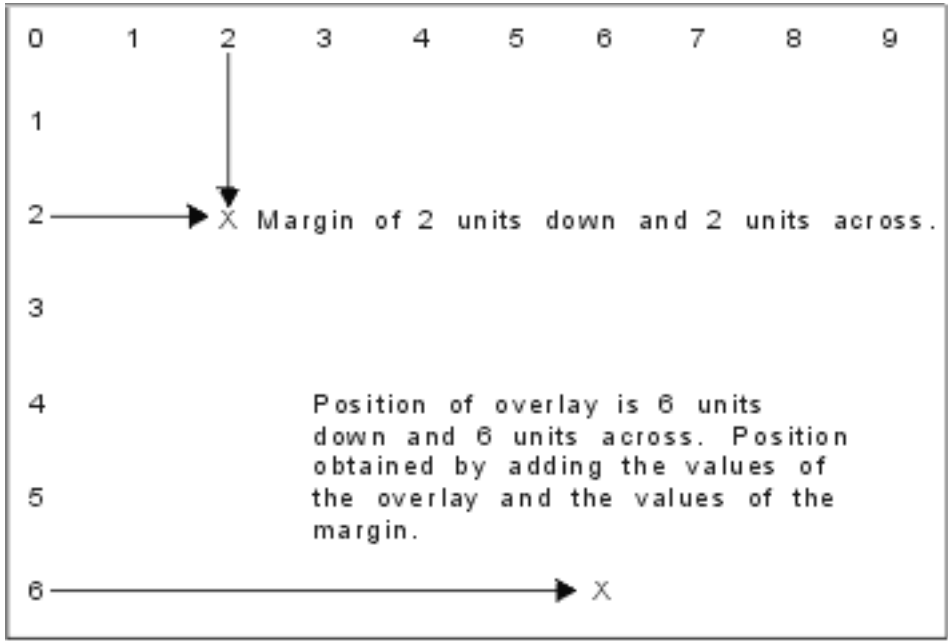
### Using margin parameters and overlays

There are two ways to position your overlays.

- Specify overlays with the front or back margin (FRONTMGN or BACKMGN) parameters so that the overlays move with the text based on margin parameters.
- Change value in data area QPRTVALS so that the overlays are not affected by the front or back margin parameters.

### Positioning overlays based on margin parameters

Overlays specified with the back or front overlay printer file parameter are affected by the front or back margin parameters. The following diagram shows how your output looks if your front margin was defined as 2 units down and 2 units across and the placement of your overlay was 4 units down and 4 units across:

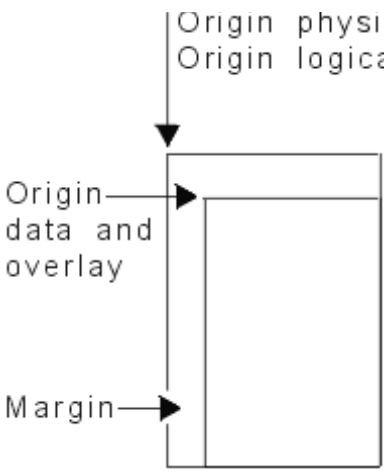


RRAFT516-C

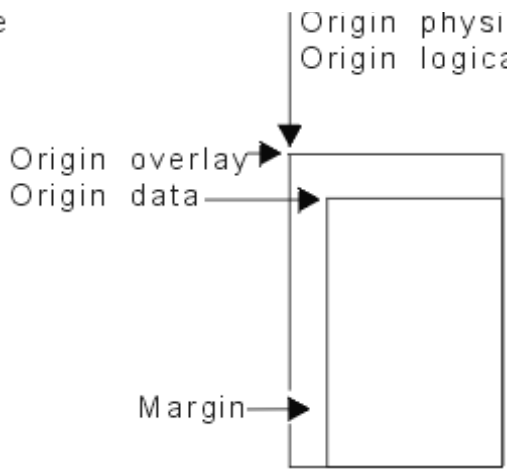
**Note:** If you use margins and overlays together, the offsets have the same unit of measure (UOM) specified: inches (\*INCH) or centimeters (\*CM), but not inches and centimeters together.

**Positioning overlays independently**

If you do not want the overlays moved by the values specified in the front or back margin parameters, you can specify a Y (uppercase) in position 4 of the data area QPRTVALS. The following diagram shows you the two ways to position the overlays:



Front and back margins used  
overlay position move  
QPRTVALS position 4=' '(blank)



Front and back margins used  
overlay position not affected  
QPRTVALS position 4='Y'

RRAFT517-1

To find out if data area QPRTVALS exists on your system, use the following command:

DSPDTAARA DTAARA(QUSRSYS/QPRTVALS)

If the data area QPRTVALS exists, but position 4 is not set to 'Y' (uppercase), use the following command:  
CHGDTAARA DTAARA (QUSRSYS/QPRTVALS (4 1)) VALUE ('Y')

If you do not have data area QPRTVALS on your system, you can create one by using the following commands:

1. CRTDTAARA DTAARA(QUSRSYS/QPRTVALS) TYPE(\*CHAR) LEN(256) VALUE(' Y')
2. CHGOBJOWN OBJ(QUSRSYS/QPRTVALS) OBJTYPE(\*DTAARA) NEWOWN(QSYS) CUROWNOUT(\*SAME)
3. CRTOBJAUT OBJ(QUSRSYS/QPRTVALS) OBJTYPE(\*DTAARA) USER(\*PUBLIC) AUT(\*ALL)

### **MULTIUP(1, 2, 3, or 4) and REDUCE(\*NONE) parameters**

To use hardware MULTIUP you can select any MULTIUP value (1 through 4), and you must specify \*NONE as the value on the REDUCE parameter of the printer file. It is then up to the application to ensure the output fits in the partition (portion) of the page.

**Note:** Hardware MULTIUP (REDUCE(\*NONE)) is only supported by some printers. The 3130 or 3935 is an example of a printer that supports hardware MULTIUP. If the target printer does not support hardware MULTIUP, the pages printed are not printed using the MULTIUP function.

The partitioning of the page is determined by the printer. This combination of MULTIUP and REDUCE parameter values also allows multiupping of data streams restricted by software multiupping. For example, \*LINE and \*AFPDSLIN data streams can be multiupped as can data streams which contain advanced functions and host resident fonts.

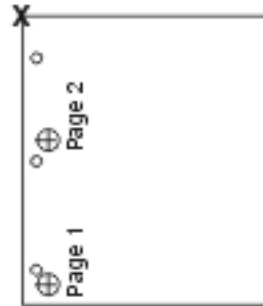
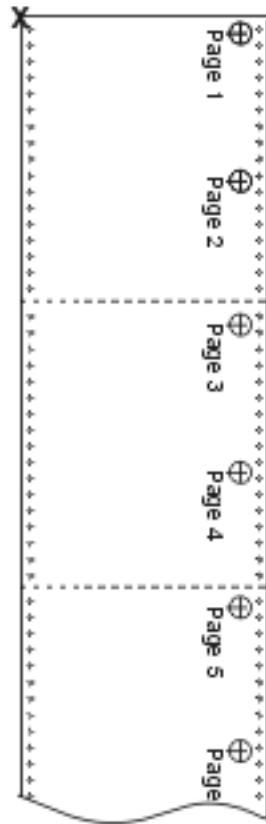
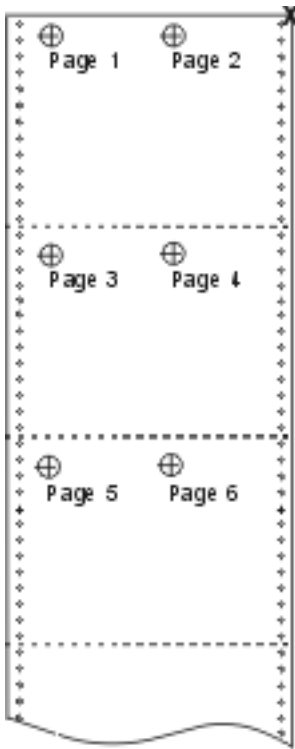
When REDUCE(\*NONE) is specified, overlays, page segments, and resources stored in the integrated file system apply to each partition of the multiupped page. Thus, if front overlay (FRONTOVL) and back overlay (BACKOVL) are used for a MULTIUP(2) REDUCE(\*NONE) job, the FRONTOVL is printed in the first partition and BACKOVL in the second partition.

#### **Notes:**

1. The corner with the X is the physical paper origin. It is the left edge of the narrow side of the page.
2. The circle with the plus (+) sign in it designates the logical paper origin.

The following diagrams are examples of how output prints when REDUCE(\*NONE) and MULTIUP(2, 3, or 4) are specified.

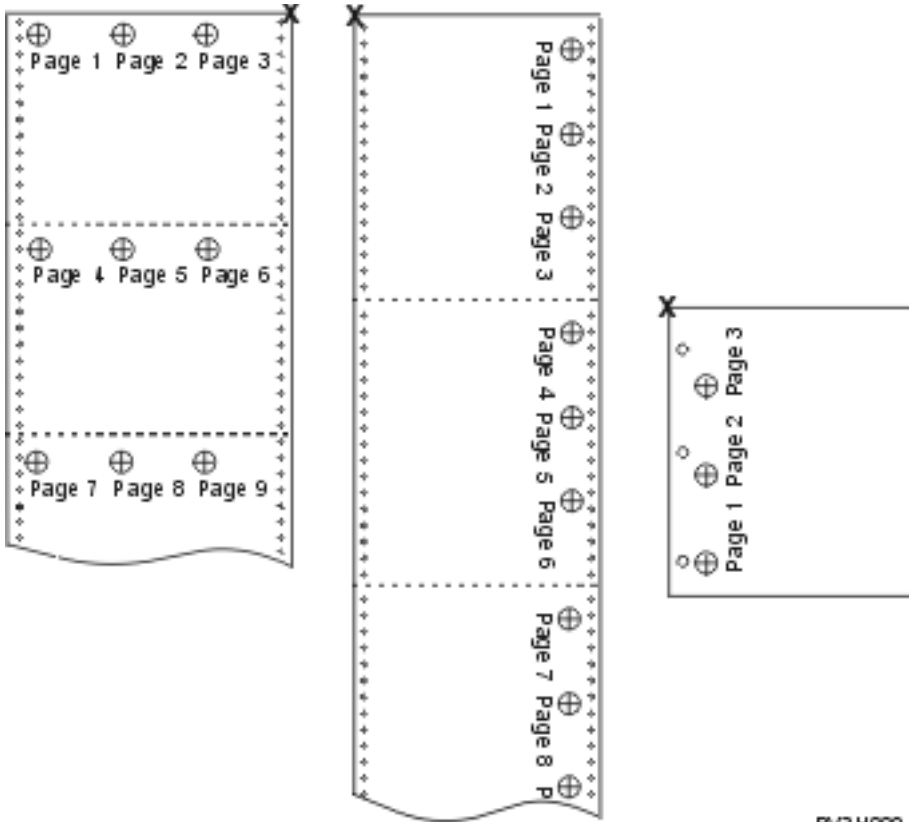
MULTIUP(2), REDUCE(\*NONE), and PAGRTT(0)



RV/3 H000-1

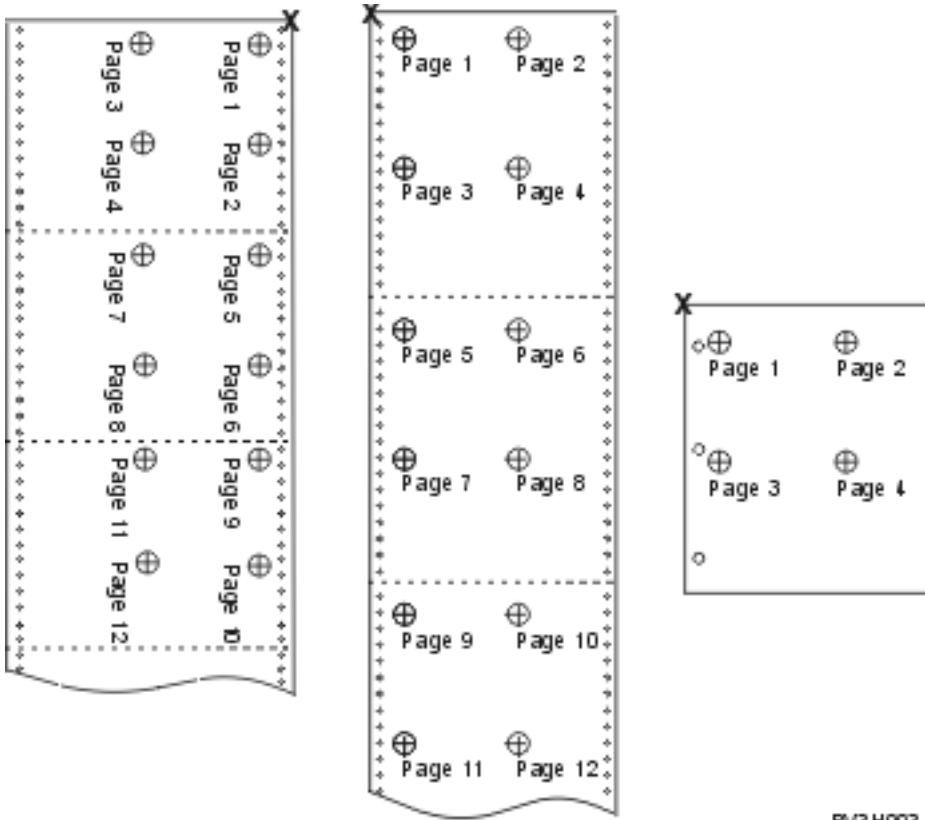
MULTIUP(3), REDUCE(\*NONE), and PAGRTT(0)





Rv/3 H002-1

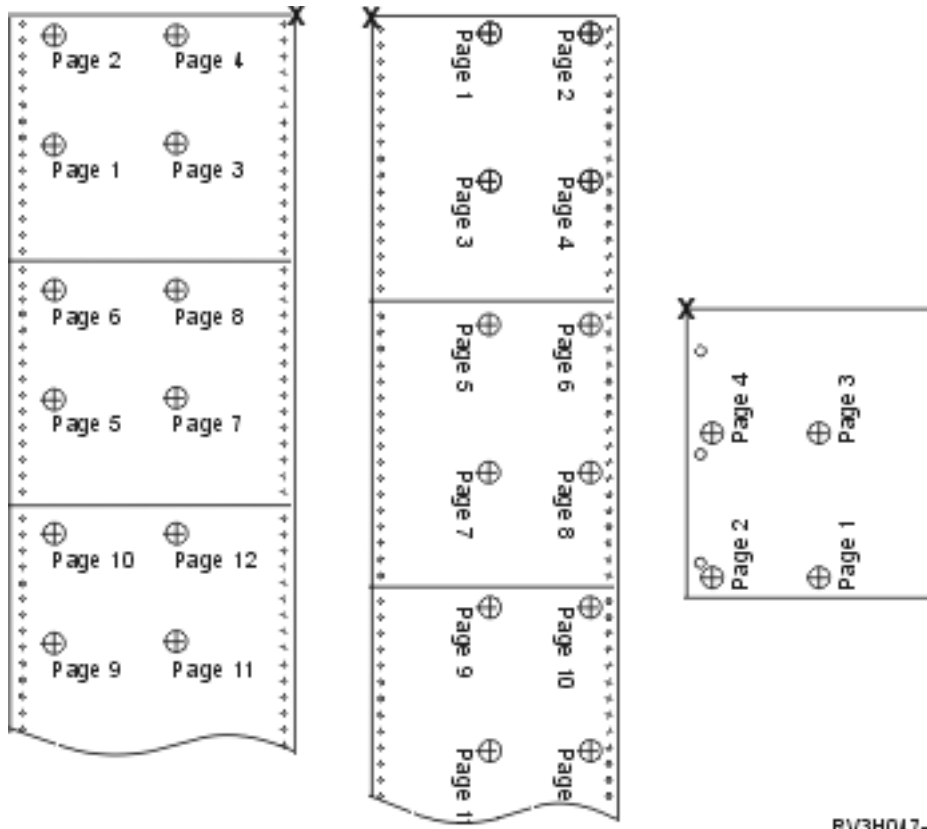
MULTIUP(4), REDUCE(\*NONE), and PAGRTT(0)



RM3 H003-1

MULTIUP(4), REDUCE(\*NONE), and PAGRTT(90)

**Note:** It is recommended that you not use rotation with MULTIUP and REDUCE(\*NONE). Following the reading direction can be difficult due to the way the data is oriented on the physical page. The example below shows how the output prints.



RV3H017-F

### **MULTIUP(1, 2, 3, or 4) and REDUCE(\*TEXT) parameters**

The MULTIUP parameter allows you to print more than one logical page of data on a piece of paper. Depending on the printer that you are using, you can print from one to eight logically formatted pages on one physical piece of paper.

Using a printer that supports duplex printing, you can have four logical pages printed on each side of the physical paper, thus reducing the number of physical pages printed from eight to one.

**Note:** The value 3 is not allowed for MULTIUP when REDUCE(\*TEXT) is specified.

The MULTIUP function can be software or hardware controlled. The software MULTIUP function is selected by specifying REDUCE(\*TEXT) on the printer file. The hardware MULTIUP function is selected by specifying REDUCE(\*NONE) on the printer file.

The information presented here is dependent on specifying a REDUCE parameter value of \*TEXT.

For information about MULTIUP when the REDUCE parameter value is \*NONE, see “MULTIUP(1, 2, 3, or 4) and REDUCE(\*NONE) parameters” on page 263.

The rotation value specified in the PAGRTT parameter of the printer file is used to determine the rotation for MULTIUP printing. When PAGRTT(\*AUTO), PAGRTT(\*DEVVD), or PAGRTT(\*COR) is specified and the MULTIUP parameter has a value of 2 or 4, the PAGRTT value used is 0.

**Note:** If overlays (using the front overlay (FRONTOVL) and back overlay (BACKOVL) parameters) are included in output using MULTIUP support, the overlay applies to the whole sheet. That is, the front overlay is placed on the front side of the paper and the back overlay on the back side.

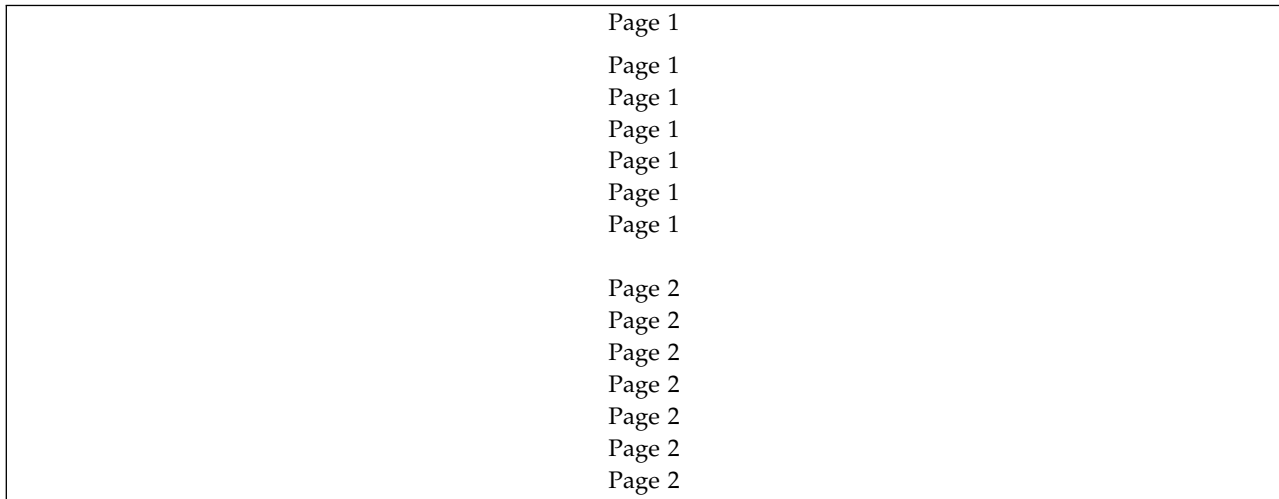
## MULTIUP support:

MULTIUP support is available on the following printers: 3130 3812 3816 3820 3825 3827 3828 3829 3831 3835 3900 3912 3916 3930 3935 4028 4312 4317 4324 Infoprint 20 Infoprint 32.

These are IPDS printers that support page rotation (PAGRTT) and pages per side (MULTIUP).

For all the above-described printers, the following diagrams are examples of how the output will print depending on the PAGRTT and MULTIUP values.

PAGRTT(0) or PAGRTT(180) and MULTIUP(2)



If the logical page width is less than or equal to 8 inches and the logical page length is less than 5 inches, the font is not changed.

If the logical page width is greater than 8 inches, or the logical page length is greater than 5 inches, the font is selected based on the following table:

Any references to the 4028 printer in the following tables also apply to the 3912, 3916, 4312, 4317, 4324, Infoprint 20, and Infoprint 32 printers. The following footnotes apply to all of the tables about MULTIUP Font Substitution that are included here.

### Notes:

1. If the target printer does not have font 230 resident, it uses font 223.
2. If the target printer is a 4028 and has font 283 (20 pitch) resident, it uses font 283. Otherwise, it uses font 281 (20 pitch).
3. If the target printer is a 3130 or 3935, it will use font 416 with point size of 4 (30 pitch). If the target printer is a 4028 and does not have font 290 resident but does have font 283 resident, it will use font 283. If the target printer is a 4028 and does not have font 290 or 283 resident, it will use font 281 (20 pitch).

Table 10. MULTIUP font substitution for page rotation 0 or 180 with MULTIUP(2)

Page Width (Characters)	Page Length (Inches)	Font Used	Pitch (CPI)
1 through 123 characters	Less than or equal to 5 inches	230 <sup>1</sup>	15
1 through 123 characters	Greater than 5 inches	254	17
124 through 139 characters	All	254	17
140 through 163 characters	All	281 <sup>2</sup>	20 <sup>2</sup>

Table 10. MULTIUP font substitution for page rotation 0 or 180 with MULTIUP(2) (continued)

Page Width (Characters)	Page Length (Inches)	Font Used	Pitch (CPI)
Greater than or equal to 164 characters	All	290 <sup>3</sup>	27 <sup>3</sup>

PAGRTT(90) or PAGRTT(270) and MULTIUP(2)

Page 1	Page 2
Page 1	Page 2
Page 1	Page 2
Page 1	Page 2
Page 1	Page 2
Page 1	Page 2
Page 1	Page 2
Page 1	Page 2

If the logical page width is less than or equal to 5 inches and the logical page length is less than 8 inches, the font is not changed.

If the logical page width is greater than 5 inches, or the logical page length is greater than 8 inches, the font is selected based on the following table:

Table 11. MULTIUP font substitution for page rotation 90 or 270 with MULTIUP(2)

Page Width (Characters)	Page Length (Inches)	Font Used	Pitch (CPI)
1 through 73 characters	Less than or equal to 8 inches	230 <sup>1</sup>	15
1 through 73 characters	Greater than 8 inches	254	17
74 through 83 characters	All	254	17
84 through 97 characters	All	281 <sup>2</sup>	20 <sup>2</sup>
Greater than or equal to 98 characters	All	290 <sup>3</sup>	27 <sup>3</sup>

PAGRTT(0) or PAGRTT(180) and MULTIUP(4)

Page 1	Page 3
Page 1	Page 3
Page 1	Page 3
Page 1	Page 3
Page 1	Page 3
Page 1	Page 3
Page 1	Page 3
Page 2	Page 4
Page 2	Page 4
Page 2	Page 4
Page 2	Page 4
Page 2	Page 4
Page 2	Page 4
Page 2	Page 4

If the logical page width is less than or equal to 3.75 inches and the logical page length is less than 5 inches, the font is not changed.

If the logical page width is greater than 3.75 inches, or the logical page length is greater than 5 inches, the font is selected based on the following table:

Table 12. MULTIUP font substitution for page rotation 0 or 180 with MULTIUP(4)

Page Width (Characters)	Page Length (Inches)	Font Used	Pitch (CPI)
1 through 54 characters	Less than or equal to 5 inches	230*	15
1 through 54 characters	Greater than 5 inches	254	17
55 through 61 characters	All	254	17
Characters 62 through 71	All	281 <sup>2</sup>	20 <sup>2</sup>
Greater than or equal to 72 characters	All	290 <sup>3</sup>	27 <sup>3</sup>

PAGRTT(90) or PAGRTT(270) and MULTIUP(4) PAGRTT(0) or PAGRTT(180) and MULTIUP(4)

Page 1	Page 3
Page 1	Page 3
Page 1	Page 3
Page 1	Page 3
Page 2	Page 4
Page 2	Page 4
Page 2	Page 4
Page 2	Page 4

If the logical page width is less than or equal to 5 inches and the logical page length is less than 3.75 inches, the font is not changed.

If the logical page width is greater than 5 inches, or the logical page length is greater than 3.75 inches, the font is selected based on the following table:

Table 13. MULTIUP font substitution for page rotation 90 or 270 with MULTIUP(4)

Page Width (Characters)	Page Length (Inches)	Font Used	Pitch (CPI)
1 through 73 characters	Less than or equal to 5 inches	230 <sup>1</sup>	15
1 through 73 characters	Greater than 5 inches	281	20
74 through 83 characters	Less than or equal to 3.75 inches	254	17
74 through 83 characters	Greater than 3.75 inches	281	20
84 through 97 characters	All	281 <sup>2</sup>	20 <sup>2</sup>
Greater than or equal to 98 characters	All	290 <sup>3</sup>	27 <sup>3</sup>

#### MULTIUP support for the 4224, 4230, 4234, 4247 printers:

These are continuous-forms printers, and do not support rotation. MULTIUP(2) with PAGRTT(0) is the only valid combination. The following example shows what the physical output page looks like. If the logical page width is less than or equal to 6-1/2 inches, the font is not changed. Otherwise, font 223, which is 15 pitch, is used.

Page 1	Page 2
Page 1	Page 2
Page 1	Page 2
Page 1	Page 2

Page 1	Page 2
Page 1	Page 2
Page 1	Page 2
Page 1	Page 2

### Restrictions for MULTIUP with REDUCE(\*TEXT):

Here are the restrictions for MULTIUP with REDUCE(\*TEXT).

- MULTIUP requires an IPDS printer. It can be configured AFP(\*YES) or AFP(\*NO).
- MULTIUP reduces the size of any lines drawn in the printed output resulting from the use of the DDS LINE or BOX keywords. Depending on the pel resolution of the printer being used, the lines might become too narrow to print.
- MULTIUP with REDUCE(\*TEXT) is not supported for spooled files created with device type (DEVTYPE) of \*LINE, \*AFPDSLIN, or \*AFPDS unless the AFP data stream was created
- MULTIUP is ignored if the spooled file contains any of the following advanced printing functions:
  - Final form text
  - Variable fonts
  - Variable lines per inch
  - Variable drawer
  - Superscript or subscript
  - Variable character ID
  - Highlighting
  - Extended 3812 fonts
  - Graphics
  - Bar codes
  - Variable page rotation
  - PC printer emulation
  - Define characters
  - Variable characters per inch
  - Transparency commands
  - Field outlining
  - AFP resources (overlays, page segments, resources stored in the integrated file system, or host resident fonts)

### Output priority (OUTPTY) parameter

Once a spooled file is available to a writer, the OUTPTY parameter determines the order in which the files are produced.

The OUTPTY parameter supports the value \*JOB (use the default output priority for the job) and a range of values from 1 to 9. All available files that have an output priority of 1 are positioned at the top of the output queue and are the first files to be printed. Next are the priority 2 files and so on. By selecting the appropriate output priority for spooled files, you can ensure that the spooled files needed right away will be printed first.

The priority of a spooled file when it is created is set from the printer file. Use the Create Printer File, Change Printer File, or Override with Printer File commands to set the OUTPTY parameter to the desired value before the file is opened. After the file is opened, the output priority of the spooled file can be changed by using the CHGSPLFA command.

## **Overlay (FRONTOVL and BACKOVL) parameters**

When an overlay is specified in a printer file, you can merge data from a spooled file onto the same piece of paper that the overlay is printed on. The FRONTOVL parameter specifies the overlay to be printed on the front side of the paper; the BACKOVL parameter specifies the overlay for the back side of the paper.

With your application program, you can use overlays that you created yourself using the Advanced Function Presentation Utilities for i5/OS licensed program, or overlays sent from the System z platform.

For more information about the Advanced Function Presentation Utilities for i5/OS licensed program, see the AFP Utilities for i5/OS User's Guide PDF.

The following diagram is an example of an overlay you can create using the AFP utilities. Additionally, the diagram shows how the merged spooled file data (Variable Page Data in the diagram) can be integrated into one document.

The merged document can be printed on any IPDS printer configured with AFP(\*YES) in the printer's device description.


The overlays can only be merged with a spooled file that was created using a device type (DEVTYPE) of \*SCS, \*IPDS, or \*AFPDS. The \*AFPDS must have been created on the IBM i operating system.



### Electronically Stored Form Overlay

 ANY TOWN AMERICA					
NAME-SERVICE ADDRESS				SERVICE PERIOD	
				FROM	
				TO	
ACCOUNT NO.					
RATE	METER READINGS		MULT.	K.W.H. USED	AMOUNT
	PRESENT	PREVIOUS			
CITY		STATE	FRANCHISE		TOTAL TAX
 PAY THIS AMOUNT					

### Variable Page Data

JOHN JONES			10/02/90		
1225 STONE STREET					
ANY TOWN, STATE			11/01/90		
65432					
1030-7817-2					
10	0134	1344	10	100	\$1.86
CITY		STATE	FRANCHISE		TOTAL TAX
 PAY THIS AMOUNT					\$4.35

Merge

### Finished Product

 ANY TOWN AMERICA					
NAME-SERVICE ADDRESS				SERVICE PERIOD	
JOHN JONES 1225 STONE STREET ANY TOWN, STATE 65432				FROM	
				TO	
ACCOUNT NO.					
1030-7817-2					
RATE	METER READINGS		MULT.	K.W.H. USED	AMOUNT
	PRESENT	PREVIOUS			
10	0134	1344	10	100	\$1.86
CITY		STATE	FRANCHISE		TOTAL TAX
 PAY THIS AMOUNT					\$4.35

RV/2H309-0

Using overlays and rotation

Rotation can occur for text and overlays. The system treats text and overlays separately. That is, text is rotated using the page rotation (PAGRTT) parameter of the printer file. Overlay rotation must be determined at the time the overlay is created. Overlays are not rotated by the PAGRTT parameter.

To use the contents of the same overlay with different degrees of rotation (0 and 90 degrees), you create the overlay twice, once for 0 degrees and once for 90 degrees.

Assuming that you want the text and the overlay to be read in the same direction, the overlay offset values (down and across) on the printer file need to change depending on the degree of rotation of the text (PAGRTT parameter). That is, an overlay created with 90 degrees rotation is most often used with text that is rotated 90 degrees.

**Note:**

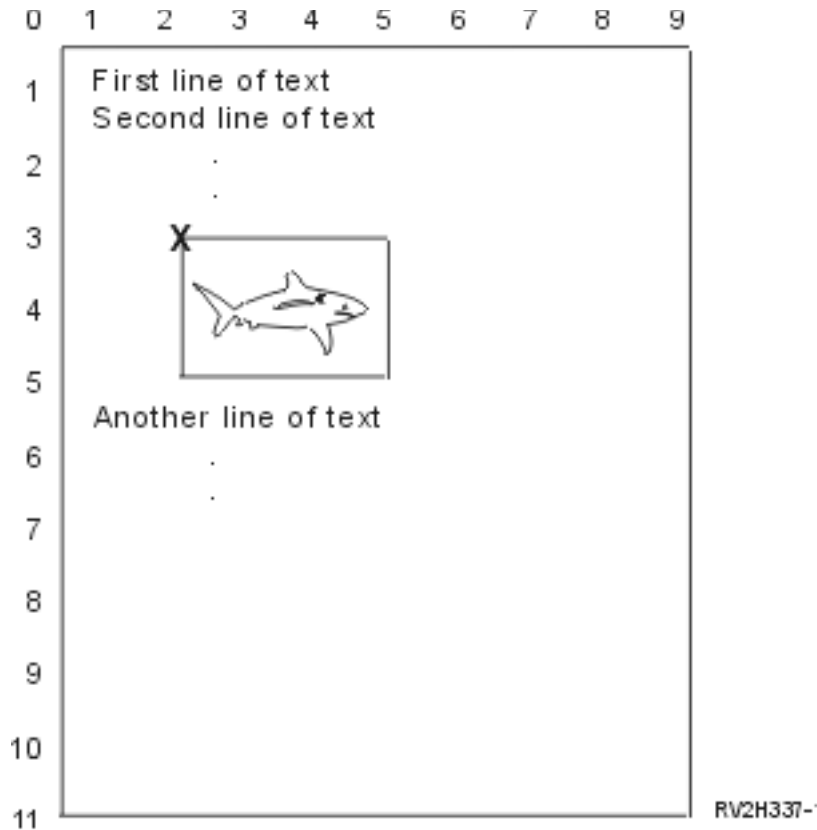
1. Determining the overlay parameter values is made easier by focusing on a particular corner of the overlay depending on the degree of rotation specified.
2. Keep in mind that paper does not rotate as it passes through the printer.
3. The page size (PAGESIZE) values (down, across) must be viewed differently depending on the page rotation value (PAGRTT) specified. The examples show this concept by having numbers along the edge of the page.

Following are examples of how to determine the overlay parameter values when page rotation (PAGRTT) is used.

***Example: Determining Overlay Values with Page Rotation (PAGRTT) of 0 Degrees***

The following example assumes:

- Page size (PAGESIZE) parameter values are (11,9)
- Page rotation (PAGRTT) parameter value is (0)



When PAGRTT is 0, the focus point for determining the overlay parameter values is the top left of the overlay (indicated by the X in the example).

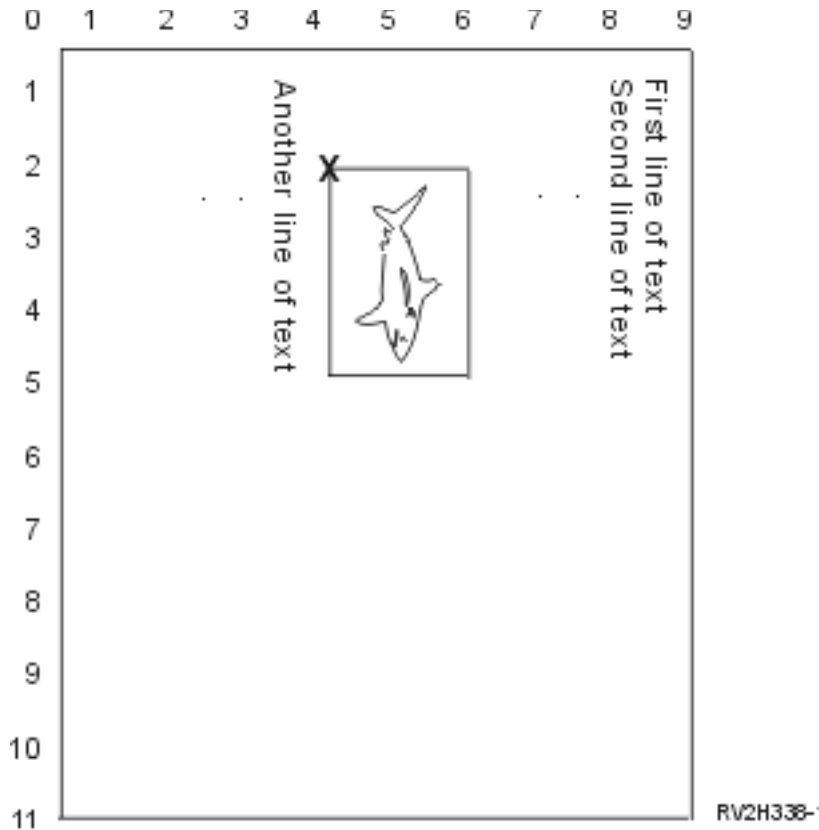
Use the numbers beside the example page to determine the overlay down (3) and across (2) values.

Overlay parameter values are (3,2).

***Example: Determining Overlay Values with Page Rotation (PAGRTT) of 90 Degrees***

The following example assumes:

- Page size (PAGESIZE) parameter values are (9,11)
- Page rotation (PAGRTT) parameter value is (90)



When PAGRTT is 90, the focus point for determining the overlay parameter values is the bottom left of the overlay (indicated by the X in the example).

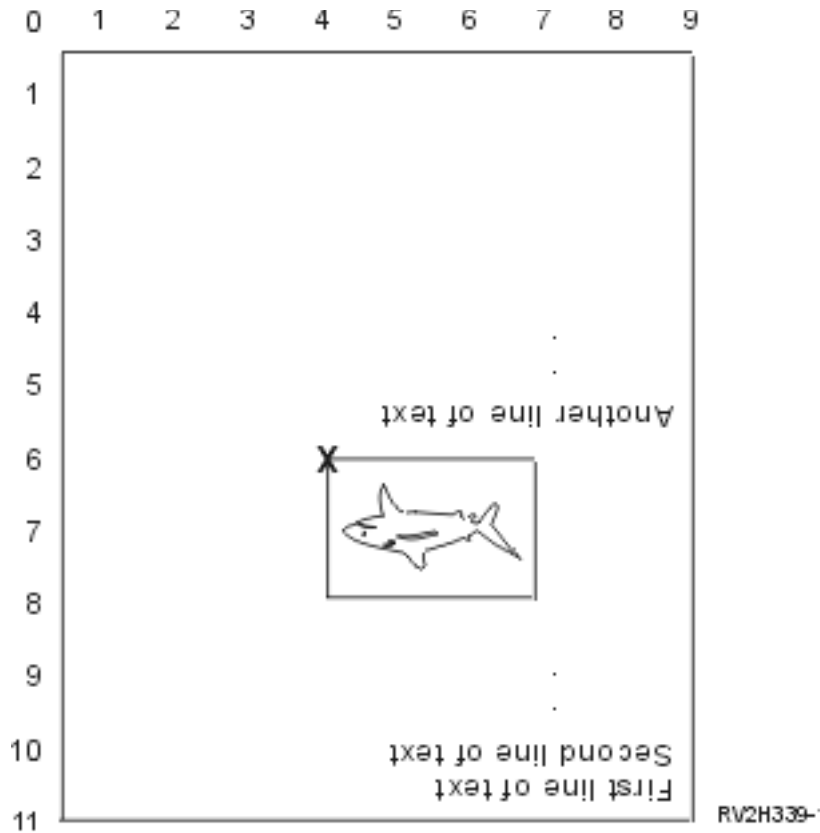
Use the numbers beside the example page to determine the overlay down (2) and across (4) values.

Overlay parameter values are (2,4).

***Example: Determining Overlay Values with Page Rotation (PAGRTT) of 180 Degrees***

The following example assumes:

- Page size (PAGESIZE) parameter values are (11,9)
- Page rotation (PAGRTT) parameter value is (180)



When PAGRTT is 180, the focus point for determining the overlay parameter values is the bottom right of the overlay (indicated by the X in the example).

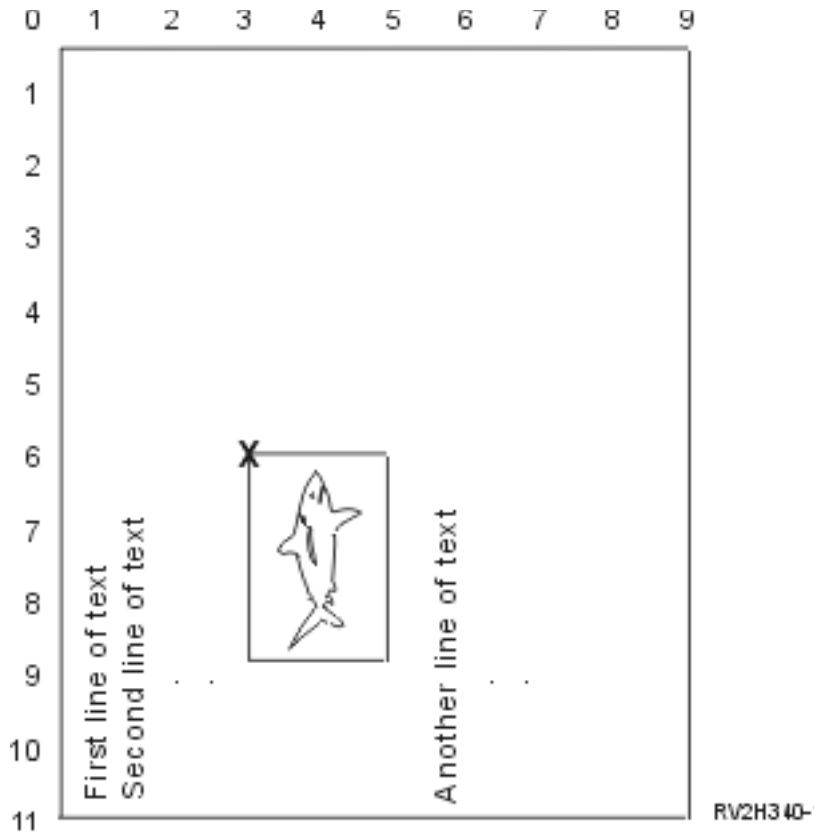
Use the numbers beside the example page to determine the overlay down (6) and across (4) values.

Overlay parameter values are (6,4).

***Example: Determining Overlay Values with Page Rotation (PAGRTT) of 270 Degrees***

The following example assumes:

- Page size (PAGESIZE) parameter values are (9,11)
- Page rotation (PAGRTT) parameter value is (270)



When PAGRTT is 270, the focus point for determining the overlay parameter values is the top right of the overlay (indicated by the X in the example).

Use the numbers beside the example page to determine the overlay down (6) and across (3) values.

Overlay parameter values are (6,3).

**Related information:**

 [AFP Utilities for iSeries User's Guide PDF](#)

**Page rotation (PAGRTT) parameter**

The PAGRTT parameter controls the rotation of text on the page. When \*AUTO or \*DEVD is specified for the PAGRTT parameter, the system determines the orientation of the printed page. For example, if the output is too wide, the page is rotated 90 degrees.

The system automatically senses when the output is too large to fit on the paper that is loaded in the printer for the following models: 3831, 3835, 3900, InfoPrint 3000, and InfoPrint 4000 printers. It also senses whether the paper is larger than 8 1/2 inches by 14 inches. In the information below, the expression "too large to fit on the form" refers to a page size that is defined in the printer file parameters. If both the length and width exceed 8.5 inches, or if the length or width is greater than 14 inches, the information is too large for the form. For example, a page defined to be 13.2 inches-wide and 11 inches long is too large for a form that is 8.5 inches-wide and 14 inches-long. This can be a candidate for computer output reduction (COR). This applies for printers other than the 3831, 3835, 3900, InfoPrint 3000, and InfoPrint 4000.

## **PAGRTT = \*AUTO:**

If PAGRTT = \*AUTO, then these are true.

- If the output is too large to fit on the form, computer output reduction is performed automatically.
- If the page is wider than it is long, the page is rotated 90 degrees. If it is not wider than it is long, the page is not rotated.
- \*AUTO is only supported for spooled files with a device type (DEVTYPE) of \*SCS, \*IPDS, or \*AFPDS that is created on the IBM i operating system. It is not supported for files with a device type of \*USERASCII, \*LINE, \*AFPDSLIN, or \*AFPDS that is not created on the IBM i operating system.

## **PAGRTT = \*DEVD:**

If PAGRTT = \*DEVD, then these are true.

- If the output is too large to fit on the form, computer output reduction is performed automatically.
- If the page is wider than it is long, the page is rotated 90 degrees. If it is not wider than it is long, the page is not rotated.
- For computer output reduction printing, the PRTQLTY (print quality) parameter value must be either \*DRAFT or \*DEVD. If the PRTQLTY parameter value is \*STD or \*NLQ, spooled files are printed without computer output reduction (COR) and without page rotation.
- When using the PAGRTT parameter, do not specify \*DEVD for the FONT parameter value. When FONT(\*DEVD) is used, the system cannot determine the exact page width; therefore, page positioning might not be as intended.

**Note:** When the PAGRTT parameter value is \*AUTO or \*DEVD, computer output reduction (COR) is not provided if the spooled file has any of the Device Requirements fields set to Y. Use the Work with Spooled Files (WRKSPLF) command and select option 8=Attributes to view the device requirements for any spooled file.

## **PAGRTT = 0, 90, 180, or 270 degrees:**

If PAGRTT = 0, 90, 180, or 270 degrees, then these are true.

When these values (0, 90, 180, or 270) are specified for the PAGRTT parameter, the page size (PAGESIZE) parameter values are not automatically changed. You must specify the PAGESIZE parameter values with reference to the way the data is printed on the page. For example, using forms that are 8.5 inches wide by 11 inches long and printing at 6 lines per inch with a 10-pitch font:

- Specify PAGESIZE(66 85) with PAGRTT(0) or PAGRTT(180).  
The page reads top to bottom with the 8.5-inch side at the top (portrait orientation).
- Specify PAGESIZE(51 110) with PAGRTT(90) or PAGRTT(270).  
The page reads top to bottom with the 11-inch side at the top (landscape orientation).

## **PAGRTT = \*COR:**

If PAGRTT = \*COR, then these are true.

- Output is rotated 90 degrees.
- Page size is set to 11 by 8.5 inches.
- Font substitution occurs as follows:
  - 12-pitch fonts are replaced with 15-pitch fonts.
  - 15-pitch fonts are replaced with 20-pitch fonts.
  - All other fonts are replaced with a 13.3-pitch font (with the exception of the 4028 printer, which uses a 15-pitch font).

**Note:** When the PAGRTT parameter value is \*COR, computer output reduction is not provided, if the spooled file is \*AFPDS, was created on the IBM i operating system, and has any of the Device Requirements fields set to Y.

Use the Work with Spooled Files (WRKSPLF) command and select option 8=Attributes to view the device requirements for any spooled file. In addition, \*COR is not supported for spooled files with any of the following device types:

- \*USERASCII
- \*LINE (if the spooled file is sent to a printer configured with AFP(\*YES))
- \*AFPDSLIN
- \*AFPDS (if the spooled file was not created on the IBM i operating system)

**Page rotation on the 3831, 3835, 3900, InfoPrint 3000, and InfoPrint 4000 printers:**

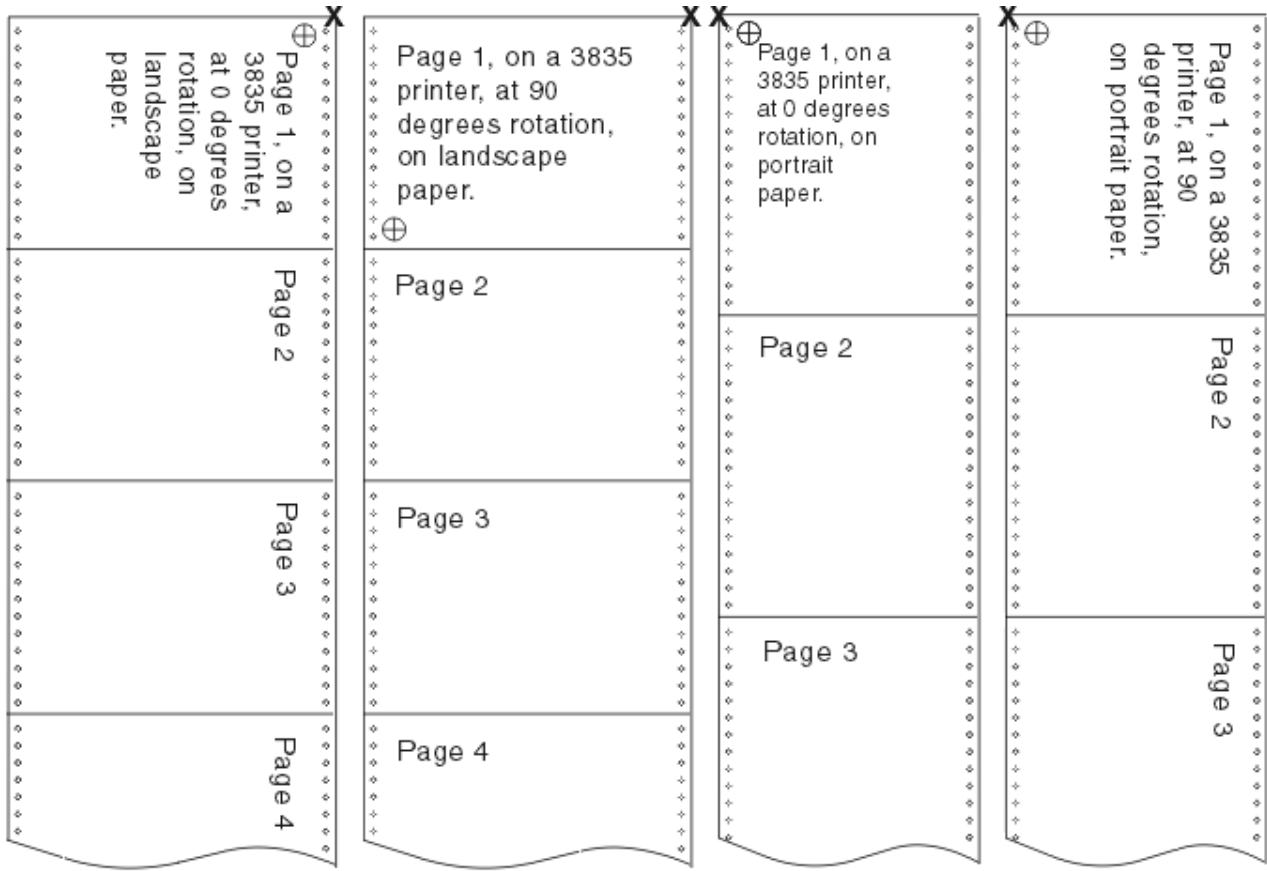
The 3831, 3835, 3900, InfoPrint 3000, and InfoPrint 4000 printers are continuous-forms printers that can accept page rotation commands.

Paper that is wider than it is long has output that is printed at 90-degree rotation. The paper rotation is in a counter-clockwise direction. Normally output rotation is in a clockwise direction. See the following diagram for an example of how printed output appears on these printers when rotation occurs.

**Notes:**

1. The corner with the X is the physical paper origin. It is the left edge of the narrow side of the page.
2. The circle with the plus (+) sign in it designates the logical paper origin.





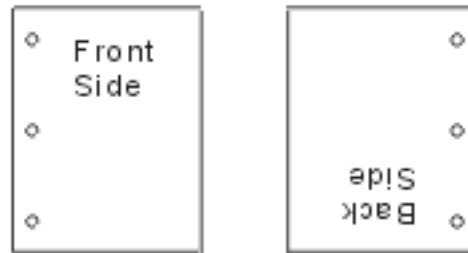
**Page rotation and duplex printing:**

This example illustrates how printed output will appear when combinations of duplex and page rotation are used. The dots on each page represent holes punched in the paper.

PAGRTT (0)  
DUPLEX (\*YES)



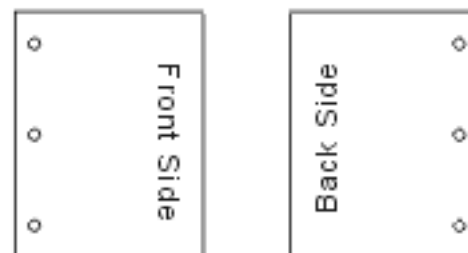
PAGRTT (0)  
DUPLEX (\*TUMBLE)



PAGRTT (90)  
DUPLEX (\*YES)



PAGRTT (90)  
DUPLEX (\*TUMBLE)



RM04310-1

**Note:** A page segment, overlay, or resource included using the DDS PAGSEG, OVERLAY, or AFPRSC keyword is not rotated with the rest of the printed output. This is also true of an overlay (front or back) specified on the printer file.

You must specify the degree of rotation when the page segment, overlay, or resource is created. Infoprint Designer for i5/OS can be used to create overlays and page segments.

### Saddle stitch (SADLSTITCH) parameter

This parameter specifies where to place the staples along the finishing margin of the media. In saddle stitching, the finishing margin is at the center of the media, and parallel to the reference edge. Page rotation does not affect the placement of a saddle stitch.

You need to check your printer documentation to determine which elements, and values of elements, are supported. Where your printer does not support any of the values for a specific element, specify a value of \*DEV D for the element.

#### Element 1: Reference Edge

Specifies which edge to use for the saddle stitch. Possible values are:

**\*DEV D**

The default that is used by the device.

**\*TOP** The reference edge is the top edge.

**\*LEFT** The reference edge is the left edge.

## Element 2: Number of Staples

Specifies the number of staples to use for saddle stitching. Possible values are:

### **\*DEV**

The default value for the device. This is the value that is used if \*DEV is also specified for the staple offsets value of this parameter. When specifying \*DEV for this and for the staple offset value, the number of staples will be the default that is used by the device. If you specify one or more offsets, the number of staples is the same as the number of staple offsets specified.

### **number of staples**

The valid value range is from 1 to 122 staples.

## Element 3: Staple Offsets

Specifies the distance between staples that are used in the edge stitching. If the staple placement will be on the left or right edge of the paper, the first staple offset is determined by measuring from the intersection of the finishing margin and the bottom edge of the paper to where the center of the staple will be. Subsequent staple offsets are measured from the same point (not from the previous staple). If the staple placement will be at the top or bottom edge of the paper, the first staple offset is determined by measuring from the intersection of the finishing margin and the left edge of the paper to where the center of the staple will be. Subsequent staple offsets are measured from the same point (not from the previous staple). Possible values are:

### **\*DEV**

The default staple offset that is used by the device. If you specify a value for the Number of staples, the printer will automatically calculate the position of each staple.

### **staple offset**

The valid value range is from 1 to 122 staple offsets. If a staple offset is other than \*DEV, the Number of staples must be \*DEV. This measurement is in centimeters (range 0 through 57.79) or inches (range 0 through 22.57)

## Spooled output schedule (SCHEDULE) parameter

The SCHEDULE parameter can be specified with one of three values to control when the spooled file is available for a writer to produce the file.

### **\*FILEEND**

Specifies that the spooled file is made available to the writer when the file is closed

### **\*JOBEND**

Specifies that the spooled file is made available to the writer when the job that created the spooled file is ended

### **\*IMMED**

Specifies that the spooled file is made available to the writer when the file is opened

### **\*JOBEND considerations**

The SCHEDULE(\*JOBEND) spooled files of a job are grouped together on their output queues when the job completes. All SCHEDULE(\*JOBEND) spooled files of the same job that are grouped together on a queue are produced together by the writer. Another spooled file can be added to the top of the queue while the writer is producing a SCHEDULE(\*JOBEND) spooled file. After a writer produces one file of a job that is SCHEDULE(\*JOBEND), it checks the following file on the queue. If this file is from the same job and is also SCHEDULE(\*JOBEND), the writer produces it next. However, if the file is from a different job or is not SCHEDULE(\*JOBEND), the first file on the queue is produced next.

If you want your SCHEDULE(\*JOBEND) spooled files grouped together on a SEQ(\*FIFO) output queue, you must be careful not to separate the spooled files. File operations such as HLDSPLF, CHGSPLFA, and RLSSPLF are performed one at a time (even from a Work with Output Queue (WRKOUTQ) display). If operations to other files on the queue are done at the same time, your spooled files can be separated on a

SEQ(\*FIFO) queue. If your SCHEDULE(\*JOBEND) spooled files separate, you can regroup them by changing their output priority with the Change Job (CHGJOB) command.

### **\*IMMED considerations**

When a writer is producing a SCHEDULE(\*IMMED) spooled file, it might catch up to the program producing the output. When this happens, the writer must wait for the program to produce more output. Because of this, you should be careful using \*IMMED for the schedule option. When this happens, the writer cannot process other spooled files. Moreover, the device cannot be used for any other work.

Normally, spooled files that are created with SCHEDULE(\*IMMED) specified are assigned a smaller internal buffer than spooled files that are created with SCHEDULE(\*JOBEND) or SCHEDULE(\*FILEEND) specified. Using SCHEDULE(\*IMMED) to assign a smaller internal buffer might allow the spooled data to be produced sooner, but can also adversely affect performance because more disk operations are needed for the same amount of spooled data.

A large internal buffer is always used for spooled print files that use certain special device requirements. Special device requirements include:

- Defined characters
- Graphics 4214
- Graphics 4234
- Graphics 522x

Changing the SCHEDULE parameter of a spooled file with the CHGSPLFA command does not affect the internal buffers used for that file.

### **To stream file (TOSTMF) parameter**

You can use the TOSTMF parameter to cause the output data to be written to a stream file in the integrated file system. When the TOSTMF parameter is used with the Workstation Customization object (WSCST) parameter, the output data is transformed to another output data format before being written to the stream file.

You can specify either a directory or a stream file name for this parameter. All directories in the path name must exist. New directories are not created. If the TOSTMF value refers to a directory, the system will create a stream file in that directory with a unique name derived from the printer file name. If the TOSTMF value is a file name, a stream file by that name must not already exist.

The default value for this parameter is \*NONE. It causes the output to be written to a spooled file or directly to the printer device.

#### **Related reference:**

“Workstation customizing object (WSCST) parameter”

You can use the WSCST parameter to transform the printer file output to final form before writing it to a stream file. This parameter is only valid when the to stream file (TOSTMF) parameter is not set to \*NONE and the device type (DEVTYPE) parameter is set to \*AFPDS.

### **Workstation customizing object (WSCST) parameter**

You can use the WSCST parameter to transform the printer file output to final form before writing it to a stream file. This parameter is only valid when the to stream file (TOSTMF) parameter is not set to \*NONE and the device type (DEVTYPE) parameter is set to \*AFPDS.

The WSCST object specified must have an attribute of TRANSFORM or CTXFORM. If the WSCST object has an attribute of TRANSFORM, the host print transform function will be used to convert the data before writing it to the stream file.

If the WSCST object has an attribute of CTXFORM, the Portable Application Solutions Environment (PASE) and the IBM Transform Services for i5/OS licensed program (5761-TS1) must be installed. Three CTXFORM WSCST objects are included with i5/OS, which will transform the output data to the Adobe Portable Document Format (PDF). These objects require option 01 (AFP to PDF Transform) of the 5761-TS1 product to be installed. These objects differ in how they map AFP font references to PDF font references.

#### **QSYS/QCTXPDPDF**

This is the default PDF WSCST object that is used when the special value \*PDF is specified. It includes tags to map all of the CJK World Type fonts included in i5/OS option 43 to corresponding registered CMaps and character collections. This allows ideographic characters to be rendered without embedding these very large fonts within the document. The smaller World Type fonts are not mapped by this object and will be embedded within the output document to allow non-Latin1 SBCS languages to be presented.

#### **QSYS/QCTXPDFWT**

This object contains tags to map all of the World Type fonts included with i5/OS option 43 to corresponding Type 1 fonts or registered CMaps and character collections. This generally results in the smallest possible PDF file size, but only reliably presents Latin and CJK languages.

#### **QSYS/QCTXPDFEMBD**

This is the minimal PDF WSCST object included. It omits all font tags, so that the default font mapping is used for all font references. The default behavior is to map all SBCS AFP font references to standard PDF fonts, map all DBCS AFP fonts to common CID fonts, and embed all TrueType font references.

#### **Related concepts:**

“Customizing WSCST object of \*CTXFORM transform type” on page 196

You can modify the way the PDF transform maps fonts by customizing your WSCST object of \*CTXFORM transform type.

#### **Related reference:**

“To stream file (TOSTMF) parameter” on page 284

You can use the TOSTMF parameter to cause the output data to be written to a stream file in the integrated file system. When the TOSTMF parameter is used with the Workstation Customization object (WSCST) parameter, the output data is transformed to another output data format before being written to the stream file.

## **Printer file return codes**

The major and minor return codes for printer files are set in the I/O feedback area of the printer file.

Return codes report the results of each operation. The appropriate return code is available to the application program that issued the operation. The program then checks the return code and acts appropriately. Refer to your high-level language documentation for information about how to access these return codes.

The return code is a 4-digit value: the first two digits contain the major code, and the last 2 digits contain the minor code. With some return codes, a message is also sent to the job log or the system operator message queue (QSYSOPR). You can refer to the message for additional information. Message IDs followed by an asterisk (\*) can be received by applications while spooling output.

**Note:** Return codes that refer to a condition on the printer are available to the application program only when printing with SPOOL = \*NO specified in the printer file. When SPOOL = \*YES has been specified, the printer writer program is the program communicating with the printer, not your application program.

## Major code 00 (00xx)

Here are the description, action, and messages of Major code 00 (00xx).

**Major Code 00** – Operation completed successfully.

**Description:** The operation issued by your program completed successfully.

**Action:** Continue with the next operation.

**0000 Description:** For output operations performed by your program, 0000 indicates that the last output operation completed successfully.

The notify messages are used after certain error conditions to give the operator the choice of continuing or canceling the printing of that file. If the reply is CANCEL, another message is issued with a nonzero return code.

**Action:** Your program can continue. One of the following diagnostic messages might have been issued to warn of an unusual condition that might be significant to your program even though it is not an error.

### Messages:

**CPA4001 (Inquiry)**

CPA5341 (Inquiry)

**CPA4003 (Inquiry)**

CPA5342 (Inquiry)

**CPA4004 (Inquiry)**

CPA5343 (Inquiry)

**CPA4005 (Inquiry)**

CPA5344 (Inquiry)

**CPA4007 (Inquiry)**

CPA5347 (Inquiry)

**CPA4008 (Inquiry)**

CPA5348 (Inquiry)

**CPA4009 (Inquiry)**

CPD4005 (Diagnostic)

**CPA4010 (Inquiry)**

CPD4006 (Diagnostic)

**CPA4011 (Inquiry)**

CPD4007 (Diagnostic)

**CPA4012 (Inquiry)**

CPD4008 (Diagnostic)

**CPA4013 (Inquiry)**

CPD4069 (Diagnostic)

**CPA4014 (Inquiry)**

CPD4071 (Diagnostic)\*

**CPA4015 (Inquiry)**

CPD4072 (Diagnostic)

**CPA4017 (Inquiry)**

CPF4032 (Diagnostic)

**CPA4019 (Inquiry)**  
CPF4033 (Diagnostic)

**CPA4037 (Inquiry)**  
CPF4056 (Diagnostic)

**CPA4038 (Inquiry)**  
CPF4057 (Diagnostic)

**CPA4039 (Inquiry)**  
CPF4239 (Escape)

**CPA4040 (Inquiry)**  
CPF4245 (Escape)

**CPA4042 (Inquiry)**  
CPF4249 (Escape)

**CPA4043 (Inquiry)**  
CPF4260 (Escape)\*

**CPA4046 (Inquiry)**  
CPF4420 (Diagnostic)

**CPA4047 (Inquiry)**  
CPF4421 (Diagnostic)

**CPA4048 (Inquiry)**  
CPF4905 (Notify)\*

**CPA4065 (Inquiry)**  
CPF4913 (Diagnostic)

**CPA4066 (Inquiry)**  
CPF4914 (Diagnostic)

**CPA4072 (Inquiry)\***  
CPF4916 (Notify)\*

**CPA4073 (Inquiry)**  
CPF4918 (Notify)\*

**CPA4074 (Inquiry)**  
CPF4919 (Notify)\*

**CPA4075 (Inquiry)**  
CPI4015 (Informational)

**CPA4076 (Inquiry)**  
CPI4016 (Informational)

**CPA4251 (Inquiry)**  
CPI4017 (Informational)

**CPA4256 (Inquiry)**  
CPI4018 (Informational)

**CPA5335 (Inquiry)**  
CPI4019 (Informational)

**CPA5339 (Inquiry)**  
CPI4020 (Informational)

**CPA5340 (Inquiry)**  
CPI4024 (Informational)

## Major code 80 (80xx)

Here are the description, action, and messages of Major code 80 (80xx).

**Major Code 80** – Permanent system or file error (unrecoverable).

**Description:** An unrecoverable file or system error occurred. Recovery is unlikely until the problem causing the error has been corrected.

**Action:** The following general actions can be taken by your program for each 80xx return code. Other specific actions are given in each return code description.

- Continue processing without the printer.
- Close the printer file and open the file again.
- End.

**8081 Description:** The operation was not successful because a system error condition was detected.

**Action:** Your printer might need to be varied off and then on again. Your program can either:

- Continue processing without the printer.
- Close the device file and open the file again.
- End.

**Messages:**

**CPF4182 (Escape)\***

CPF5409 (Escape)

**CPF4289 (Escape)**

CPF5410 (Escape)

**CPF4510 (Escape)\***

CPF5414 (Escape)

**CPF4516 (Escape)**

CPF5416 (Escape)

**CPF4552 (Escape)**

CPF5418 (Escape)

**CPF4591 (Escape)**

CPF5423 (Escape)

**CPF5159 (Escape)**

CPF5429 (Escape)

**CPF5196 (Escape)**

CPF5431 (Escape)\*

**CPF5246 (Escape)**

CPF5433 (Escape)

**CPF5257 (Escape)\***

CPF5434 (Escape)

**CPF5261 (Escape)**

CPF5447 (Escape)

**CPF5262 (Escape)\***

CPF5453 (Escape)

**CPF5401 (Escape)**

CPF5507 (Escape)

**CPF5408 (Escape)**



**8082 Description:** The operation attempted was not successful because the printer is unusable. This might occur because:

- A cancel reply has been taken to an error recovery message for the device.
- A cancel reply was returned to a maximum records reached inquiry message.
- The printer has been held by a Hold Communications Device (HLDCMNDEV) command.

No operations should be issued to the device.

**Action:** Communications with the printer cannot be resumed until the device has been reset to a varied-on state. If the device has been held, use the Release Communications Device (RLSCMNDEV) command to reset the device. If the device is in an error state, vary the device off and then on again. Once the device is reset, normal operation can be started again by reopening the printer file.

**Messages:**

- CPF4502 (Escape)
- CPF5104 (Escape)
- CPF5116 (Escape)\*
- CPF5269 (Escape)

**80B3 Description:** The open operation was not successful because the printer file or printer device is not available.

**Action:** The printer file cannot be opened again until the necessary resources are available. Your program can wait for the resources to become available, and then issue another open operation. Otherwise, you can continue other processing or end the program. The Work with Configuration Status (WRKCFGSTS) command might be used to determine whether the printer device is in use or not varied on. If the device is in use, the WRKCFGSTS command will also identify the job that is using it.

Consider increasing the WAITFILE parameter with the Change Printer File (CHGPRTF) or Override with Printer File (OVRPRTF) command to allow more time for the file resources to become available.

**Messages:**

- CPF4128 (Escape)\*
- CPF9808 (Diagnostic)\*

**80C0 Description:** An unrecoverable error has occurred on the printer device.

**Action:** Your printer might need to be varied off and then on again. Your program can either:

- Continue processing without the printer.
- Close the printer file and open the file again.
- End.

**Messages:**

**CPF4262 (Escape)**  
CPF5413 (Escape)

**CPF4509 (Escape)**  
CPF5419 (Escape)

**CPF5103 (Escape)**  
CPF5420 (Escape)

**CPF5247 (Escape)**  
CPF5430 (Escape)

**CPF5412 (Escape)**

CPF5437 (Escape)

**80EB Description:** An open operation was not successful because an open option that was not valid or an invalid combination of options was specified in your program, in the printer file, or in an override command.

**Action:** Close the printer file, correct the problem, and issue the open operation again. See the individual messages to determine what options are not valid.

**Messages:****CPD4012 (Diagnostic)**

CPF4209 (Escape)

**CPD4013 (Diagnostic)\***

CPF4214 (Escape)\*

**CPD4020 (Diagnostic)**

CPF4217 (Escape)

**CPD4021 (Diagnostic)\***

CPF4219 (Escape)

**CPD4023 (Diagnostic)**

CPF4224 (Escape)

**CPD4024 (Diagnostic)**

CPF4237 (Escape)\*

**CPD4025 (Diagnostic)**

CPF4238 (Escape)

**CPD4033 (Diagnostic)**

CPF4263 (Escape)\*

**CPD4034 (Diagnostic)\***

CPF4295 (Escape)\*

**CPD4036 (Diagnostic)\***

CPF4296 (Escape)\*

**CPD4037 (Diagnostic)\***

CPF4335 (Escape)

**CPD4038 (Diagnostic)\***

CPF4336 (Escape)

**CPF411E (Escape)**

CPF4264 (Escape)\*

**CPF4133 (Escape)**

CPF4337 (Escape)

**CPF4138 (Escape)\***

CPF4338 (Escape)

**CPF4139 (Escape)\***

CPF4339 (Escape)\*

**CPF4148 (Escape)**

CPF4340 (Escape)

**CPF4156 (Escape)**

CPF4345 (Escape)

**CPF4157 (Escape)\***  
CPF4352 (Escape)

**CPF4159 (Escape)\***  
CPF4637 (Escape)

**CPF4162 (Escape)**  
CPF5370 (Escape)

**CPF4181 (Escape)\***

**CPF4196 (Escape)\***

**CPF4206 (Escape)\***

**80ED Description:** An open operation was not successful because the record format descriptions in the printer file have changed since your program was compiled.

**Action:** Close the printer file and end the program. Determine whether the changes affect your application program. If they do, then recompile the program. If the changes do not affect your program, the file should be changed or overridden to LVLCHK(\*NO). When LVLCHK(\*NO) is specified, the system does not compare the record format descriptions.

**Messages:**

- CPF4131 (Escape)\*

**80EF Description:** An open operation was not successful because your program is not authorized to the printer device.

**Action:** Close the file, correct the problem, and then issue the open operation again. Obtain authority to the device from your security officer or the device owner.

**Messages:**

- CPF4104 (Escape)\*

**80F8 Description:** An operation was not successful because the file is marked in error.

**Action:** Close the file. Refer to messages in the job log to determine what errors occurred. Take the appropriate recovery action for those errors.

**Messages:**

- CPF4132 (Escape)\*
- CPF5129 (Escape)\*
- CPF5293 (Escape)\*
- CPF5427 (Escape)\*

## Major code 81 (81xx)

Here are the description, action, and messages of Major code 81 (81xx).

**Major Code 81 – Permanent device error (unrecoverable).**

**Description:** An unrecoverable device-related error occurred during an I/O operation. Any attempt to continue using this printer device will probably fail again until the cause of the problem is found and corrected.

**Action:** The following general actions can be taken for each 81xx return code. Other specific actions are given in each return code description.

- Continue processing without the printer device.
- Close the file, correct the problem, and open the file again. If the operation is still unsuccessful, try it again only a limited number of times. (The number of times should be specified in your program.)
- End.

Several return codes indicate that an error condition must be corrected by varying the device off and on again.

**8181 Description:** A system error condition was detected during an I/O operation to the printer device.

**Action:** Close the file. You might need to vary the device off and on again to clear the error. Determine the cause of the failure from the accompanying message. Check for any system operator messages indicating that additional corrective action must be performed. Open the file again to continue.

**Messages:**

- CPF4289 (Escape)
- CPF4552 (Escape)
- CPF4553 (Escape)
- CPF5105 (Escape)
- CPF5159 (Escape)
- CPF5507 (Escape)

**8191 Description:** The operation was not successful because a permanent line error occurred, and the system operator took a recovery option in response to the line error message. (You can find out what type of line error occurred by asking the system operator.) The device has been marked unusable.

**Action:** Close the file. Vary the device off and on again to clear the error. Open the file again to continue.

**Messages:**

- CPF4146 (Escape)
- CPF4193 (Escape)
- CPF4526 (Escape)
- CPF4542 (Escape)
- CPF5128 (Escape)
- CPF5198 (Escape)

**8197 Description:** An unrecoverable error condition was detected at the device.

**Action:** Close the file. Vary the device off and on again to clear the error. Refer to the accompanying error message for additional information regarding the source of the specific error detected. Open the file again to continue.

**Messages:**

- CPF4149 (Escape)**  
CPF4583 (Escape)
- CPF4192 (Escape)**  
CPF5106 (Escape)
- CPF4197 (Escape)**  
CPF5143 (Escape)
- CPF4216 (Escape)**  
CPF5199 (Escape)
- CPF4524 (Escape)**  
CPF5201 (Escape)
- CPF4533 (Escape)**  
CPF5268 (Escape)
- CPF4538 (Escape)**  
CPF5360 (Escape)

**81C2** **Description:** The operation issued by your program was not successful because the Systems Network Architecture (SNA) session with the printer is not active.

**Action:** Close the file. Vary the device off and on again to clear the error. Open the file again to continue.

**Messages:**

- CPF5422 (Escape)

## Major code 82 (82xx)

Here are the description, action, and messages of Major code 82 (82xx).

**Major Code 82 – Open operation failed.**

**Description:** An attempt to open the printer file was not successful. The error might be recoverable or permanent, but is limited to the printer device. Recovery is unlikely until the problem causing the error has been corrected.

**Action:** The following general actions can be taken for each 82xx return code. Other specific actions are given in each return code description. You can either:

- Continue processing without the device.
- Close the file, correct the problem, and open the file again. A subsequent operation can be successful if the error occurred because of some temporary condition such as the device being in use at the time.  
If the operation is still unsuccessful, try it again only a limited number of times. (The number of times should be specified in your program.)
- End.

Several return codes indicate that an error condition must be corrected by changing a value in the file. To change a parameter value for the file, use the Change Printer File (CHGPRTF) or the Override with Printer File (OVRPRTF) command.

**8281** **Description:** A system error condition was detected on an open operation that was not successful. The printer file might previously have been in error, or the printer file might not be opened due to a system error.

**Action:** Your printer might need to be varied off and then on again to clear the error. Your program can either:

- Continue processing without the printer.
- Close the file, correct the problem, and open the file again.
- End.

Determine the cause of the failure from the accompanying message.

**Messages:**

- CPF4168 (Escape)\*

**8282** **Description:** The open operation was not successful because the printer device is unusable. This might occur because a cancel reply has been taken to an error recovery message for the printer or because the printer has been held by a Hold Communications Device (HLDCMNDEV) command. No operations should be issued to the device.

**Action:** Close the file. Communications with the printer cannot be resumed until the device has been reset to a varied-on state. If the device has been held, use the Release Communications Device (RLSCMNDEV) command to reset the device. If the device is in an error state, vary the device off and then on again. After the device is reset, normal operation can be started by opening the printer device file again.

**Messages:**

- CPF4110 (Escape)
- CPF4298 (Escape)

- CPF4354 (Escape)

**8291 Description:** A permanent line error occurred during an open operation. The printer device has been marked unusable.

**Action:** Close the file. Vary the device off and on again to clear the error. Open the file again to continue.

**Messages:**

- CPF4179 (Escape)
- CPF4291 (Escape)

**82A6 Description:** The open operation failed because of a protocol violation with Systems Network Architecture (SNA).

**Action:** Ensure that the printer with which your program is communicating is configured properly. Refer to the device response codes in the accompanying error message for additional information regarding the specific error detected.

**Messages:**

**CPF4124 (Escape)**

CPF4533 (Escape)

**CPF4190 (Escape)**

CPF5103 (Escape)

**CPF4192 (Escape)**

CPF5143 (Escape)

**CPF4527 (Escape)**

CPF5453 (Escape)

**82AA Description:** The open operation was not successful because the printer device description was not found.

**Action:** Your program can continue without the printer, attempt to use a different printer, or end.

Verify that the name of the printer was correctly specified in the DEV parameter on the CRTPRTE, CHGPRTE, OVRPRTE, or CRTPRTE command.

**Messages:**

- CPF4103 (Escape)\*

**82B3 Description:** The open operation was not successful because the printer you requested is in use in another file in your job.

**Action:** Close both of the printer device files, then open the one that you want to use again.

**Messages:**

- CPF4106 (Escape)

**82EE Description:** An open operation was attempted to a device that is not supported for a printer file. Your program is attempting to open a device that is not a valid printer.

**Action:** Your program can continue without the printer, attempt to use a different printer, or close the file and end.

Verify that the name of the printer was specified correctly on the CHGPRTE or OVRPRTE command.

**Messages:**

- CPF4105 (Escape)

**82EF** **Description:** An open operation was attempted for a device that the user is not authorized to or that is in service mode.

**Action:** Your program can continue without the printer, attempt to use a different printer, or end. Close the file, correct the problem, and then issue the open operation again.

For authority errors, obtain authority to the device from your security officer or device owner. If the device is in service mode, the system service tools (SST) function is currently using the device. Wait until the device is available to issue the operation again.

**Messages:**

- CPF4104 (Escape)\*
- CPF4186 (Escape)
- CPF9802 (Diagnostic)\*

### Major code 83 (83xx)

Here are the description, action, and messages of Major code 83 (83xx).

<p><b>Major Code 83</b> – Device error occurred (recoverable).</p> <p><b>Description:</b> An error occurred during an I/O operation, but the printer device is still usable. Recovery within your program might be possible.</p> <p><b>Action:</b> The following general actions can be taken for each 83xx return code. Other specific actions are given in each return code description.</p> <ul style="list-style-type: none"><li>• Continue processing without the printer device.</li><li>• Correct the problem and continue processing with the printer device. If the attempt to recover from the operation is unsuccessful, try it again only a limited number of times. (The number of times should be specified in your program.)</li><li>• End.</li></ul> <p>Several return codes indicate that an error condition must be corrected by changing a value in the file. To change a parameter value for the file, use the Change Printer File (CHGPRTF) or Override with Printer File (OVRPRTF) command.</p>
---

**8319** **Description:** A negative response was received to the last printer operation attempted by your program. The error might have been caused by the user pressing the Cancel key on the printer.

**Action:** Your program can try a different operation, or close the file and end. Refer to the device response code in the accompanying message to determine why the operation was rejected. Correct the error in your program before attempting to try the operation again.

**Messages:**

- CPF4158 (Escape)
- CPF4531 (Escape)
- CPF5050 (Escape)

**831D** **Description:** The operation just attempted by your program was rejected because a parameter was not valid, was out of limits, or was missing.

**Action:** Your program can bypass the failing step and continue, or close the file and end. Refer to the accompanying message to determine what parameter was incorrect. Correct the error in your program before attempting to try the operation again.

**Messages:**

- CPD4016 (Diagnostic)\***  
CPF5275 (Escape)

**CPD4017 (Diagnostic)\***  
CPF5276 (Escape)\*

**CPD4027 (Diagnostic)\***  
CPF5288 (Escape)\*

**CPD4028 (Diagnostic)\***  
CPF5289 (Escape)\*

**CPD4029 (Diagnostic)\***  
CPF5324 (Escape)\*

**CPD4030 (Diagnostic)\***  
CPF5359 (Escape)\*

**CPD4041 (Diagnostic)\***  
CPF5363 (Escape)

**CPF4909 (Notify)\***  
CPF5366 (Escape)\*

**CPF5108 (Escape)\***  
CPF5367 (Escape)\*

**CPF5148 (Escape)\***  
CPF5368 (Escape)\*

**CPF5273 (Escape)\***

**831E Description:** The operation just issued by your program was not valid or an invalid combination of operations was specified.

**Action:** Your program can bypass the invalid operation and continue, or close the file and end. Refer to the accompanying message to determine why the operation was rejected. Correct the error in your program before attempting to try the failing operation again.

**Messages:**

**CPD4015 (Diagnostic)\***  
CPF5290 (Escape)\*

**CPD4018 (Diagnostic)\***  
CPF5320 (Escape)\*

**CPD4031 (Diagnostic)**  
CPF5321 (Escape)\*

**CPF4915 (Notify)\***  
CPF5322 (Escape)\*

**CPF5149 (Escape)\***  
CPF5323 (Escape)\*

**CPF5185 (Escape)\***  
CPF5325 (Escape)\*

**CPF5245 (Escape)\***  
CPF5362 (Escape)\*

**831F Description:** A length that is not valid was specified on the operation.

On an output operation, your program has tried to send a data record having a length that exceeds the maximum record length allowed for the printer or the page size defined for the file. If you are using direct I/O, you have exceeded the maximum number of bytes allowed per page. The data has been truncated.



**Action:** Issue the output operation again with a smaller output length. The record length for a program-described printer file cannot exceed the page size. The record length for any printer file must be no greater than 32 767 characters.

**Messages:**

- CPF4906 (Notify)\*
- CPF5160 (Escape)

**8343 Description:** The designated page overflow line number has been reached.

**Action:** Your program should take whatever application-dependent action is appropriate. This might include printing page totals or a running foot line.

**Messages:**

- CPF5004 (Status)\*

**83E0 Description:** Your program attempted to issue an operation using a record format that was not defined for the printer file, or the program omitted the record format name.

**Action:** Check the name of the record format in your program to be sure it is correct. Then check that the record format is defined properly in the DDS for the file.

**Messages:**

- CPF5186 (Escape)\*
- CPF5187 (Escape)\*

**83F6 Description:** Your program sent invalid data to the printer. The data type might be incorrect for the field in which it is used.

**Action:** Check the name of the record format in your program to be sure it is correct. Verify that the data definition statements in your program match the output record defined in the DDS for the file. Correct the error in your program before attempting to repeat the failing operation.

**Messages:**

**CPD4014 (Diagnostic)\***  
CPF5075 (Notify)\*

**CPD4022 (Diagnostic)\***  
CPF5234 (Escape)\*

**CPD4026 (Diagnostic)\***  
CPF5246 (Escape)

**CPD4035 (Diagnostic)\***  
CPF5261 (Escape)

**CPD4516 (Informational)**  
CPF5297 (Escape)\*

**CPD4591 (Escape)**  
CPF5364 (Escape)

**CPF4634 (Escape)**  
CPF5365 (Escape)

**CPF4635 (Escape)**  
CPF5369 (Escape)

**CPF4636 (Escape)**  
CPF5372 (Escape)

**CPF4642 (Escape)**  
CPF5373 (Escape)

**CPF4643 (Escape)**  
CPF5374 (Escape)

**CPF4644 (Escape)**  
CPF5375 (Escape)

**CPF4645 (Escape)**  
CPF5376 (Escape)

**CPF4646 (Escape)**  
CPF5377 (Escape)

**CPF4647 (Escape)**  
CPF5411 (Escape)

## Printer file feedback areas

This topic describes the open and I/O feedback areas associated with any opened printer file.

The following information is presented for each item in the feedback areas:

- Offset, which is the number of bytes from the start of the feedback area to the location of each item
- Data type
- Length, which is given in number of bytes
- Contents, which is the description of the item and the valid values for it

The support provided by the high-level language you are using determines how to access this information and how the data types are represented. See your high-level language manual for more information.

### Open feedback area for printer

The open data path (ODP) contains information about the merged file attributes and information returned by input or output operations. The ODP only exists while the file is open.

The open feedback area is the part of the open data path (ODP) that contains general information about the file after it has been opened. It also contains file-specific information, depending on the file type, plus information about each device defined for the file. This information is set during open processing and can be updated as other operations are performed.

Offset	Data type	Length in bytes	Contents
0	Character	2	Open data path (ODP) type: <b>DS</b> Printer file not being spooled. Display, tape, intersystem communications function (ICF), save, or diskette file not being spooled. <b>SP</b> Printer or diskette file being spooled or inline data file. <b>ST</b> Printer file data is being written to an IFS streamed file.
2	Character	10	Name of the file being opened. If the ODP type is DS, this is the name of the device file or save file. If the ODP type is SP, this is the name of the device file or the inline data file.
12	Character	10	Name of the library containing the file.
22	Character	10	Name of the spooled file. The name of a database file containing the spooled input or output records.

Offset	Data type	Length in bytes	Contents
32	Character	10	Name of the library in which the spooled file is located.
42	Binary	2	Spooled file number.
44	Binary	2	Maximum record length.
46	Character	2	Reserved.
48	Character	10	Member name: <ul style="list-style-type: none"> <li>• If the ODP type is SP, the member name in the file is named at offset 22.</li> </ul>
58	Binary	4	Reserved.
62	Binary	4	Reserved.
66	Binary	2	File type: 1      Display 2      Printer 4      Diskette 5      Tape 9      Save 10     DDM 11     ICF 20     Inline data 21     Database
68	Character	3	Reserved.
71	Binary	2	Number of lines on a printed page.
73	Binary	2	Number of positions on a printed line.
75	Binary	4	Not applicable to printer.
79	Binary	4	Spooled file number.
83	Character	10	Reserved.
93	Character	10	Reserved.
103	Binary	2	Not applicable to printer.
105	Binary	2	Maximum number of records that can be read or written in a block when using blocked record I/O.
107	Binary	2	Overflow line number.
109	Binary	2	Blocked record I/O record increment. Number of bytes that must be added to the start of each record in a block to address the next record in the block.
111	Binary	4	Reserved.

Offset	Data type	Length in bytes	Contents
115	Character	1	<p>Miscellaneous flags.</p> <p><b>Bit 1:</b> Reserved.</p> <p><b>Bit 2:</b> File sharable</p> <p><b>0</b> File was not opened as sharable.</p> <p><b>1</b> File was opened as sharable (SHARE(*YES)).</p> <p><b>Bits 3-5:</b> Not applicable to printer.</p> <p><b>Bit 6:</b> Field-level descriptions</p> <p><b>0</b> File does not contain field-level descriptions.</p> <p><b>1</b> File contains field-level descriptions.</p> <p><b>Bit 7:</b> DBCS-capable file</p> <p><b>0</b> File is not DBCS-capable.</p> <p><b>1</b> File is DBCS-capable.</p> <p><b>Bit 8:</b> Not applicable to printer.</p>
116	Character	10	Not applicable to printer.
126	Binary	2	File open count. If the file has not been opened as sharable, this field contains a 1. If the file has been opened as sharable, this field contains the number of programs currently attached to this file.
128	Binary	2	Reserved.
130	Binary	2	Not applicable to printer.
132	Character	1	Miscellaneous flags.

Offset	Data type	Length in bytes	Contents
			<p><b>Bits 1-4:</b> Not applicable to printer.</p> <p><b>Bit 5:</b> Separate indicator area</p> <p><b>0</b> Indicators are in the I/O buffer of the program.</p> <p><b>1</b> Indicators are not in the I/O buffer of the program. The DDS keyword, INDARA, was used when the file was created.</p> <p><b>Bit 6:</b> User buffers</p> <p><b>0</b> System creates I/O buffers for the program.</p> <p><b>1</b> User program supplies I/O buffers.</p> <p><b>Bits 7-8:</b> Reserved.</p>
133	Character	2	Open identifier. The value is unique for a full (not shared) open operation of a file. It allows you to match this file to an entry on the associated data queue.
135	Binary	2	The field value is the maximum record format length, including both data and file-specific information, such as first-character forms control, option indicators, response indicators, source sequence numbers, and program-to-system data. If the value is zero, then use the field at offset 44.
137	Binary	2	CCSID — Coded Character Set Identifier.
139	Character	1	Reserved.
140	Binary	4	Offset to the extensions list. Number of bytes from the start of the open feedback area to the extensions offset table.  The open feedback extensions table occurs after the device name list in the ODP.
144	Binary	2	Number of devices defined for this ODP. For printers this always has a value of 1.
146	Character		Device name definition list. See Device definition list for a description of this array.

### Device definition list:

The device definition list part of the open feedback area is an array structure. Each entry in the array contains information about each device or communications session attached to the file.

The number of entries in this array is determined by the number at offset 146 of the open feedback area. The device definition list begins at offset 148 of the open feedback area. The offsets shown for it are from the start of the device definition list rather than the start of the open feedback area.

Offset	Data Type	Length in Bytes	Contents
0	Character	10	Program device name. For printer or diskette files being spooled, the value is *N. For database files, the value is DATABASE. For save files, the value is *NONE. For ICF files, the value is the name of the program device from the Add ICF Device Entry (ADDICFDEVE) or Override ICF Pgm Device Entry (OVRICFDEVE) command. For all other files, the value is the name of the device description.
10	Character	50	Reserved.
60	Character	10	Device description name. For printer or diskette files being spooled, the value is *N. For save files, the value is *NONE. For all other files, the value is the name of the device description.
70	Character	1	Device class. <b>hex 01</b> Display <b>hex 02</b> Printer <b>hex 04</b> Diskette <b>hex 05</b> Tape <b>hex 09</b> Save <b>hex 0B</b> ICF
71	Character	1	Device type. <b>hex 02</b> 5256 Printer <b>hex 0C</b> 5224/5225 printers <b>hex 0F</b> 5219 Printer <b>hex 10</b> 5583 Printer (DBCS) <b>hex 11 (DBCS)</b> 5553 Printer <b>hex 14</b> 3270 Printer <b>hex 21</b> 4234 (SCS) Printer <b>hex 22</b> 3812 (SCS) Printer <b>hex 23</b> 4214 Printer <b>hex 24</b> 4224 (IPDS) Printer <b>hex 25</b> 4245 Printer <b>hex 29</b> 5262 Printer <b>hex 30</b> 3812 (IPDS) Printer <b>hex 31</b> 4234 (IPDS) Printer <b>hex 32</b> IPDS printer, model unknown <b>hex 55</b> 6252 (SCS) Printer <b>hex 57</b> 4230 (IPDS) Printer <b>hex 63</b> 3935 (IPDS) Printer

Offset	Data Type	Length in Bytes	Contents
72	Binary	2	Not applicable to printer.
74	Binary	2	Not applicable to printer.
76	Character	2	Not applicable to printer.
78	Character	1	Not applicable to printer.
79	Character	1	Not applicable to printer.
80	Character	50	Reserved.

### Open feedback extensions list:

The open feedback extension list contains an array of offsets to new sections of the open feedback area.

Currently, only the first open feedback extension list entry is defined.

Offset	Data type	Length in bytes	Contents
0	Binary	4	Length of the open feedback extensions list.
4	Binary	4	Offset from the beginning of the open feedback area to the stream file name extension. This field is only valid when the open data path (ODP) type is ST.
8	Binary		Reserved.

### Stream file name open feedback extension:

The stream file name open feedback extension is an optional part of the open feedback area that describes the output path.

The stream file name is always expressed as an absolute path name (for example, '/home/username/mydir/myfile.pdf') even if a relative path was specified when the file was opened.

Offset	Data type	Length in bytes	Contents
0	Binary	2	CCSID of the stream file path name
2	Binary	4	Length in bytes of the path name
6	Binary	2	Offset from the beginning of the stream file name open feedback extension to the path string
	Char		Absolute path name of the stream file

### I/O feedback area

The results of I/O operations are communicated to the program using i5/OS messages and I/O feedback information.

The I/O feedback area is updated for every I/O operation unless your program is using blocked record I/O. In that case, the feedback area is updated only when a block of records is written. Some of the information reflects the last record in the block. Other information, such as the count of I/O operations, reflects the number of operations on blocks of records and not the number of records. See your high-level language manual to determine if your program uses blocked record I/O.

The I/O feedback area consists of two parts: a common area and a file-dependent area. The file-dependent area varies by the file type. This topic discusses device file type of printers only.

#### Common I/O feedback area

Offset	Data Type	Length in Bytes	Contents
0	Binary	2	Offset to file-dependent feedback area.
2	Binary	4	Write operation count. Updated only when a write operation completes successfully. For blocked record I/O operations, this count is the number of blocks, not the number of records.
6	Binary	4	Read operation count. Not applicable to printers.
10	Binary	4	Write-read operation count. Not applicable to printers.
14	Binary	4	Other operation count. Number of successful operations other than write, read, or write-read. Updated only when the operation completes successfully. This count includes force-end-of-data.
18	Character	1	Reserved.
19	Character	1	Current operation. <b>hex 05</b> Write or write block <b>hex 09</b> Force-end-of-data
20	Character	10	Name of the record format just processed, which is either: <ul style="list-style-type: none"> <li>Specified on the I/O request, or</li> <li>Determined by default or format selection processing</li> </ul>



Offset	Data Type	Length in Bytes	Contents
30	Character	2	Device class:  Byte 1: <b>hex 00</b> Database <b>hex 01</b> Display <b>hex 02</b> Printer <b>hex 04</b> Diskette <b>hex 05</b> Tape <b>hex 09</b> Save <b>hex 0B</b> ICF  Byte 2 (if byte 1 is 02 for printer): <b>hex 02</b> 5256 Printer <b>hex 0C</b> 5224/5225 printers <b>hex 0F</b> 5219 Printer <b>hex 10</b> 5583 Printer (DBCS) <b>hex 11</b> 5553 Printer (DBCS) <b>hex 14</b> 3270 Printer <b>hex 21</b> 4234 (SCS) Printer <b>hex 22</b> 3812 (SCS) Printer <b>hex 23</b> 4214 Printer <b>hex 24</b> 4224 (IPDS) Printer <b>hex 25</b> 4245 Printer <b>hex 29</b> 5262 Printer <b>hex 30</b> 3812 (IPDS) Printer <b>hex 31</b> 4234 (IPDS) Printer <b>hex 32</b> IPDS printer, model unspecified <b>hex 55</b> 6252 (SCS) Printer <b>hex 57</b> 4230 (IPDS) Printer <b>hex 63</b> 3935 (IPDS) Printer
32	Character	10	Device name. The name of the device for which the operation just completed. The device name is supplied only for printer, display, tape, diskette, and ICF files. For printer files being spooled, the value is *N. For printer files not being spooled, the value is the device description name.
42	Binary	4	Not applicable to printer.
46	Character	80	Reserved.
126	Binary	2	Not applicable to printer.

Offset	Data Type	Length in Bytes	Contents
128	Binary	2	For printers, the field value is the record format length, including first-character forms control, option indicators, source sequence numbers, and program-to-system data. If the value is zero, use the field at offset 42.
130	Character	2	Reserved.
132	Binary	4	Not applicable to printer.
136	Character	8	Reserved.

### I/O feedback area for printer files

Offset	Data Type	Length in Bytes	Contents
0	Binary	2	Current line number in a page.
2	Binary	4	Current page count.
6	Character	1	Miscellaneous flags <b>Bit 1:</b> Spooled file deleted. <b>Bit 2–8:</b> Reserved.
7	Character	27	Reserved.
34	Character	2	Major return code. <b>00</b> Operation completed successfully <b>80</b> Permanent system or file error <b>81</b> Permanent device error <b>82</b> Open operation failed <b>83</b> Recoverable device error occurred
36	Character	2	Minor return code.

## AFP compatibility fonts and font substitution

This section contains information about Advanced Function Presentation (AFP) compatibility fonts, character identifiers and other printing characteristics as well as several tables to help you understand font substitution.

Font substitution is done by the i5/OS operating system when the application specifies a font ID that is not supported by the designated printer or cannot be downloaded from the system to the designated printer.

#### Notes:

1. i5/OS supports Font Object Content Architecture (FOCA) 2 font character sets. This means it does not support the font character sets used by the various models of the IBM 3800 printer. For more information about FOCA 2 fonts, see the Font Object Content Architecture (FOCA) Reference PDF.
2. Font global identifier (FGID) is used interchangeably with font ID in these tables.

#### Related information:

 [Font Object Content Architecture \(FOCA\) Reference PDF](#)

## Font-related examples

These examples help you get familiar with fonts and font substitution on the i5/OS operating system.

### Verify whether your font ID is supported by a certain printer

If you want to verify that a font ID is supported by a certain printer, refer to Printer font support. For example, locate font ID 112. Font ID 112 is supported by the 3812 and 3816 SCS and IPDS printers and the 4028 printer (this is indicated by blanks in those spaces). The 4019 printer supports font ID 112 on a font card resident in the 4019 printer. The 4224, 4234, and 5219 printers substitute font ID 86 or 87.

**Note:** A font card is a hardware card that can have many font character sets resident on it. Font cards can be installed in printers to provide additional fonts.

### Find out what font ID the system substitutes if the printer you want to use does not support your font ID

If your application uses a font ID that is not supported on all printers, you can determine the substitution by referring to Printer font support. For example, locate font ID 30. The table shows that font ID 30 is supported on the 3812 and 3816 SCS and IPDS printers. However, if you are using any of the other printers listed in the table, font ID 11 is substituted for font ID 30.

### Find out what font ID the system substitutes if your font ID is not available on the system or on the printer

For example, assume your application calls for font ID 4 and you want to print the spooled file on a 4224 printer. To determine if font ID 4 is a supported font or one that is substituted for, follow these steps:

1. Refer to Printer font support and see if your font ID is listed. Since font ID 4 is not listed, continue with the next step.
2. Refer to Font substitution by font ID range and see what font is substituted. Font ID 11 is substituted for fonts 0 through 65.
3. Refer to Printer font support and locate the substituted font ID. Font ID 11 is supported on the 4224 printer. The result of the font ID substitution is that your application will print using font ID 11.

#### Related reference:

“Printer font support” on page 317

This table contains font ID substitution information which identifies which fonts are supported by certain printers. For example, if your application specified a certain font ID that your printer did not support, you can find out which printers support that font and route your printed output to a printer that supports that font ID.

“Font substitution by font ID range” on page 330

If your application specifies a font ID (FGID) that is not found in Printer font support or is not resident in the printer (font card), the system makes a substitution based on the font ID ranges in the following table. For example, if font ID 4 is specified in your application, i5/OS substitutes font ID 11 as shown in the following table.

## Font information

This table contains the Advanced Function Presentation (AFP) compatibility fonts supported by the i5/OS operating system.

For additional information about font attributes and font types, see Font attributes and font types.

Font global identifier (FGID)	Name	Type of Font	Attributes	Point	Pitch (CPI)
2	Delegate	Uniformly spaced			10

Font global identifier (FGID)	Name	Type of Font	Attributes	Point	Pitch (CPI)
3	OCR-B	Uniformly spaced			10
5	Rhetoric/Orator	Uniformly spaced			
8	Scribe/Symbol	Uniformly spaced			10
10	Cyrillic 22	Uniformly spaced			10
11	Courier	Uniformly spaced			10
12	Prestige	Uniformly spaced			10
13	Artisan	Uniformly spaced			10
18	Courier Italic	Uniformly spaced	Italic		10
19	OCR-A	Uniformly spaced			10
20	Pica	Uniformly spaced			10
21	Katakana	Uniformly spaced			10
25	Presentor	Uniformly spaced			10
26	Matrix Gothic	Uniformly spaced			10
30	Symbol	Uniformly spaced			10
31	Aviv	Uniformly spaced			10
36	Letter Gothic	Uniformly spaced			10
38	Orator Bold	Uniformly spaced	Bold		10
39	Gothic Bold	Uniformly spaced	Bold		10
40	Gothic	Uniformly spaced			10
41	Roman Text	Uniformly spaced			10
42	Serif	Uniformly spaced			10
43	Serif Italic	Uniformly spaced	Italic		10
44	Katakana Gothic	Uniformly spaced			10

Font global identifier (FGID)	Name	Type of Font	Attributes	Point	Pitch (CPI)
46	Courier Bold	Uniformly spaced	Bold		10
49	Shalom	Uniformly spaced			10
50	Shalom Bold	Uniformly spaced	Bold		10
51	Matrix Gothic	Uniformly spaced			10
52	Courier	Uniformly spaced			10
55	Aviv Bold	Uniformly spaced	Bold		10
61	Nasseem	Uniformly spaced			10
62	Nasseem Italic	Uniformly spaced	Italic		10
63	Nasseem Bold	Uniformly spaced	Bold		10
64	Nasseem Italic Bold	Uniformly spaced	Bold and italic		10
66	Gothic	Uniformly spaced		12	
68	Gothic Italic	Uniformly spaced	Italic		12
69	Gothic Bold	Uniformly spaced	Bold		12
70	Serif	Uniformly spaced			12
71	Serif Italic	Uniformly spaced	Italic		12
72	Serif Bold	Uniformly spaced	Bold		12
74	Matrix Gothic	Uniformly spaced			12
75	Courier	Uniformly spaced			12
76	APL	Uniformly spaced			12
78	Katakana	Uniformly spaced			12
80	Symbol	Uniformly spaced			12
84	Script	Uniformly spaced			12
85	Courier	Uniformly spaced			12

Font global identifier (FGID)	Name	Type of Font	Attributes	Point	Pitch (CPI)
86	Prestige	Uniformly spaced			12
87	Letter Gothic	Uniformly spaced			12
91	Light Italic	Uniformly spaced	Italic		12
92	Courier Italic	Uniformly spaced	Italic		12
95	Adjutant	Uniformly spaced			12
96	Old World	Uniformly spaced			12
98	Shalom	Uniformly spaced			12
99	Aviv	Uniformly spaced			12
101	Shalom Bold	Uniformly spaced	Bold		12
102	Aviv Bold	Uniformly spaced	Bold		12
103	Nasseem	Uniformly spaced			12
109	Letter Gothic Italic	Uniformly spaced	Italic		12
110	Letter Gothic Bold	Uniformly spaced	Bold		12
111	Prestige Bold	Uniformly spaced	Bold		12
112	Prestige Italic	Uniformly spaced	Italic		12
154	Essay	Mixed pitch			12
155	Boldface Italic	Mixed pitch	Bold and italic		12
157	Title	Mixed pitch			12
158	Modern	Mixed pitch			12
159	Boldface	Mixed pitch	Bold		12
160	Essay	Mixed pitch			12
162	Essay Italic	Mixed pitch	Italic		12
163	Essay Bold	Mixed pitch	Bold		12
164	Prestige	Mixed pitch			12
167	Barak	Mixed pitch			12
168	Barak Bold	Mixed pitch	Bold		12
173	Essay	Mixed pitch			12
174	Gothic	Mixed pitch			12
175	Document	Mixed pitch			12
178	Barak	Mixed pitch			18

Font global identifier (FGID)	Name	Type of Font	Attributes	Point	Pitch (CPI)
179	Barak Bold	Mixed pitch	Bold		18
180	Barak	Mixed pitch			15
181	Barak Mixed Bold	Mixed pitch	Bold		15
182	Barak	Mixed pitch			5
183	Barak Bold	Mixed pitch	Bold		5
186	Press Roman	Mixed pitch			12
187	Press Roman Bold	Mixed pitch	Bold		12
188	Press Roman Italic	Mixed pitch	Italic		12
189	Press Roman Italic Bold	Mixed pitch	Bold and italic		12
190	Foundry	Mixed pitch			12
191	Foundry Bold	Mixed pitch	Bold		12
194	Foundry Italic	Mixed pitch	Italic		12
195	Foundry Italic Bold	Mixed pitch	Bold and italic		12
203	Data 1	Uniformly spaced			13
204	Matrix Gothic	Uniformly spaced			13
205	Matrix Gothic	Uniformly spaced			13
211	Shalom	Uniformly spaced			15
212	Shalom Bold	Uniformly spaced	Bold		15
221	Prestige	Uniformly spaced			15
222	Gothic	Uniformly spaced			15
223	Courier	Uniformly spaced			15
225	Symbol	Uniformly spaced			15
226	Shalom	Uniformly spaced			15
229	Serif	Uniformly spaced			15
230	Gothic	Uniformly spaced			15
232	Matrix Gothic	Uniformly spaced			15
233	Matrix Courier	Uniformly spaced			15
234	Shalom Bold	Uniformly spaced	Bold		15
244	Courier Double Wide	Uniformly spaced	Double wide		5

Font global identifier (FGID)	Name	Type of Font	Attributes	Point	Pitch (CPI)
245	Courier Bold Double Wide	Uniformly spaced	Double wide and bold		5
247	Shalom Bold	Uniformly spaced	Bold		17
248	Shalom	Uniformly spaced			17
249	Katakana	Uniformly spaced			17
252	Courier	Uniformly spaced			17
253	Courier Bold	Uniformly spaced	Bold		17
254	Courier	Uniformly spaced			17
255	Matrix Gothic	Uniformly spaced			17
256	Prestige	Uniformly spaced			17
258	Matrix Gothic	Uniformly spaced			18
259	Matrix Gothic	Uniformly spaced			18
279	Nasseem	Uniformly spaced			17
281	Gothic Text	Uniformly spaced			20
282	Aviv	Uniformly spaced			20
283	Letter Gothic	Uniformly spaced			20
285	Letter Gothic	Uniformly spaced			25
290	Gothic Text	Uniformly spaced			27
300	Gothic	Uniformly spaced		Scalable	17, scalable
304	Gothic Text	Uniformly spaced		Scalable	Scalable
305	OCR-A	Uniformly spaced		Scalable	Scalable
306	OCR-B	Uniformly spaced		Scalable	Scalable
307	APL	Uniformly spaced		Scalable	Scalable
318	Prestige Bold	Uniformly spaced	Bold	Scalable	Scalable



Font global identifier (FGID)	Name	Type of Font	Attributes	Point	Pitch (CPI)
319	Prestige Italic	Uniformly spaced	Italic	Scalable	Scalable
322	APL Bold	Uniformly spaced	Bold	Scalable	Scalable
400	Gothic	Uniformly spaced		Scalable	17, scalable
404	Letter Gothic Bold	Uniformly spaced	Bold	Scalable	Scalable
416	Courier Roman Medium	Uniformly spaced		Scalable	Scalable
420	Courier Roman Bold	Uniformly spaced	Bold	Scalable	Scalable
424	Courier Roman Italic	Uniformly spaced	Italic	Scalable	Scalable
428	Courier Roman Italic Bold	Uniformly spaced	Bold and italic	Scalable	Scalable
432	Prestige	Uniformly spaced		Scalable	Scalable
434	Orator Bold	Uniformly spaced	Bold		8
435	Orator Bold	Uniformly spaced	Bold		6
751	Sonoran Serif	Typographic		8P	27
752	Nasseem	Typographic		12P	18
753	Nasseem Bold	Typographic	Bold	12P	18
754	Nasseem Bold	Typographic	Bold	18P	12
755	Nasseem Bold	Typographic	Bold	24P	9
756	Nasseem Italic	Typographic	Italic	12P	18
757	Nasseem Bold Italic	Typographic	Bold and italic	12P	18
758	Nasseem Bold Italic	Typographic	Bold and italic	18P	12
759	Nasseem Bold Italic	Typographic	Bold and italic	24P	9
760	Times Roman	Typographic		6P	36
761	Times Roman Bold	Typographic	Bold	12P	18
762	Times Roman Bold	Typographic	Bold	10P	15
763	Times Roman Italic	Typographic	Italic	12P	18
764	Times Roman Bold Italic	Typographic	Bold and italic	10P	21
765	Times Roman Bold Italic	Typographic	Bold and italic	12P	18
1051	Sonoran Serif	Typographic		10P	21
1053	Sonoran Serif Bold	Typographic	Bold	10P	21
1056	Sonoran Serif Italic	Typographic	Italic	10P	21
1351	Sonoran Serif	Typographic		12P	18
1653	Sonoran Serif Bold	Typographic	Bold		13
1803	Sonoran Serif Bold	Typographic	Bold	18P	12

Font global identifier (FGID)	Name	Type of Font	Attributes	Point	Pitch (CPI)
2103	Sonoran Serif Bold	Typographic	Bold	24P	9
2304	Helvetica Roman Medium	Typographic		Scalable	Scalable
2305	Helvetica Roman Bold	Typographic	Bold	Scalable	Scalable
2306	Helvetica Roman Italic	Typographic	Italic	Scalable	Scalable
2307	Helvetica Roman Italic Bold	Typographic	Bold and italic	Scalable	Scalable
2308	Times New Roman Medium	Typographic		Scalable	Scalable
2309	Times New Roman Bold	Typographic	Bold	Scalable	Scalable
2310	Times New Roman Italic	Typographic	Italic	Scalable	Scalable
2311	Times New Roman Italic Bold	Typographic	Bold and italic	Scalable	Scalable
4407	Sonoran Serif	Typographic		8P	*27
4407	Sonoran Serif	Typographic		10P	*21
4407	Sonoran Serif	Typographic		12P	*18
4427	Sonoran Serif Bold	Typographic	Bold	10P	*21
4427	Sonoran Serif Bold	Typographic	Bold	16P	*13
4427	Sonoran Serif Bold	Typographic	Bold	24P	*9
4535	Sonoran Serif Italic	Typographic	Italic	10P	*21
4919	Goudy	Typographic		6P	*36
4919	Goudy	Typographic		8P	*27
4919	Goudy	Typographic		10P	*21
4919	Goudy	Typographic		12P	*18
4939	Goudy Bold	Typographic	Bold	10P	*21
4939	Goudy Bold	Typographic	Bold	14P	*15
4939	Goudy Bold	Typographic	Bold	18P	*12
5047	Goudy Italic	Typographic	Italic	10P	*21
5067	Goudy Bold Italic	Typographic	Bold and italic	10P	*21
5687	Times Roman	Typographic		6P	*36
5687	Times Roman	Typographic		8P	*27
5687	Times Roman	Typographic		10P	*21
5687	Times Roman	Typographic		12P	*18
5707	Times Roman Bold	Typographic	Bold	10P	*21
5707	Times Roman Bold	Typographic	Bold	12P	*18
5707	Times Roman Bold	Typographic	Bold	14P	*15
5707	Times Roman Bold	Typographic	Bold	18P	*12
5707	Times Roman Bold	Typographic	Bold	24P	*12
5815	Times Roman Italic	Typographic	Italic	10P	*21
5815	Times Roman Italic	Typographic	Italic	12P	*18
5835	Times Roman Italic Bold	Typographic	Bold and italic	10P	*21
5835	Times Roman Italic Bold	Typographic	Bold and italic	12P	*18
5943	University	Typographic		12P	*18
5943	University	Typographic		14P	*15

Font global identifier (FGID)	Name	Type of Font	Attributes	Point	Pitch (CPI)
5943	University	Typographic		18P	*12
6199	Palatino	Typographic		6P	*36
6199	Palatino	Typographic		8P	*27
6199	Palatino	Typographic		10P	*21
6199	Palatino	Typographic		12P	*18
6219	Palatino Bold	Typographic	Bold	10P	*21
6219	Palatino Bold	Typographic	Bold	14P	*15
6219	Palatino Bold	Typographic	Bold	18P	*12
6327	Palatino Italic	Typographic	Italic	10P	*21
6347	Palatino Italic Bold	Typographic	Bold and italic	10P	*21
8503	Baskerville	Typographic		6P	*36
8503	Baskerville	Typographic		8P	*27
8503	Baskerville	Typographic		10P	*21
8503	Baskerville	Typographic		12P	*18
8523	Baskerville Bold	Typographic	Bold	10P	*21
8523	Baskerville Bold	Typographic	Bold	14P	*15
8523	Baskerville Bold	Typographic	Bold	18P	*12
8631	Baskerville Italic	Typographic	Italic	10P	*21
8651	Baskerville Italic Bold	Typographic	Bold and italic	10P	*21
8759	Nasseem	Typographic		12P	*18
8779	Nasseem Bold	Typographic	Bold	12P	*18
8779	Nasseem Bold	Typographic	Bold	18P	*12
8779	Nasseem Bold	Typographic	Bold	24P	*9
8887	Nasseem Italic	Typographic	Italic	12P	*18
8907	Nasseem Italic Bold	Typographic	Bold and italic	12P	*18
8907	Nasseem Italic Bold	Typographic	Bold and italic	18P	*12
8907	Nasseem Italic Bold	Typographic	Bold and italic	24P	*9
12855	Narkisim	Typographic		8P	*27
12855	Narkisim	Typographic		10P	*21
12855	Narkisim	Typographic		18P	*12
12855	Narkisim	Typographic		24P	*9
12875	Narkisim Bold	Typographic	Bold	8P	*27
12875	Narkisim Bold	Typographic	Bold	10P	*21
12875	Narkisim Bold	Typographic	Bold	12P	*18
16951	Century Schoolbook	Typographic		6P	*36
16951	Century Schoolbook	Typographic		8P	*27
16951	Century Schoolbook	Typographic		10P	*21
16951	Century Schoolbook	Typographic		12P	*18
16971	Century Schoolbook Bold	Typographic	Bold	10P	*21
16971	Century Schoolbook Bold	Typographic	Bold	14P	*15

Font global identifier (FGID)	Name	Type of Font	Attributes	Point	Pitch (CPI)
16971	Century Schoolbook Bold	Typographic	Bold	18P	*12
17079	Century Schoolbook Italic	Typographic	Italic	10P	*21
17099	Century Schoolbook Italic Bold	Typographic	Bold and italic	10P	*21
20224	Boldface	Typographic	Bold	Scalable	Scalable
33335	Optima	Typographic		6P	*36
33335	Optima	Typographic		8P	*27
33335	Optima	Typographic		10P	*21
33335	Optima	Typographic		12P	*18
33355	Optima Bold	Typographic	Bold	10P	*21
33355	Optima Bold	Typographic	Bold	14P	*15
33355	Optima Bold	Typographic	Bold	18P	*12
33463	Optima Italic	Typographic	Italic	10P	*21
33483	Optima Italic Bold	Typographic	Bold and italic	10P	*21
33591	Futura	Typographic		6P	*36
33591	Futura	Typographic		8P	*27
33591	Futura	Typographic		10P	*21
33591	Futura	Typographic		12P	*18
33601	Futura Bold	Typographic	Bold	10P	*21
33601	Futura Bold	Typographic	Bold	14P	*15
33601	Futura Bold	Typographic	Bold	18P	*12
33719	Futura Italic	Typographic	Italic	10P	*21
33729	Futura Italic Bold	Typographic	Bold and italic	10P	*21
34103	Helvetica	Typographic		6P	*36
34103	Helvetica	Typographic		8P	*27
34103	Helvetica	Typographic		10P	*21
34103	Helvetica	Typographic		12P	*18
34123	Helvetica Bold	Typographic	Bold	10P	*21
34123	Helvetica Bold	Typographic	Bold	14P	*15
34123	Helvetica Bold	Typographic	Bold	18P	*12
34231	Helvetica Italic	Typographic	Italic	10P	*21
34251	Helvetica Italic Bold	Typographic	Bold and italic	10P	*21
37431	Old English	Typographic		12P	*18
37431	Old English	Typographic		14P	*15
37431	Old English	Typographic		18P	*12
41783	Coronet Cursive	Typographic		12P	*18
41803	Coronet Cursive Bold	Typographic	Bold	14P	*15
41803	Coronet Cursive Bold	Typographic	Bold	18P	*12

**Note:** Pitch or CPI column for typographic fonts indicates the width of the space character between printed characters. Width, pitch, and CPI of other space characters will vary.

**Related reference:**

“Font attributes and font types”

Font attributes are the characteristics or properties that combine to give a font identity. For example, attributes can be 14 point (height of the font), bold, and italic.

**Font attributes and font types**

Font attributes are the characteristics or properties that combine to give a font identity. For example, attributes can be 14 point (height of the font), bold, and italic.

The following are types of fonts:

**Mixed pitch**

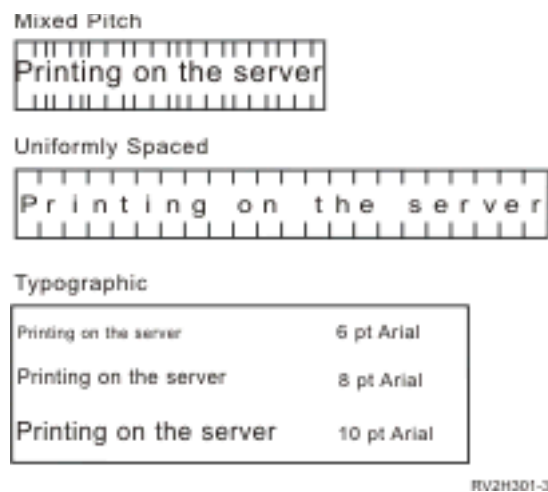
Simulates proportionally spaced fonts. Characters in the font have a limited number of widths. Overall spacing is about 12 characters per inch. Examples are Document or Essay fonts.

**Uniformly spaced**

Similar to typewriter fonts. Characters in the font are all the same width. Examples are Courier and Gothic Text fonts. Some uniformly spaced fonts and many typographic fonts are scalable. For scalable fonts, specify a point size to indicate the size of the font. For example a 12 point uniformly spaced font corresponds to 10 CPI. An example of such a font is font 416, Courier Roman Medium. If no point size is specified when using scalable fonts, 10 point is defaulted.

**Typographic**

Typographic fonts have variable height, measured in points (1 point = 1/72 inch). Therefore, a 36-point font has characters that are 1/2 inch high. Typographic fonts have variable widths. Width is part of the design and varies on a character-by-character basis. Examples are Sonoran Serif and Century Schoolbook.



**Printer font support**

This table contains font ID substitution information which identifies which fonts are supported by certain printers. For example, if your application specified a certain font ID that your printer did not support, you can find out which printers support that font and route your printed output to a printer that supports that font ID.

This table also provides information about what fonts are substituted if the font ID is not supported by the printer the document is routed to.

In the following table:

- A blank indicates that the font is supported by the printer.
- A number indicates the font ID that is substituted

- An asterisk indicates that the substituted font can have a different pitch. When the substituted font is scalable, the pitch is the same.

Font global identifier (FGID)	Printers							
	4224 4230 IPDS	4234 IPDS	3812 or 3816 SCS	3812 or 3816 IPDS	5219	3112 3116 3912 3916 4028 4312 4317 4324 Infoprint 20 Infoprint 32 Infoprint 2060 ES Infoprint 2075 ES Infoprint 2085 Infoprint 2090 ES Infoprint 2105 Infoprint 2105 ES	4019 <sup>1</sup>	3130 3160 3935 Infoprint 3000 Infoprint 4000
2	11	11	11	11	11	11 <sup>2</sup>		11
3					11			
5	11	26				11 <sup>2</sup>		11
8	11	11	11	11	11	11		11
10	11	11	11	11	11	11		*416
11								
12	11	26						
13	11	11				11	11	11
18	11	26			11			
19					11			
20	11	26				11	11	*432
21	11	11	11	11	11	11		*304
25	11	11	11	11	11	11 <sup>2</sup>		11
26						11	11	11
30	11	11			11	11	11	11
31	26	26	26		26	11	11	*416
36	11	11	11	11	11	11 <sup>2</sup>		*400
38	11	26			11	46	46	46
39	26	26			11	46	46	46
40	26	26			11	11	11	*304
41	11	26			11	11	11	11
42	11	26			11	11	11	11
43	11	26			11	18	11	18
44	11	11			11	11	11	*304
46	11	26			11			
49	26	26	26		26	11		*416
50	26	26			26	46		*420

Font global identifier (FGID)	Printers							
	4224 4230 IPDS	4234 IPDS	3812 or 3816 SCS	3812 or 3816 IPDS	5219	3112 3116 3912 3916 4028 4312 4317 4324 Infoprint 20 Infoprint 32 Infoprint 2060 ES Infoprint 2075 ES Infoprint 2085 Infoprint 2090 ES Infoprint 2105 Infoprint 2105 ES	4019 <sup>1</sup>	3130 3160 3935 Infoprint 3000 Infoprint 4000
51	26				26	11	11	11
52	11				11	11	11	11
55	26	26	26		26	46	46	*420
61	11	11	11	11	11	11		*416
62	11	11	11	11	11	18		*424
63	11	11	11	11	11	46		*420
64	11	11	11	11	11	46		*428
66	87	87			87	85	85	*304
68	87	87			87	92	85	92
69	87	87			87	111	85	111
70	87	87			87	85	85	85
71	87	87			87	92	85	92
72	87	87			87	111	85	111
74	87		87	87	87	85	85	85
75	85		85	85	85	85	85	85
76	85	85	85	85	85			
78	85	85	85	85	85	85		*304
80	87	87				85		85
84	87	87				85 <sup>2</sup>		85
85								
86	87	87						
87						85 <sup>2</sup>		*400
91	87	87				92 <sup>2</sup>		92
92	85	85	85	85	85			
95	85	85	85	85	85	85 <sup>2</sup>		85
96	85	85	85	85	85	85 <sup>2</sup>		85
98	87	87	87		87	85		*416
99	87	87	87		87	85	85	*416
101	87	87	87		87	111	85	*416

Font global identifier (FGID)	Printers							
	4224 4230 IPDS	4234 IPDS	3812 or 3816 SCS	3812 or 3816 IPDS	5219	3112 3116 3912 3916 4028 4312 4317 4324 Infoprint 20 Infoprint 32 Infoprint 2060 ES Infoprint 2075 ES Infoprint 2085 Infoprint 2090 ES Infoprint 2105 Infoprint 2105 ES	4019 <sup>1</sup>	3130 3160 3935 Infoprint 3000 Infoprint 4000
102	87	87	87		87	111	85	*420
103	85	85	85	85	85	85		*416
109	85	85	85	85	85	92 <sup>2</sup>		92
110	87	87			87	11 <sup>2</sup>		*404
111	87	87			86			
112	87	87			86			
154	85		160	160	160	164	159	159
155	160	160			160	159 <sup>2</sup>		159
157	160	160	160	160	160	164 <sup>2</sup>		159
158	160	160				164 <sup>2</sup>		159
159	160	160						
160						164 <sup>2</sup>		159
162	160	160				164 <sup>2</sup>		159
163	160	160			160	159	159	159
164	160	160	160	160	160			159
167	160	160	160		160	164		*416
168	160	160	160		160	159	159	*420
173	160	160			160	164	159	159
174	160	160	160	160	160	164	159	159
175	160	160			160	164	159	159
178	*400	*258	*281		*222	*281	*254	*416
179	*400	*258	*281		*222	*281	*254	*420
180	*222	*222	*230		*222	*223	*254	*416
181	*222	*222	*230		*222	*223	*254	*420
182	*11	*11	*244		*11	*11	*11	*416
183	*11	*11	*244		*11	*46	*46	*420
186	160	160	160	160	160	164 <sup>2</sup>		159
187	160	160	160	160	160	159 <sup>2</sup>		159
188	160	160	160	160	160	164 <sup>2</sup>		159



Font global identifier (FGID)	Printers							
	4224 4230 IPDS	4234 IPDS	3812 or 3816 SCS	3812 or 3816 IPDS	5219	3112 3116 3912 3916 4028 4312 4317 4324 Infoprint 20 Infoprint 32 Infoprint 2060 ES Infoprint 2075 ES Infoprint 2085 Infoprint 2090 ES Infoprint 2105 Infoprint 2105 ES	4019 <sup>1</sup>	3130 3160 3935 Infoprint 3000 Infoprint 4000
189	160	160	160	160	160	159 <sup>2</sup>		159
190	160	160	160	160	160	164 <sup>2</sup>		159
191	160	160	160	160	160	159 <sup>2</sup>		159
194	160	160	160	160	160	164 <sup>2</sup>		159
195	160	160	160	160	160	159 <sup>2</sup>		159
203	*222	204	204	204	*222		*254	*416
204	*222				*222	*223	*254	*304
205	*222		204	204	*222	*223	*254	*416
211	222	222	230		222	223	*254	*416
212	222	222	230		222	223	*254	*420
221	222	222						
222			230	230		223 <sup>2</sup>		*304
223								
225	222	222				223	*254	223
226	222	222	230		222	223		*416
229	222	222			222	223	*254	223
230	222	222			222	223	*254	*304
232	222		230	230	222	223	*254	*223
233	223		230	230	223	223	*254	223
234	222	222	230		222	223	*254	*420
244	*11	*26			*11	*11		*416
245	*11	*26			*11	*46		*420
247	*400	*258	252		*222	254	254	*420
248	*400	*258	252		*222	254	254	*416
249	*400	*258	252	252	*222	254		*304
252	*400	*258			*222	254	254	254
253	*400	*258			*222	254	254	*420
254	*400	*258			*222			
255	*400	*258	252	252	*222	254		254

Font global identifier (FGID)	Printers							
	4224 4230 IPDS	4234 IPDS	3812 or 3816 SCS	3812 or 3816 IPDS	5219	3112 3116 3912 3916 4028 4312 4317 4324 Infoprint 20 Infoprint 32 Infoprint 2060 ES Infoprint 2075 ES Infoprint 2085 Infoprint 2090 ES Infoprint 2105 Infoprint 2105 ES	4019 <sup>1</sup>	3130 3160 3935 Infoprint 3000 Infoprint 4000
256	*400	*258	252	252	*222			
258	*400		*281	*281	*222	*281	*254	*416
259	*400		*281	*281	*222	*281	*254	*416
279	*400	*258	252	252	*222	254		*416
281	*400	*258			*222			
282	*400	*258	281		*222	281		*416
283	*400	*258	281	281.	*222	281		*400
285	*400	*258	*290	*290	*222	281 <sup>2</sup>		*400
290	*400	*258			*222		*254	*416
300	400		*252	*252	*222	*254	*254	*416
304	*26	*26	*26	*26	*26	*11	*11	
305	*19	*19	*19	*19	*11	*19	*19	
306	*3	*3	*3	*3	*11	*3	*3	
307	*85	*85	*85	*85	*85	*76	*76	
318	*11	*26	*12	*12	*12	*12	*12	
319	*11	*26	*12	*12	*12	*12	*12	
322	*85	*85	*85	*85	*85	*76	*76	
400			*252	*252	*222	*254	*254	
404	*26	*26	*39	*39	*11	*46	*46	
416	*11	*11	*11	*11	*11	*11	*11	
420	*11	*26	*46	*46	*11	*46	*46	
424	*11	*26	*18	*18	*11	*18	*18	
428	*11	*26	*18	*18	*11	*18	*18	
432	*11	*26	*12	*12	*12	*12	*12	
434	*11	*11	*11	*11	*11	46 <sup>2</sup>		*420
435	*11	*11	*11	*11	*11	46 <sup>2</sup>		*420
751	*400	*258			*222.		*254	
752	*400	*258	*281	*281	*222		*254	*2308
753	*400	*258	*281	*281	*222		*254	*2309

Font global identifier (FGID)	Printers							
	4224 4230 IPDS	4234 IPDS	3812 or 3816 SCS	3812 or 3816 IPDS	5219	3112 3116 3912 3916 4028 4312 4317 4324 Infoprint 20 Infoprint 32 Infoprint 2060 ES Infoprint 2075 ES Infoprint 2085 Infoprint 2090 ES Infoprint 2105 Infoprint 2105 ES	4019 <sup>1</sup>	3130 3160 3935 Infoprint 3000 Infoprint 4000
754	*85	*85	*85	*85	*85		*85	*2309
755	*11	*11	*11	*11	*11		*46	*2309
756	*400	*258	*281	*281	*222		*254	*2310
757	*400	*258	*281	*281	*222		*254	*2311
758	*85	*85	*85	*85	*85		*85	*2311
759	*11	*11	*11	*11	*11		*46	*2311
760	*400	*258	*290	*290	*222		*254	
761	*400	*258	*281	*281	*222		*254	
762	*222	*222	*230	*230	*222		*254	
763	*400	*258	*281	*281	*222		*254	
764	*400	*258	*290	*290	*222		*254	
765	*400	*258	*281	*281	*222		*254	
1051	*400	*258			*222		*254	
1053	*400	*258			*222		*254	
1056	*400	*258			*222		*254	
1351	*400	*258			*222		*254	
1653	*222	*222			*222		*254	*2309
1803	*85	*85	*85	*85	*85		*85	
2103	*11	*11			*11		*46	
2304	*400	*258	*290	*290	*222	*760	*254	
2305	*400	*258	*281	*281	*222	*761	*254	
2306	*400	*258	*281	*281	*222	*763	*254	
2307	*400	*258	*280	*290	*222	*764	*254	
2308	*400	*258	*290	*290	*222	*760	*254	
2309	*400	*258	*281	*281	*222	*761	*254	
2310	*400	*258	*281	*281	*222	*763	*254	
2311	*400	*258	*290	*290	*222	*764	*254	
4407 (8P)	*400	*258			*222	5687 <sup>2</sup>	*254	*2308

Font global identifier (FGID)	Printers							
	4224 IPDS	4234 IPDS	3812 or 3816 SCS	3812 or 3816 IPDS	5219	3112 3116 3912 3916 4028 4312 4317 4324 Infoprint 20 Infoprint 32 Infoprint 2060 ES Infoprint 2075 ES Infoprint 2085 Infoprint 2090 ES Infoprint 2105 Infoprint 2105 ES	4019 <sup>1</sup>	3130 3160 3935 Infoprint 3000 Infoprint 4000
4407 (10P)	*400	*258			*222	5687 <sup>2</sup>	*254	*2308
4407 (12P)	*400	*258			*222	5687 <sup>2</sup>	*254	*2308
4427 (10P)	*400	258			*222	5687 <sup>2</sup>	*254	*2309
4427 (16P)	*222	*222			*11	*5707	*254	*2309
4427 (24P)	*11	*11			*11	5707 <sup>2</sup>	*254	*2309
4535 (10P)	*400	*258			*222	5687 <sup>2</sup>	*46	*2310
4919 (6P)	*400	*258	*290	*290	*222	5687 <sup>2</sup>		*2308
4919 (8P)	*400	*258	*751	*751	*222	5687 <sup>2</sup>		*2308
4919 (10P)	*400	*258	*1051	*1051	*222	5687 <sup>2</sup>		*2308
4919 (12P)	*400	*258	*1351	*1351	*222	5687 <sup>2</sup>		*2308
4939 (10P)	*400	*258	*1053	*1053	*222	5707 <sup>2</sup>		*2309
4939 (14P)	*222	*222	*1351	*1351	*222	5707 <sup>2</sup>		*2309
4939 (18P)	*85	*85	*1653	*1653	*85	5707 <sup>2</sup>		*2309
5047 (10P)	*400	*258	*1056	*1056	*222	5687 <sup>2</sup>		*2310
5067 (10P)	*400	*258	*1053	*1053	*222	5687 <sup>2</sup>		*2311
5687 (6P)	*400	*258	*290	*290	*222			
5687 (8P)	*400	*258	*751	*751	*222			
5687 (10P)	*400	*258	*1051	*1051	*222			

Font global identifier (FGID)	Printers							
	4224 4230 IPDS	4234 IPDS	3812 or 3816 SCS	3812 or 3816 IPDS	5219	3112 3116 3912 3916 4028 4312 4317 4324 Infoprint 20 Infoprint 32 Infoprint 2060 ES Infoprint 2075 ES Infoprint 2085 Infoprint 2090 ES Infoprint 2105 Infoprint 2105 ES	4019 <sup>1</sup>	3130 3160 3935 Infoprint 3000 Infoprint 4000
5687 (12P)	*400	*258	*1351	*1351	*222			
5707 (10P)	*400	*258	*1053	*1053	*222			
5707 (12P)	*400	*258	*1351	*1351	*222		*254	
5707 (14P)	*222	*222	*1351	*1351	*222			
5707 (18P)	*85	*85	*1653	*1653	*85			
5707 (24P)	*11	*11	*2103	*2103	*11			
5815 (10P)	*400	*258	*1056	*1056	*222			
5815 (12P)	*400	*258	*1351	*1351	*222		*254	
5835 (10P)	*400	*258	*1053	*1053	*222			
5835 (12P)	*400	*258	*1351	*1351	*222			
5943 (12P)	*400	*258	*1351	*1351	*222	5687 <sup>2</sup>		*2308
5943 (14P)	*222	*222	*1351	*1351	*222	5707 <sup>2</sup>		*2308
5943 (18P)	*85	*85	*1653	*1653	*85	5707 <sup>2</sup>		*2308
6199 (6P)	*400	*258	*290	*290	*222	5687 <sup>2</sup>		*2308
6199 (8P)	*400	*258	*751	*751	*222	5687 <sup>2</sup>		*2308
6199 (10P)	*400	*258	*1051	*1051	*222	5687 <sup>2</sup>		*2308
6199 (12P)	*400	*258	*1351	*1351	*222	5687 <sup>2</sup>		*2308
6219 (10P)	*400	*258	*1053	*1053	*222	5687 <sup>2</sup>		*2309

Font global identifier (FGID)	Printers							
	4224 4230 IPDS	4234 IPDS	3812 or 3816 SCS	3812 or 3816 IPDS	5219	3112 3116 3912 3916 4028 4312 4317 4324 Infoprint 20 Infoprint 32 Infoprint 2060 ES Infoprint 2075 ES Infoprint 2085 Infoprint 2090 ES Infoprint 2105 Infoprint 2105 ES	4019 <sup>1</sup>	3130 3160 3935 Infoprint 3000 Infoprint 4000
6219 (14P)	*222	*222	*1351	*1351	*222	5707 <sup>2</sup>		*2309
6219 (18P)	*85	*85	*1653	*1653	*85	5707 <sup>2</sup>		*2309
6327 (10P)	*400	*258	*1056	*1056	*222	5687 <sup>2</sup>		*2310
6347 (10P)	*400	*258	*1053	*1053	*222	5686 <sup>2</sup>		*2311
8503 (6P)	*400	*258	*290	*290	*222	5687 <sup>2</sup>		*2308
8503 (8P)	*400	*258	*751	*751	*222	5687 <sup>2</sup>		*2308
8503 (10P)	*400	*258	*1051	*1051	*222	5687 <sup>2</sup>		*2308
8503 (12P)	*400	*258	*1351	*1351	*222	5687 <sup>2</sup>		*2308
8523 (10P)	*400	*258	*1053	*1053	*222	5687 <sup>2</sup>		*2309
8523 (14P)	*222	*222	*1351	*1351	*222	5707 <sup>2</sup>		*2309
8523 (18P)	*85	*85	*1653	*1653	*85	5707 <sup>2</sup>		*2309
8631 (10P)	*400	*258	*1056	*1056	*222	5687 <sup>2</sup>		*2310
8651 (10P)	*400	*258	*1053	*1053	*222	5687 <sup>2</sup>		*2311
8759 (12P)	*400	*258	*1351	*1351	*222	5687 <sup>2</sup>		*2308
8779 (12P)	*400	*258	*1351	*1351	*222	5707 <sup>2</sup>		*2309
8779 (18P)	*85	*85	*1653	*1653	*85	5707 <sup>2</sup>		*2309
8779 (24P)	*11	*11	*2103	*2103	*11	5707 <sup>2</sup>		*2309
8887 (12P)	*400	*258	*1351	*1351	*222	5687 <sup>2</sup>		*2310

Font global identifier (FGID)	Printers							
	4224 4230 IPDS	4234 IPDS	3812 or 3816 SCS	3812 or 3816 IPDS	5219	3112 3116 3912 3916 4028 4312 4317 4324 Infoprint 20 Infoprint 32 Infoprint 2060 ES Infoprint 2075 ES Infoprint 2085 Infoprint 2090 ES Infoprint 2105 Infoprint 2105 ES	4019 <sup>1</sup>	3130 3160 3935 Infoprint 3000 Infoprint 4000
8907 (12P)	*400	*258	*1351	*1351	*222	5687 <sup>2</sup>		*2311
8907 (18P)	*85	*85	*1653	*1653	*85	5707 <sup>2</sup>		*2311
8907 (24P)	*11	*11	*2103	*2103	*11	5707 <sup>2</sup>		*2311
12855 (8P)	*400	*258	*751		*222	5687 <sup>2</sup>		*2308
12855 (10P)	*400	*258	*1051	*1051	*222	5687 <sup>2</sup>		*2308
12855 (18P)	*85	*85	*1653	*1653	*85	5707 <sup>2</sup>		*2308
12855 (24P)	*11	*11	*2103	*2103	*11	5707 <sup>2</sup>		*2308
12875 (8P)	*400	*258	*751		*222	5687 <sup>2</sup>		*2309
12875 (10P)	*400	*258	*1053	*1053	*222	5687 <sup>2</sup>		*2309
12875 (12P)	*400	*258	*1351	*1351	*222	5687 <sup>2</sup>		*2309
16951 (6P)	*400	*258	*290	*290	*222	5687 <sup>2</sup>		*2308
16951 (8P)	*400	*258	*751	*751	*222	5687 <sup>2</sup>		*2308
16951 (10P)	*400	*258	*1051	*1051	*222	5687 <sup>2</sup>		*2308
16951 (12P)	*400	*258	*1351	*1351	*222	5687 <sup>2</sup>		*2308
16971 (10P)	*400	*258	*1053	*1053	*222	5687 <sup>2</sup>		*2309
16971 (14P)	*222	*222	*1351	*1351	*222	5707 <sup>2</sup>		*2309
16971 (18P)	*85	*85	*1653	*1653	*85	5707 <sup>2</sup>		*2309

Font global identifier (FGID)	Printers							
	4224 4230 IPDS	4234 IPDS	3812 or 3816 SCS	3812 or 3816 IPDS	5219	3112 3116 3912 3916 4028 4312 4317 4324 Infoprint 20 Infoprint 32 Infoprint 2060 ES Infoprint 2075 ES Infoprint 2085 Infoprint 2090 ES Infoprint 2105 Infoprint 2105 ES	4019 <sup>1</sup>	3130 3160 3935 Infoprint 3000 Infoprint 4000
17079 (10P)	*400	*258	*1056	*1056	*222	5687 <sup>2</sup>		*2311
17099 (10P)	*400	*258	*1053	*1053	*222	5687 <sup>2</sup>		*2311
20224	*160	*160	*159	*159	*159	*159	*159	
33335 (6P)	*400	*258	*290	*290	*222	5687 <sup>2</sup>		*2308
33335 (8P)	*400	*258	*751	*751	*222	5687 <sup>2</sup>		*2308
33335 (10P)	*400	*258	*1051	*1051	*222	5687 <sup>2</sup>		*2308
33335 (12P)	*400	*258	*1351	*1351	*222	5687 <sup>2</sup>		*2308
33355 (10P)	*400	*258	*1053	*1053	*222	5687 <sup>2</sup>		*2309
33355 (14P)	*222	*222	*1351	*1351	*222	5707 <sup>2</sup>		*2309
33355 (18P)	*85	*85	*1653	*1653	*85	5707 <sup>2</sup>		*2309
33463 (10P)	*400	*258	*1056	*1056	*222	5687 <sup>2</sup>		*2310
33483 (10P)	*400	*258	*1053	*1053	*222	5687 <sup>2</sup>		*2311
33591 (6P)	*400	*258	*290	*290	*222	5687 <sup>2</sup>		*2308
33591 (8P)	*400	*258	*751	*751	*222	5687 <sup>2</sup>		*2308
33591 (10P)	*400	*258	*1051	*1051	*222	5687 <sup>2</sup>		*2308
33591 (12P)	*400	*258	*1351	*1351	*222	5687 <sup>2</sup>		*2308
33601 (10P)	*400	*258	*1053	*1053	*222	5687 <sup>2</sup>		*2309



Font global identifier (FGID)	Printers							
	4224 4230 IPDS	4234 IPDS	3812 or 3816 SCS	3812 or 3816 IPDS	5219	3112 3116 3912 3916 4028 4312 4317 4324 Infoprint 20 Infoprint 32 Infoprint 2060 ES Infoprint 2075 ES Infoprint 2085 Infoprint 2090 ES Infoprint 2105 Infoprint 2105 ES	4019 <sup>1</sup>	3130 3160 3935 Infoprint 3000 Infoprint 4000
33601 (14P)	*222	*222	*1351	*1351	*222	5707 <sup>2</sup>		*2309
33601 (18P)	*85	*85	*1653	*1653	*85	5707 <sup>2</sup>		*2309
33719 (10P)	*400	*258	*1056	*1056	*222	5687 <sup>2</sup>		*2310
33729 (10P)	*400	*258	*1053	*1053	*222	5687 <sup>2</sup>		*2311
34103 (6P)	*400	*258	*290	*290	*222	5687 <sup>2</sup>		*2304
34103 (8P)	*400	*258	*751	*751	*222	5687 <sup>2</sup>		*2304
34103 (10P)	*400	*258	*1051	*1051	*222	5687 <sup>2</sup>		*2304
34103 (12P)	*400	*258	*1351	*1351	*222	5687 <sup>2</sup>		*2304
34123 (10P)	*400	*258	*1053	*1053	*222	5687 <sup>2</sup>		*2305
34123 (14P)	*222	*222	*1351	*1351	*222	5707 <sup>2</sup>		*2305
34123 (18P)	*85	*85	*1653	*1653	*85	5707 <sup>2</sup>		*2305
34231 (10P)	*400	*258	*1056	*1056	*222	5687 <sup>2</sup>		*2306
34251 (10P)	*400	*258	*1053	*1053	*222	5687 <sup>2</sup>		*2307
37431 (12)	*400	*258	*1351	*1351	*222	5687 <sup>2</sup>		*2308
37431 (14P)	*222	*222	*1351	*1351	*222	5707 <sup>2</sup>		*2308
37431 (18P)	*85	*85	*1653	*1653	*85	5707 <sup>2</sup>		*2308
41783 (12P)	*400	*258	*1351	*1351	*222	5687 <sup>2</sup>		*2308

Printers								
<b>Font global identifier (FGID)</b>	4224 4230 IPDS	4234 IPDS	3812 or 3816 SCS	3812 or 3816 IPDS	5219	3112 3116 3912 3916 4028 4312 4317 4324 Infoprint 20 Infoprint 32 Infoprint 2060 ES Infoprint 2075 ES Infoprint 2085 Infoprint 2090 ES Infoprint 2105 Infoprint 2105 ES	4019 <sup>1</sup>	3130 3160 3935 Infoprint 3000 Infoprint 4000
41803 (14P)	*222	*222	*1351	*1351	*222	5707 <sup>2</sup>		*2309
41803 (18P)	*85	*85	*1653	*1653	*85	5707 <sup>2</sup>		*2309
<b>Notes:</b>								
<sup>1</sup> The 4019 printer has five resident fonts: FGID 11, 46, 85, 159, and 254. The system sends any of those FGIDs that do not show a substitution to the emulator to which the 4019 is attached. The emulator might not support all of the FGIDs and might report an error or perform a substitution of its own.								
<sup>2</sup> The 4028 performs the font substitution as shown unless a font card has been installed that contains that FGID. For example, if a font card with an FGID of 2 is installed, the system sends the FGID of 2 to the printer. However, if the font card is not installed, the system substitutes an FGID of 11.								

### Font substitution by font ID range

If your application specifies a font ID (FGID) that is not found in Printer font support or is not resident in the printer (font card), the system makes a substitution based on the font ID ranges in the following table. For example, if font ID 4 is specified in your application, i5/OS substitutes font ID 11 as shown in the following table.

The following table divides font IDs into ranges. The ranges represent fonts of the same weight and size. A default font is selected in each range for substitution when a font is not found.

FGID	Substituted FGID
Fonts 0 through 65	11
Fonts 66 through 153	85
Fonts 154 through 200	160
Fonts 201 through 210	204
Fonts 211 through 239	223
Fonts 240 through 246	245
Fonts 247 through 257	252
Fonts 258 through 259	259
Fonts 260 through 273	434
Fonts 274 through 279	279
Fonts 280 through 284	281

FGID		Substituted FGID
Fonts 285 through 289		285
Fonts 290 through 299		290
Fonts 300 through 511		252
Fonts 512 through 2303		252
Fonts 2304 through 3839 or fonts 4069 through 65279	Fonts with point size equal to 0 or not specified	252
	Fonts with point size greater than 0 but less than 7.6	5687-6P
	Fonts with point size greater than or equal to 7.6 but less than 9.6	5687-8P
	Fonts with point size greater than or equal to 9.6 but less than 11.6	5687-10P
	Fonts with point size greater than or equal to 11.6 but less than 13.6	5687-12p
	Fonts with point size greater than or equal 13.6 but less than 17.6	5707-14P
	Fonts with point size greater than or equal to 17.6 but less than 23.6	5707-18P
	Fonts with point size greater than or equal to 23.6	5707-24P
Fonts 3840 through 4095 (User-defined)		No Substitution
Fonts 65280 through 65534 (User-defined)		No Substitution

**Related reference:**

“Printer font support” on page 317

This table contains font ID substitution information which identifies which fonts are supported by certain printers. For example, if your application specified a certain font ID that your printer did not support, you can find out which printers support that font and route your printed output to a printer that supports that font ID.

**Mapping host-resident to printer-resident font character sets**

If your application specifies a host resident font character set (fonts are stored on the system) and you want to print the spooled file on an AFP-configured 4224, 4230, 4234, or 64xx printer, the system must substitute a printer-resident font character set (fonts stored on the printer).

The following table can help you determine what printer resident font character set is requested when your spooled file references a host resident font character set instead of a registered font identifier (ID). This font substitution is necessary because these printers do not support the downloading of 240-pel host resident font character sets. Depending on the host resident font character set that is requested, the appropriate registered ID value, font width value, and font attributes are selected to match (as closely as possible) your font request.

The Font width column specifies the width of the blank character in 1440ths of an inch. This is an indicator of how many characters will fit per inch of space on the paper.

The Map fidelity column indicates whether the substituted printer resident font is considered to be an exact match to the font character set that is requested in your spooled file.

*Table 14. Host Resident to Printer Resident Font Character Set Mapping*

<b>Font character set name</b>	<b>Registered font ID</b>	<b>Font width</b>	<b>Font attributes</b>	<b>Map fidelity</b>
C0A053A0	33077	73	Normal	Exact
C0A053B0	33077	80	Normal	Exact
C0A053D0	33077	93	Normal	Exact
C0A053F0	33077	107	Normal	Exact
C0A053H0	33077	120	Normal	Exact
C0A053J0	33077	133	Normal	Exact
C0A053N0	33077	160	Normal	Exact
C0A053T0	33077	200	Normal	Exact
C0A053Z0	33077	240	Normal	Exact
C0A05300	33077	67	Normal	Exact
C0A05360	33077	40	Normal	Exact
C0A05370	33077	47	Normal	Exact
C0A05380	33077	53	Normal	Exact
C0A05390	33077	60	Normal	Exact
C0A055A0	33079	73	Normal	Exact
C0A055B0	33079	80	Normal	Exact
C0A055B1	33079	320	Normal	Exact
C0A055D0	33079	93	Normal	Exact
C0A055F0	33079	107	Normal	Exact
C0A055H0	33079	120	Normal	Exact
C0A055J0	33079	133	Normal	Exact
C0A055N0	33079	160	Normal	Exact
C0A055N1	33079	400	Normal	Exact
C0A055T0	33079	200	Normal	Exact
C0A055Z0	33079	240	Normal	Exact
C0A055Z1	33079	480	Normal	Exact
C0A05500	33079	67	Normal	Exact
C0A05560	33079	40	Normal	Exact
C0A05570	33079	47	Normal	Exact
C0A05580	33079	53	Normal	Exact
C0A05590	33079	60	Normal	Exact
C0A057A0	33081	73	Normal	Exact
C0A057B0	33081	80	Normal	Exact
C0A057D0	33081	93	Normal	Exact
C0A057F0	33081	107	Normal	Exact

Table 14. Host Resident to Printer Resident Font Character Set Mapping (continued)

Font character set name	Registered font ID	Font width	Font attributes	Map fidelity
C0A057H0	33081	120	Normal	Exact
C0A057J0	33081	133	Normal	Exact
C0A057N0	33081	160	Normal	Exact
C0A057T0	33081	200	Normal	Exact
C0A057Z0	33081	240	Normal	Exact
C0A05700	33081	67	Normal	Exact
C0A05760	33081	40	Normal	Exact
C0A05770	33081	47	Normal	Exact
C0A05780	33081	53	Normal	Exact
C0A05790	33081	60	Normal	Exact
C0A073A0	33097	73	Bold	Exact
C0A073B0	33097	80	Bold	Exact
C0A073D0	33097	93	Bold	Exact
C0A073F0	33097	107	Bold	Exact
C0A073H0	33097	120	Bold	Exact
C0A073J0	33097	133	Bold	Exact
C0A073N0	33097	160	Bold	Exact
C0A073T0	33097	200	Bold	Exact
C0A073Z0	33097	240	Bold	Exact
C0A07300	33097	67	Bold	Exact
C0A07360	33097	40	Bold	Exact
C0A07370	33097	47	Bold	Exact
C0A07380	33097	53	Bold	Exact
C0A07390	33097	60	Bold	Exact
C0A075A0	33099	73	Bold	Exact
C0A075B0	33099	80	Bold	Exact
C0A075B1	33099	320	Bold	Exact
C0A075D0	33099	93	Bold	Exact
C0A075F0	33099	107	Bold	Exact
C0A075H0	33099	120	Bold	Exact
C0A075J0	33099	133	Bold	Exact
C0A075N0	33099	160	Bold	Exact
C0A075N1	33099	400	Bold	Exact
C0A075T0	33099	200	Bold	Exact
C0A075Z0	33099	240	Bold	Exact
C0A075Z1	33099	480	Bold	Exact
C0A07500	33099	67	Bold	Exact
C0A07560	33099	40	Bold	Exact
C0A07570	33099	47	Bold	Exact

Table 14. Host Resident to Printer Resident Font Character Set Mapping (continued)

Font character set name	Registered font ID	Font width	Font attributes	Map fidelity
C0A07580	33099	53	Bold	Exact
C0A07590	33099	60	Bold	Exact
C0A077A0	33101	73	Bold	Exact
C0A077B0	33101	80	Bold	Exact
C0A077D0	33101	93	Bold	Exact
C0A077F0	33101	107	Bold	Exact
C0A077H0	33101	120	Bold	Exact
C0A077J0	33101	133	Bold	Exact
C0A077N0	33101	160	Bold	Exact
C0A077T0	33101	200	Bold	Exact
C0A077Z0	33101	240	Bold	Exact
C0A07700	33101	67	Bold	Exact
C0A07760	33101	40	Bold	Exact
C0A07770	33101	47	Bold	Exact
C0A07780	33101	53	Bold	Exact
C0A07790	33101	60	Bold	Exact
C0A153A0	33205	73	Italic	Exact
C0A153B0	33205	80	Italic	Exact
C0A153D0	33205	93	Italic	Exact
C0A153F0	33205	107	Italic	Exact
C0A153H0	33205	120	Italic	Exact
C0A153J0	33205	133	Italic	Exact
C0A153N0	33205	160	Italic	Exact
C0A153T0	33205	200	Italic	Exact
C0A153Z0	33205	240	Italic	Exact
C0A15300	33205	67	Italic	Exact
C0A15360	33205	40	Italic	Exact
C0A15370	33205	47	Italic	Exact
C0A15380	33205	53	Italic	Exact
C0A15390	33205	60	Italic	Exact
C0A155A0	33207	73	Italic	Exact
C0A155B0	33207	80	Italic	Exact
C0A155B1	33207	320	Italic	Exact
C0A155D0	33207	93	Italic	Exact
C0A155F0	33207	107	Italic	Exact
C0A155H0	33207	120	Italic	Exact
C0A155J0	33207	133	Italic	Exact
C0A155N0	33207	160	Italic	Exact
C0A155N1	33207	400	Italic	Exact

Table 14. Host Resident to Printer Resident Font Character Set Mapping (continued)

Font character set name	Registered font ID	Font width	Font attributes	Map fidelity
C0A155T0	33207	200	Italic	Exact
C0A155Z0	33207	240	Italic	Exact
C0A155Z1	33207	480	Italic	Exact
C0A15500	33207	67	Italic	Exact
C0A15560	33207	40	Italic	Exact
C0A15570	33207	47	Italic	Exact
C0A15580	33207	53	Italic	Exact
C0A15590	33207	60	Italic	Exact
C0A175A0	33227	73	Bold italic	Exact
C0A175B0	33227	80	Bold italic	Exact
C0A175B1	33227	320	Bold italic	Exact
C0A175D0	33227	93	Bold italic	Exact
C0A175F0	33227	107	Bold italic	Exact
C0A175H0	33227	120	Bold italic	Exact
C0A175J0	33227	133	Bold italic	Exact
C0A175N0	33227	160	Bold italic	Exact
C0A175N1	33227	400	Bold italic	Exact
C0A175T0	33227	200	Bold italic	Exact
C0A175Z0	33227	240	Bold italic	Exact
C0A175Z1	33227	480	Bold italic	Exact
C0A17500	33227	67	Bold italic	Exact
C0A17560	33227	40	Bold italic	Exact
C0A17570	33227	47	Bold italic	Exact
C0A17580	33227	53	Bold italic	Exact
C0A17590	33227	60	Bold italic	Exact
C0BPOSA0	323	120	Normal	Exact
C0BPOSBN	323	240	Normal	Exact
C0BPOSB0	323	168	Normal	Exact
C0BPOS91	323	144	Normal	Exact
C0B20CA0	335	73	Normal	Exact
C0B20CB0	335	80	Normal	Exact
C0B20CD0	335	93	Normal	Exact
C0B20CF0	335	107	Normal	Exact
C0B20CH0	335	120	Normal	Exact
C0B20CJ0	335	133	Normal	Exact
C0B20CN0	335	160	Normal	Exact
C0B20CT0	335	200	Normal	Exact
C0B20CZ0	335	240	Normal	Exact
C0B20C00	335	67	Normal	Exact

Table 14. Host Resident to Printer Resident Font Character Set Mapping (continued)

Font character set name	Registered font ID	Font width	Font attributes	Map fidelity
COB20C50	335	33	Normal	Exact
COB20C60	335	40	Normal	Exact
COB20C70	335	47	Normal	Exact
COB20C80	335	53	Normal	Exact
COB20C90	335	60	Normal	Exact
COB200A0	335	73	Normal	Exact
COB200B0	335	80	Normal	Exact
COB200D0	335	93	Normal	Exact
COB200F0	335	107	Normal	Exact
COB200H0	335	120	Normal	Exact
COB200J0	335	133	Normal	Exact
COB200N0	335	160	Normal	Exact
COB200T0	335	200	Normal	Exact
COB200Z0	335	240	Normal	Exact
COB20000	335	67	Normal	Exact
COB20050	335	33	Normal	Exact
COB20060	335	40	Normal	Exact
COB20070	335	47	Normal	Exact
COB20080	335	53	Normal	Exact
COB20090	335	60	Normal	Exact
COB30CA0	337	73	Italic	Exact
COB30CB0	337	80	Italic	Exact
COB30CD0	337	93	Italic	Exact
COB30CF0	337	107	Italic	Exact
COB30CH0	337	120	Italic	Exact
COB30CJ0	337	133	Italic	Exact
COB30CN0	337	160	Italic	Exact
COB30CT0	337	200	Italic	Exact
COB30CZ0	337	240	Italic	Exact
COB30C00	337	67	Italic	Exact
COB30C50	337	33	Italic	Exact
COB30C60	337	40	Italic	Exact
COB30C70	337	47	Italic	Exact
COB30C80	337	53	Italic	Exact
COB30C90	337	60	Italic	Exact
COB300A0	337	73	Italic	Exact
COB300B0	337	80	Italic	Exact
COB300D0	337	93	Italic	Exact
COB300F0	337	107	Italic	Exact



Table 14. Host Resident to Printer Resident Font Character Set Mapping (continued)

Font character set name	Registered font ID	Font width	Font attributes	Map fidelity
COB300H0	337	120	Italic	Exact
COB300J0	337	133	Italic	Exact
COB300N0	337	160	Italic	Exact
COB300T0	337	200	Italic	Exact
COB300Z0	337	240	Italic	Exact
COB30000	337	67	Italic	Exact
COB30050	337	33	Italic	Exact
COB30060	337	40	Italic	Exact
COB30070	337	47	Italic	Exact
COB30080	337	53	Italic	Exact
COB30090	337	60	Italic	Exact
COB40CA0	336	73	Bold	Exact
COB40CB0	336	80	Bold	Exact
COB40CD0	336	93	Bold	Exact
COB40CF0	336	107	Bold	Exact
COB40CH0	336	120	Bold	Exact
COB40CJ0	336	133	Bold	Exact
COB40CN0	336	160	Bold	Exact
COB40CT0	336	200	Bold	Exact
COB40CZ0	336	240	Bold	Exact
COB40C00	336	67	Bold	Exact
COB40C50	336	33	Bold	Exact
COB40C60	336	40	Bold	Exact
COB40C70	336	47	Bold	Exact
COB40C80	336	53	Bold	Exact
COB40C90	336	60	Bold	Exact
COB400A0	336	73	Bold	Exact
COB400B0	336	80	Bold	Exact
COB400D0	336	93	Bold	Exact
COB400F0	336	107	Bold	Exact
COB400H0	336	120	Bold	Exact
COB400J0	336	133	Bold	Exact
COB400N0	336	160	Bold	Exact
COB400T0	336	200	Bold	Exact
COB400Z0	336	240	Bold	Exact
COB40000	336	67	Bold	Exact
COB40050	336	33	Bold	Exact
COB40060	336	40	Bold	Exact
COB40070	336	47	Bold	Exact

Table 14. Host Resident to Printer Resident Font Character Set Mapping (continued)

Font character set name	Registered font ID	Font width	Font attributes	Map fidelity
COB40080	336	53	Bold	Exact
COB40090	336	60	Bold	Exact
COB50CA0	338	73	Bold italic	Exact
COB50CB0	338	80	Bold italic	Exact
COB50CD0	338	93	Bold italic	Exact
COB50CF0	338	107	Bold italic	Exact
COB50CH0	338	120	Bold italic	Exact
COB50CJ0	338	133	Bold italic	Exact
COB50CN0	338	160	Bold italic	Exact
COB50CT0	338	200	Bold italic	Exact
COB50CZ0	338	240	Bold italic	Exact
COB50C00	338	67	Bold italic	Exact
COB50C50	338	33	Bold italic	Exact
COB50C60	338	40	Bold italic	Exact
COB50C70	338	47	Bold italic	Exact
COB50C80	338	53	Bold italic	Exact
COB50C90	338	60	Bold italic	Exact
COB500A0	338	73	Bold italic	Exact
COB500B0	338	80	Bold italic	Exact
COB500D0	338	93	Bold italic	Exact
COB500F0	338	107	Bold italic	Exact
COB500H0	338	120	Bold italic	Exact
COB500J0	338	133	Bold italic	Exact
COB500N0	338	160	Bold italic	Exact
COB500T0	338	200	Bold italic	Exact
COB500Z0	338	240	Bold italic	Exact
COB50000	338	67	Bold italic	Exact
COB50050	338	33	Bold italic	Exact
COB50060	338	40	Bold italic	Exact
COB50070	338	47	Bold italic	Exact
COB50080	338	53	Bold italic	Exact
COB50090	338	60	Bold italic	Exact
COB60CA0	339	73	Reverse	Exact
COB60CB0	339	80	Reverse	Exact
COB60CD0	339	93	Reverse	Exact
COB60CF0	339	107	Reverse	Exact
COB60CH0	339	120	Reverse	Exact
COB60CJ0	339	133	Reverse	Exact
COB60CN0	339	160	Reverse	Exact

Table 14. Host Resident to Printer Resident Font Character Set Mapping (continued)

Font character set name	Registered font ID	Font width	Font attributes	Map fidelity
COB60CT0	339	200	Reverse	Exact
COB60CZ0	339	240	Reverse	Exact
COB60C00	339	67	Reverse	Exact
COB60C50	339	33	Reverse	Exact
COB60C60	339	40	Reverse	Exact
COB60C70	339	47	Reverse	Exact
COB60C80	339	53	Reverse	Exact
COB60C90	339	60	Reverse	Exact
COB600A0	339	73	Reverse	Exact
COB600B0	339	80	Reverse	Exact
COB600D0	339	93	Reverse	Exact
COB600F0	339	107	Reverse	Exact
COB600H0	339	120	Reverse	Exact
COB600J0	339	133	Reverse	Exact
COB600N0	339	160	Reverse	Exact
COB600T0	339	200	Reverse	Exact
COB600Z0	339	240	Reverse	Exact
COB60000	339	67	Reverse	Exact
COB60050	339	33	Reverse	Exact
COB60060	339	40	Reverse	Exact
COB60070	339	47	Reverse	Exact
COB60080	339	53	Reverse	Exact
COB60090	339	60	Reverse	Exact
C0C055A0	16951	73	Normal	Exact
C0C055B0	16951	80	Normal	Exact
C0C055D0	16951	93	Normal	Exact
C0C055F0	16951	107	Normal	Exact
C0C055H0	16951	120	Normal	Exact
C0C055J0	16951	133	Normal	Exact
C0C055N0	16951	160	Normal	Exact
C0C055T0	16951	200	Normal	Exact
C0C055Z0	16951	240	Normal	Exact
C0C05500	16951	67	Normal	Exact
C0C05560	16951	40	Normal	Exact
C0C05570	16951	47	Normal	Exact
C0C05580	16951	53	Normal	Exact
C0C05590	16951	60	Normal	Exact
C0C075A0	16971	73	Bold	Exact
C0C075B0	16971	80	Bold	Exact

Table 14. Host Resident to Printer Resident Font Character Set Mapping (continued)

Font character set name	Registered font ID	Font width	Font attributes	Map fidelity
C0C075D0	16971	93	Bold	Exact
C0C075F0	16971	107	Bold	Exact
C0C075H0	16971	120	Bold	Exact
C0C075J0	16971	133	Bold	Exact
C0C075N0	16971	160	Bold	Exact
C0C075T0	16971	200	Bold	Exact
C0C075Z0	16971	240	Bold	Exact
C0C07500	16971	67	Bold	Exact
C0C07560	16971	40	Bold	Exact
C0C07570	16971	47	Bold	Exact
C0C07580	16971	53	Bold	Exact
C0C07590	16971	60	Bold	Exact
C0C155A0	17079	73	Italic	Exact
C0C155B0	17079	80	Italic	Exact
C0C155D0	17079	93	Italic	Exact
C0C155F0	17079	107	Italic	Exact
C0C155H0	17079	120	Italic	Exact
C0C155J0	17079	133	Italic	Exact
C0C155N0	17079	160	Italic	Exact
C0C155T0	17079	200	Italic	Exact
C0C155Z0	17079	240	Italic	Exact
C0C15500	17079	67	Italic	Exact
C0C15560	17079	40	Italic	Exact
C0C15570	17079	47	Italic	Exact
C0C15580	17079	53	Italic	Exact
C0C15590	17079	60	Italic	Exact
C0C175A0	17099	73	Bold italic	Exact
C0C175B0	17099	80	Bold italic	Exact
C0C175D0	17099	93	Bold italic	Exact
C0C175F0	17099	107	Bold italic	Exact
C0C175H0	17099	120	Bold italic	Exact
C0C175J0	17099	133	Bold italic	Exact
C0C175N0	17099	160	Bold italic	Exact
C0C175T0	17099	200	Bold italic	Exact
C0C175Z0	17099	240	Bold italic	Exact
C0C17500	17099	67	Bold italic	Exact
C0C17560	17099	40	Bold italic	Exact
C0C17570	17099	47	Bold italic	Exact
C0C17580	17099	53	Bold italic	Exact

Table 14. Host Resident to Printer Resident Font Character Set Mapping (continued)

Font character set name	Registered font ID	Font width	Font attributes	Map fidelity
C0C17590	17099	60	Bold italic	Exact
C0D0GB10	39	144	Bold	Exact
C0D0GB12	69	120	Bold	Exact
C0D0GC15	231	96	Normal	Not exact
C0D0GI12	68	120	Italic	Exact
C0D0GL10	303	144	Normal	Not exact
C0D0GL12	303	120	Normal	Not exact
C0D0GL15	303	96	Normal	Not exact
C0D0GP12	174	120	Normal	Exact
C0D0GR10	310	144	Normal	Not exact
C0D0GT10	40	144	Normal	Exact
C0D0GT12	66	120	Normal	Exact
C0D0GT13	203	108	Normal	Exact
C0D0GT15	230	96	Normal	Exact
C0D0GT18	275	78	Normal	Not exact
C0D0GT20	281	72	Normal	Exact
C0D0GT24	290	54	Normal	Exact
C0D0RT10	41	144	Normal	Exact
C0D0SB12	72	120	Bold	Exact
C0D0SI10	43	144	Italic	Exact
C0D0SI12	71	120	Italic	Exact
C0D0SO12	332	120	Normal	Not exact
C0D0ST10	42	144	Normal	Exact
C0D0ST12	70	120	Normal	Exact
C0D0ST15	229	96	Normal	Exact
C0G055A0	4663	73	Normal	Exact
C0G055B0	4663	80	Normal	Exact
C0G055D0	4663	93	Normal	Exact
C0G055F0	4663	107	Normal	Exact
C0G055H0	4663	120	Normal	Exact
C0G055J0	4663	133	Normal	Exact
C0G055N0	4663	160	Normal	Exact
C0G055T0	4663	200	Normal	Exact
C0G055Z0	4663	240	Normal	Exact
C0G05500	4663	67	Normal	Exact
C0G05560	4663	40	Normal	Exact
C0G05570	4663	47	Normal	Exact
C0G05580	4663	53	Normal	Exact
C0G05590	4663	60	Normal	Exact

Table 14. Host Resident to Printer Resident Font Character Set Mapping (continued)

Font character set name	Registered font ID	Font width	Font attributes	Map fidelity
C0G075A0	4683	73	Bold	Exact
C0G075B0	4683	80	Bold	Exact
C0G075D0	4683	93	Bold	Exact
C0G075F0	4683	107	Bold	Exact
C0G075H0	4683	120	Bold	Exact
C0G075J0	4683	133	Bold	Exact
C0G075N0	4683	160	Bold	Exact
C0G075T0	4683	200	Bold	Exact
C0G075Z0	4683	240	Bold	Exact
C0G07500	4683	67	Bold	Exact
C0G07560	4683	40	Bold	Exact
C0G07570	4683	47	Bold	Exact
C0G07580	4683	53	Bold	Exact
C0G07590	4683	60	Bold	Exact
C0G155A0	4791	73	Italic	Exact
C0G155B0	4791	80	Italic	Exact
C0G155D0	4791	93	Italic	Exact
C0G155F0	4791	107	Italic	Exact
C0G155H0	4791	120	Italic	Exact
C0G155J0	4791	133	Italic	Exact
C0G155N0	4791	160	Italic	Exact
C0G155T0	4791	200	Italic	Exact
C0G155Z0	4791	240	Italic	Exact
C0G15500	4791	67	Italic	Exact
C0G15560	4791	40	Italic	Exact
C0G15570	4791	47	Italic	Exact
C0G15580	4791	53	Italic	Exact
C0G15590	4791	60	Italic	Exact
C0G175A0	4811	73	Bold italic	Exact
C0G175B0	4811	80	Bold italic	Exact
C0G175D0	4811	93	Bold italic	Exact
C0G175F0	4811	107	Bold italic	Exact
C0G175H0	4811	120	Bold italic	Exact
C0G175J0	4811	133	Bold italic	Exact
C0G175N0	4811	160	Bold italic	Exact
C0G175T0	4811	200	Bold italic	Exact
C0G175Z0	4811	240	Bold italic	Exact
C0G17500	4811	67	Bold italic	Exact
C0G17560	4811	40	Bold italic	Exact

Table 14. Host Resident to Printer Resident Font Character Set Mapping (continued)

Font character set name	Registered font ID	Font width	Font attributes	Map fidelity
C0G17570	4811	47	Bold italic	Exact
C0G17580	4811	53	Bold italic	Exact
C0G17590	4811	60	Bold italic	Exact
C0H200A0	2304	73	Normal	Exact
C0H200B0	2304	80	Normal	Exact
C0H200D0	2304	93	Normal	Exact
C0H200F0	2304	107	Normal	Exact
C0H200H0	2304	120	Normal	Exact
C0H200J0	2304	133	Normal	Exact
C0H200N0	2304	160	Normal	Exact
C0H200T0	2304	200	Normal	Exact
C0H200Z0	2304	240	Normal	Exact
C0H20000	2304	67	Normal	Exact
C0H20060	2304	40	Normal	Exact
C0H20070	2304	47	Normal	Exact
C0H20080	2304	53	Normal	Exact
C0H20090	2304	60	Normal	Exact
C0H300A0	2306	73	Italic	Exact
C0H300B0	2306	80	Italic	Exact
C0H300D0	2306	93	Italic	Exact
C0H300F0	2306	107	Italic	Exact
C0H300H0	2306	120	Italic	Exact
C0H300J0	2306	133	Italic	Exact
C0H300N0	2306	160	Italic	Exact
C0H300T0	2306	200	Italic	Exact
C0H300Z0	2306	240	Italic	Exact
C0H30000	2306	67	Italic	Exact
C0H30060	2306	40	Italic	Exact
C0H30070	2306	47	Italic	Exact
C0H30080	2306	53	Italic	Exact
C0H30090	2306	60	Italic	Exact
C0H400A0	2305	73	Bold	Exact
C0H400B0	2305	80	Bold	Exact
C0H400D0	2305	93	Bold	Exact
C0H400F0	2305	107	Bold	Exact
C0H400H0	2305	120	Bold	Exact
C0H400J0	2305	133	Bold	Exact
C0H400N0	2305	160	Bold	Exact
C0H400T0	2305	200	Bold	Exact

Table 14. Host Resident to Printer Resident Font Character Set Mapping (continued)

Font character set name	Registered font ID	Font width	Font attributes	Map fidelity
C0H400Z0	2305	240	Bold	Exact
C0H40000	2305	67	Bold	Exact
C0H40060	2305	40	Bold	Exact
C0H40070	2305	47	Bold	Exact
C0H40080	2305	53	Bold	Exact
C0H40090	2305	60	Bold	Exact
C0H500A0	2307	73	Bold italic	Exact
C0H500B0	2307	80	Bold italic	Exact
C0H500D0	2307	93	Bold italic	Exact
C0H500F0	2307	107	Bold italic	Exact
C0H500H0	2307	120	Bold italic	Exact
C0H500J0	2307	133	Bold italic	Exact
C0H500N0	2307	160	Bold italic	Exact
C0H500T0	2307	200	Bold italic	Exact
C0H500Z0	2307	240	Bold italic	Exact
C0H50000	2307	67	Bold italic	Exact
C0H50060	2307	40	Bold italic	Exact
C0H50070	2307	47	Bold italic	Exact
C0H50080	2307	53	Bold italic	Exact
C0H50090	2307	60	Bold italic	Exact
C0J055J0	37431	133	Normal	Not exact
C0J055Z0	37431	240	Normal	Not exact
C0L0AD10	45	144	Normal	Exact
C0L0AD12	76	120	Normal	Exact
C0L0AG10	45	144	Normal	Not exact
C0L0AG12	76	120	Normal	Not exact
C0L0AG15	219	96	Normal	Not exact
C0L0AI10	58	144	Italic	Not exact
C0L0AI12	105	120	Italic	Exact
C0L0AT10	45	144	Normal	Not exact
C0L0AT12	76	120	Normal	Not exact
C0L0DUMP	230	96	Normal	Not exact
C0L0FM10	30	144	Normal	Not exact
C0L0FM12	80	120	Normal	Not exact
C0L0FM15	225	96	Normal	Not exact
C0L0GU10	312	144	Normal	Not exact
C0L0GU12	312	120	Normal	Not exact
C0L0GU15	312	96	Normal	Not exact
C0L0KATA	433	144	Normal	Exact



Table 14. Host Resident to Printer Resident Font Character Set Mapping (continued)

Font character set name	Registered font ID	Font width	Font attributes	Map fidelity
C0L0KN12	433	120	Normal	Exact
C0L0KN20	433	84	Normal	Not exact
C0L0TU10	334	144	Normal	Not exact
C0L00AOA	19	144	Normal	Exact
C0L00AON	19	144	Normal	Exact
C0L00APL	45	144	Normal	Not exact
C0L00BOA	3	144	Normal	Exact
C0L00BON	3	144	Normal	Exact
C0L00GSC	398	96	Normal	Not exact
C0L00GUC	311	96	Normal	Not exact
C0L00OAB	3	144	Normal	Exact
C0L00T11	333	144	Normal	Not exact
C0MO55A0	50231	73	Normal	Not exact
C0MO55B0	50231	80	Normal	Not exact
C0MO55B1	50231	320	Normal	Not exact
C0MO55D0	50231	93	Normal	Not exact
C0MO55F0	50231	107	Normal	Not exact
C0MO55H0	50231	120	Normal	Not exact
C0MO55H1	50231	360	Normal	Not exact
C0MO55J0	50231	133	Normal	Not exact
C0MO55L0	50231	147	Normal	Not exact
C0MO55N0	50231	160	Normal	Not exact
C0MO55N1	50231	400	Normal	Not exact
C0MO55R0	50231	187	Normal	Not exact
C0MO55T0	50231	200	Normal	Not exact
C0MO55V0	50231	213	Normal	Not exact
C0MO55Z0	50231	240	Normal	Not exact
C0MO55Z1	50231	480	Normal	Not exact
C0MO5500	50231	67	Normal	Not exact
C0MO5541	50231	267	Normal	Not exact
C0MO5560	50231	40	Normal	Not exact
C0MO5570	50231	47	Normal	Not exact
C0MO5580	50231	53	Normal	Not exact
C0MO5581	50231	293	Normal	Not exact
C0MO5590	50231	60	Normal	Not exact
C0MP55A0	49463	73	Normal	Not exact
C0MP55B0	49463	80	Normal	Not exact
C0MP55D0	49463	93	Normal	Not exact
C0MP55F0	49463	107	Normal	Not exact

Table 14. Host Resident to Printer Resident Font Character Set Mapping (continued)

Font character set name	Registered font ID	Font width	Font attributes	Map fidelity
C0MP55H0	49463	120	Normal	Not exact
C0MP55N0	49463	160	Normal	Not exact
C0MP55Z0	49463	240	Normal	Not exact
C0MP5500	49463	67	Normal	Not exact
C0MP5560	49463	40	Normal	Not exact
C0MP5570	49463	47	Normal	Not exact
C0MP5580	49463	53	Normal	Not exact
C0MP5590	49463	60	Normal	Not exact
C0MP75A0	49483	73	Bold	Not exact
C0MP75B0	49483	80	Bold	Not exact
C0MP75D0	49483	93	Bold	Not exact
C0MP75F0	49483	107	Bold	Not exact
C0MP75H0	49483	120	Bold	Not exact
C0MP75N0	49483	160	Bold	Not exact
C0MP75Z0	49483	240	Bold	Not exact
C0MP7500	49483	67	Bold	Not exact
C0MP7560	49483	40	Bold	Not exact
C0MP7570	49483	47	Bold	Not exact
C0MP7580	49483	53	Bold	Not exact
C0MP7590	49483	60	Bold	Not exact
C0MQ55A0	49719	73	Normal	Not exact
C0MQ55B0	49719	80	Normal	Not exact
C0MQ55D0	49719	93	Normal	Not exact
C0MQ55F0	49719	107	Normal	Not exact
C0MQ55H0	49719	120	Normal	Not exact
C0MQ55N0	49719	160	Normal	Not exact
C0MQ55Z0	49719	240	Normal	Not exact
C0MQ5500	49719	67	Normal	Not exact
C0MQ5560	49719	40	Normal	Not exact
C0MQ5570	49719	47	Normal	Not exact
C0MQ5580	49719	53	Normal	Not exact
C0MQ5590	49719	60	Normal	Not exact
C0MQ75A0	49739	73	Bold	Not exact
C0MQ75B0	49739	80	Bold	Not exact
C0MQ75D0	49739	93	Bold	Not exact
C0MQ75F0	49739	107	Bold	Not exact
C0MQ75H0	49739	120	Bold	Not exact
C0MQ75N0	49739	160	Bold	Not exact
C0MQ75Z0	49739	240	Bold	Not exact

Table 14. Host Resident to Printer Resident Font Character Set Mapping (continued)

Font character set name	Registered font ID	Font width	Font attributes	Map fidelity
C0MQ7500	49739	67	Bold	Not exact
C0MQ7560	49739	40	Bold	Not exact
C0MQ7570	49739	47	Bold	Not exact
C0MQ7580	49739	53	Bold	Not exact
C0MQ7590	49739	60	Bold	Not exact
C0N200A0	2308	73	Normal	Exact
C0N200B0	2308	80	Normal	Exact
C0N200D0	2308	93	Normal	Exact
C0N200F0	2308	107	Normal	Exact
C0N200H0	2308	120	Normal	Exact
C0N200J0	2308	133	Normal	Exact
C0N200N0	2308	160	Normal	Exact
C0N200T0	2308	200	Normal	Exact
C0N200Z0	2308	240	Normal	Exact
C0N20000	2308	67	Normal	Exact
C0N20060	2308	40	Normal	Exact
C0N20070	2308	47	Normal	Exact
C0N20080	2308	53	Normal	Exact
C0N20090	2308	60	Normal	Exact
C0N204B0	2308	80	Normal	Not exact
C0N300A0	2310	73	Italic	Exact
C0N300B0	2310	80	Italic	Exact
C0N300D0	2310	93	Italic	Exact
C0N300F0	2310	107	Italic	Exact
C0N300H0	2310	120	Italic	Exact
C0N300J0	2310	133	Italic	Exact
C0N300N0	2310	160	Italic	Exact
C0N300T0	2310	200	Italic	Exact
C0N300Z0	2310	240	Italic	Exact
C0N30000	2310	67	Italic	Exact
C0N30060	2310	40	Italic	Exact
C0N30070	2310	47	Italic	Exact
C0N30080	2310	53	Italic	Exact
C0N30090	2310	60	Italic	Exact
C0N304B0	2310	80	Italic	Not exact
C0N400A0	2309	73	Bold	Exact
C0N400B0	2309	80	Bold	Exact
C0N400D0	2309	93	Bold	Exact
C0N400F0	2309	107	Bold	Exact

Table 14. Host Resident to Printer Resident Font Character Set Mapping (continued)

Font character set name	Registered font ID	Font width	Font attributes	Map fidelity
C0N400H0	2309	120	Bold	Exact
C0N400J0	2309	133	Bold	Exact
C0N400N0	2309	160	Bold	Exact
C0N400T0	2309	200	Bold	Exact
C0N400Z0	2309	240	Bold	Exact
C0N40000	2309	67	Bold	Exact
C0N40060	2309	40	Bold	Exact
C0N40070	2309	47	Bold	Exact
C0N40080	2309	53	Bold	Exact
C0N40090	2309	60	Bold	Exact
C0N404B0	2309	80	Bold	Not exact
C0N404H0	2309	120	Bold	Not exact
C0N404N0	2309	160	Bold	Not exact
C0N500A0	2311	73	Bold italic	Exact
C0N500B0	2311	80	Bold italic	Exact
C0N500D0	2311	93	Bold italic	Exact
C0N500F0	2311	107	Bold italic	Exact
C0N500H0	2311	120	Bold italic	Exact
C0N500J0	2311	133	Bold italic	Exact
C0N500N0	2311	160	Bold italic	Exact
C0N500T0	2311	200	Bold italic	Exact
C0N500Z0	2311	240	Bold italic	Exact
C0N50000	2311	67	Bold italic	Exact
C0N50060	2311	40	Bold italic	Exact
C0N50070	2311	47	Bold italic	Exact
C0N50080	2311	53	Bold italic	Exact
C0N50090	2311	60	Bold italic	Exact
C0N504B0	2311	80	Bold italic	Not exact
C0N504H0	2311	120	Bold italic	Not exact
C0N504N0	2311	160	Bold italic	Not exact
C0OCRA10	19	144	Normal	Exact
C0OCRB10	3	144	Normal	Exact
C0P055B0	49719	80	Normal	Exact
C0P05500	49719	67	Normal	Exact
C0P05560	49719	40	Normal	Exact
C0P05580	49719	53	Normal	Exact
C0P075B0	49739	80	Bold	Exact
C0P07500	49739	67	Bold	Exact
C0P07560	49739	40	Bold	Exact

Table 14. Host Resident to Printer Resident Font Character Set Mapping (continued)

Font character set name	Registered font ID	Font width	Font attributes	Map fidelity
C0P07580	49739	53	Bold	Exact
C0Q055B0	49719	80	Normal	Not exact
C0Q05500	49463	67	Normal	Not exact
C0Q05560	49463	40	Normal	Not exact
C0Q05580	49463	53	Normal	Not exact
C0Q075B0	49483	80	Bold	Not exact
C0Q07500	49483	67	Bold	Not exact
C0Q07560	49483	40	Bold	Not exact
C0Q07580	49483	53	Bold	Not exact
C0S0AE10	45	144	Normal	Exact
C0S0AE20	280	72	Normal	Exact
C0S0AP13	206	108	Normal	Not exact
C0S0AP20	280	72	Normal	Not exact
C0S0BITR	155	120	Italic	Exact
C0S0BRTR	159	120	Normal	Exact
C0S0CB10	46	144	Bold	Exact
C0S0CB12	108	120	Bold	Exact
C0S0CB15	214	96	Bold	Exact
C0S0CD15	417	96	Double Wide	Exact
C0S0CE10	11	144	Normal	Not exact
C0S0CE12	85	120	Normal	Exact
C0S0CH10	37	144	Normal	Not exact
C0S0CI10	18	144	Italic	Exact
C0S0CI12	92	120	Italic	Exact
C0S0CI15	215	96	Italic	Exact
C0S0CO10	302	144	Normal	Not exact
C0S0CR10	11	144	Normal	Exact
C0S0CR12	85	120	Normal	Exact
C0S0CR15	223	96	Normal	Exact
C0S0CW15	425	96	Double Wide Italic	Exact
C0S0DOTR	175	120	Normal	Exact
C0S0D224	203	108	Normal	Exact
C0S0D225	203	108	Normal	Exact
C0S0D226	201	108	Bold	Exact
C0S0D227	202	108	Italic	Exact
C0S0EBTR	163	120	Bold	Exact
C0S0EITR	162	120	Italic	Exact
C0S0ELTR	173	120	Normal	Exact
C0S0EOTR	196	120	Normal	Not exact

Table 14. Host Resident to Printer Resident Font Character Set Mapping (continued)

Font character set name	Registered font ID	Font width	Font attributes	Map fidelity
COS0ESTR	160	120	Normal	Exact
COS0LB12	110	120	Bold	Exact
COS0LR12	87	120	Normal	Exact
COS0OB10	38	144	Bold	Exact
COS0OR10	5	144	Normal	Exact
COS0PB12	111	120	Bold	Exact
COS0PI12	112	120	Italic	Exact
COS0PR10	12	144	Normal	Exact
COS0PR12	86	120	Normal	Exact
COS0SR12	84	120	Normal	Exact
COS0SYM0	49975	67	Normal	Not exact
COS0SYM2	49975	80	Normal	Not exact
COS0S192	80	120	Normal	Not exact
COS0S193	80	120	Normal	Not exact
COS0S198	30	144	Normal	Not exact
COS055A0	28983	73	Normal	Exact
COS055B0	28983	80	Normal	Exact
COS055D0	28983	93	Normal	Exact
COS055F0	28983	107	Normal	Exact
COS055H0	28983	120	Normal	Exact
COS055J0	28983	133	Normal	Exact
COS055N0	28983	160	Normal	Exact
COS055T0	28983	200	Normal	Exact
COS055Z0	28983	240	Normal	Exact
COS05500	28983	67	Normal	Exact
COS05560	28983	40	Normal	Exact
COS05570	28983	47	Normal	Exact
COS05580	28983	53	Normal	Exact
COS05590	28983	60	Normal	Exact
COS075A0	29003	73	Bold	Exact
COS075B0	29003	80	Bold	Exact
COS075D0	29003	93	Bold	Exact
COS075F0	29003	107	Bold	Exact
COS075H0	29003	120	Bold	Exact
COS075J0	29003	133	Bold	Exact
COS075N0	29003	160	Bold	Exact
COS075T0	29003	200	Bold	Exact
COS075Z0	29003	240	Bold	Exact
COS07500	29003	67	Bold	Exact

Table 14. Host Resident to Printer Resident Font Character Set Mapping (continued)

Font character set name	Registered font ID	Font width	Font attributes	Map fidelity
C0S07560	29003	40	Bold	Exact
C0S07570	29003	47	Bold	Exact
C0S07580	29003	53	Bold	Exact
C0S07590	29003	60	Bold	Exact
C0S155A0	29111	73	Italic	Exact
C0S155B0	29111	80	Italic	Exact
C0S155D0	29111	93	Italic	Exact
C0S155F0	29111	107	Italic	Exact
C0S155H0	29111	120	Italic	Exact
C0S155J0	29111	133	Italic	Exact
C0S155N0	29111	160	Italic	Exact
C0S155T0	29111	200	Italic	Exact
C0S155Z0	29111	240	Italic	Exact
C0S15500	29111	67	Italic	Exact
C0S15560	29111	40	Italic	Exact
C0S15570	29111	47	Italic	Exact
C0S15580	29111	53	Italic	Exact
C0S15590	29111	60	Italic	Exact
C0S175A0	29131	73	Bold italic	Exact
C0S175B0	29131	80	Bold italic	Exact
C0S175D0	29131	93	Bold italic	Exact
C0S175F0	29131	107	Bold italic	Exact
C0S175H0	29131	120	Bold italic	Exact
C0S175J0	29131	133	Bold italic	Exact
C0S175N0	29131	160	Bold italic	Exact
C0S175T0	29131	200	Bold italic	Exact
C0S175Z0	29131	240	Bold italic	Exact
C0S17500	29131	67	Bold italic	Exact
C0S17560	29131	40	Bold italic	Exact
C0S17570	29131	47	Bold italic	Exact
C0S17580	29131	53	Bold italic	Exact
C0S17590	29131	60	Bold italic	Exact
C0T055A0	4407	73	Normal	Exact
C0T055B0	4407	80	Normal	Exact
C0T055B1	4407	320	Normal	Exact
C0T055D0	4407	93	Normal	Exact
C0T055F0	4407	107	Normal	Exact
C0T055H0	4407	120	Normal	Exact
C0T055J0	4407	133	Normal	Exact

Table 14. Host Resident to Printer Resident Font Character Set Mapping (continued)

Font character set name	Registered font ID	Font width	Font attributes	Map fidelity
C0T055N0	4407	160	Normal	Exact
C0T055N1	4407	400	Normal	Exact
C0T055T0	4407	200	Normal	Exact
C0T055Z0	4407	240	Normal	Exact
C0T055Z1	4407	480	Normal	Exact
C0T05500	4407	67	Normal	Exact
C0T05560	4407	40	Normal	Exact
C0T05570	4407	47	Normal	Exact
C0T05580	4407	53	Normal	Exact
C0T05590	4407	60	Normal	Exact
C0T075A0	4427	73	Bold	Exact
C0T075B0	4427	80	Bold	Exact
C0T075B1	4427	320	Bold	Exact
C0T075D0	4427	93	Bold	Exact
C0T075F0	4427	107	Bold	Exact
C0T075H0	4427	120	Bold	Exact
C0T075J0	4427	133	Bold	Exact
C0T075N0	4427	160	Bold	Exact
C0T075N1	4427	400	Bold	Exact
C0T075T0	4427	200	Bold	Exact
C0T075Z0	4427	240	Bold	Exact
C0T075Z1	4427	480	Bold	Exact
C0T07500	4427	67	Bold	Exact
C0T07560	4427	40	Bold	Exact
C0T07570	4427	47	Bold	Exact
C0T07580	4427	53	Bold	Exact
C0T07590	4427	60	Bold	Exact
C0T155A0	4535	73	Italic	Exact
C0T155B0	4535	80	Italic	Exact
C0T155B1	4535	320	Italic	Exact
C0T155D0	4535	93	Italic	Exact
C0T155F0	4535	107	Italic	Exact
C0T155H0	4535	120	Italic	Exact
C0T155J0	4535	133	Italic	Exact
C0T155N0	4535	160	Italic	Exact
C0T155N1	4535	400	Italic	Exact
C0T155T0	4535	200	Italic	Exact
C0T155Z0	4535	240	Italic	Exact
C0T155Z1	4535	480	Italic	Exact



Table 14. Host Resident to Printer Resident Font Character Set Mapping (continued)

Font character set name	Registered font ID	Font width	Font attributes	Map fidelity
C0T15500	4535	67	Italic	Exact
C0T15560	4535	40	Italic	Exact
C0T15570	4535	47	Italic	Exact
C0T15580	4535	53	Italic	Exact
C0T15590	4535	60	Italic	Exact
C0T175A0	4555	73	Bold italic	Exact
C0T175B0	4555	80	Bold italic	Exact
C0T175B1	4555	320	Bold italic	Exact
C0T175D0	4555	93	Bold italic	Exact
C0T175F0	4555	107	Bold italic	Exact
C0T175H0	4555	120	Bold italic	Exact
C0T175J0	4555	133	Bold italic	Exact
C0T175N0	4555	160	Bold italic	Exact
C0T175N1	4555	400	Bold italic	Exact
C0T175T0	4555	200	Bold italic	Exact
C0T175Z0	4555	240	Bold italic	Exact
C0T175Z1	4555	480	Bold italic	Exact
C0T17500	4555	67	Bold italic	Exact
C0T17560	4555	40	Bold italic	Exact
C0T17570	4555	47	Bold italic	Exact
C0T17580	4555	53	Bold italic	Exact
C0T17590	4555	60	Bold italic	Exact
C0V055A0	33847	73	Normal	Exact
C0V055B0	33847	80	Normal	Exact
C0V055D0	33847	93	Normal	Exact
C0V055F0	33847	107	Normal	Exact
C0V055H0	33847	120	Normal	Exact
C0V055J0	33847	133	Normal	Exact
C0V055N0	33847	160	Normal	Exact
C0V055T0	33847	200	Normal	Exact
C0V055Z0	33847	240	Normal	Exact
C0V05500	33847	67	Normal	Exact
C0V05560	33847	40	Normal	Exact
C0V05570	33847	47	Normal	Exact
C0V05580	33847	53	Normal	Exact
C0V05590	33847	60	Normal	Exact
C0V075A0	33867	73	Bold	Exact
C0V075B0	33867	80	Bold	Exact
C0V075D0	33867	93	Bold	Exact

Table 14. Host Resident to Printer Resident Font Character Set Mapping (continued)

Font character set name	Registered font ID	Font width	Font attributes	Map fidelity
C0V075F0	33867	107	Bold	Exact
C0V075H0	33867	120	Bold	Exact
C0V075J0	33867	133	Bold	Exact
C0V075N0	33867	160	Bold	Exact
C0V075T0	33867	200	Bold	Exact
C0V075Z0	33867	240	Bold	Exact
C0V07500	33867	67	Bold	Exact
C0V07560	33867	40	Bold	Exact
C0V07570	33867	47	Bold	Exact
C0V07580	33867	53	Bold	Exact
C0V07590	33867	60	Bold	Exact
C0V155A0	33975	73	Italic	Exact
C0V155B0	33975	80	Italic	Exact
C0V155D0	33975	93	Italic	Exact
C0V155F0	33975	107	Italic	Exact
C0V155H0	33975	120	Italic	Exact
C0V155J0	33975	133	Italic	Exact
C0V155N0	33975	160	Italic	Exact
C0V155T0	33975	200	Italic	Exact
C0V155Z0	33975	240	Italic	Exact
C0V15500	33975	67	Italic	Exact
C0V15560	33975	40	Italic	Exact
C0V15570	33975	47	Italic	Exact
C0V15580	33975	53	Italic	Exact
C0V15590	33975	60	Italic	Exact
C0V175A0	33995	73	Bold italic	Exact
C0V175B0	33995	80	Bold italic	Exact
C0V175D0	33995	93	Bold italic	Exact
C0V175F0	33995	107	Bold italic	Exact
C0V175H0	33995	120	Bold italic	Exact
C0V175J0	33995	133	Bold italic	Exact
C0V175N0	33995	160	Bold italic	Exact
C0V175T0	33995	200	Bold italic	Exact
C0V175Z0	33995	240	Bold italic	Exact
C0V17500	33995	67	Bold italic	Exact
C0V17560	33995	40	Bold italic	Exact
C0V17570	33995	47	Bold italic	Exact
C0V17580	33995	53	Bold italic	Exact
C0V17590	33995	60	Bold italic	Exact

Table 14. Host Resident to Printer Resident Font Character Set Mapping (continued)

Font character set name	Registered font ID	Font width	Font attributes	Map fidelity
C0Z05640	33080	27	Normal	Not exact
C04200B0	416	144	Normal	Exact
C04200D0	416	168	Normal	Exact
C04200J0	416	240	Normal	Exact
C0420000	416	120	Normal	Exact
C0420070	416	84	Normal	Exact
C0420080	416	96	Normal	Exact
C04202B0	416	144	Normal	Exact
C0420200	416	120	Normal	Exact
C0420270	416	84	Normal	Exact
C0420280	416	96	Normal	Exact
C04203B0	416	144	Normal	Exact
C0420300	416	120	Normal	Exact
C0420380	416	96	Normal	Exact
C04204B0	416	144	Normal	Exact
C0420400	416	120	Normal	Exact
C0420480	416	96	Normal	Not exact
C04205B0	49	416	Normal	Exact
C0420500	416	120	Normal	Exact
C0420570	416	84	Normal	Not exact
C0420580	416	96	Normal	Not exact
C04300B0	424	144	Italic	Exact
C04300D0	424	168	Italic	Exact
C04300J0	424	240	Italic	Exact
C0430000	424	120	Italic	Exact
C0430070	424	84	Italic	Exact
C0430080	424	96	Italic	Exact
C04304B0	424	144	Italic	Exact
C0430400	424	120	Italic	Exact
C04400B0	420	144	Bold	Exact
C04400D0	420	168	Bold	Exact
C04400J0	420	240	Bold	Exact
C0440000	420	120	Bold	Exact
C0440070	420	84	Bold	Exact
C0440080	420	96	Bold	Exact
C0440200	420	120	Bold	Not exact
C0440300	420	120	Bold	Not exact
C04404B0	420	144	Bold	Exact
C04404D0	420	168	Bold	Not exact

Table 14. Host Resident to Printer Resident Font Character Set Mapping (continued)

Font character set name	Registered font ID	Font width	Font attributes	Map fidelity
C0440470	420	84	Bold	Exact
C04405B0	420	144	Bold	Exact
C0440500	420	120	Bold	Not exact
C04500B0	428	144	Bold italic	Exact
C04500D0	428	168	Bold italic	Exact
C04500J0	428	240	Bold italic	Exact
C0450000	428	120	Bold italic	Exact
C0450070	428	84	Bold italic	Exact
C0450080	428	96	Bold italic	Exact
C0450300	428	120	Bold italic	Not exact
C04504B0	428	144	Bold italic	Exact
C04504D0	428	168	Bold italic	Not exact

### Mapping host-resident to printer-resident code pages

Code pages are necessary for jobs to print. Some printers have code pages stored in memory or on a font card (printer resident code page); other printers do not.

Refer to the following table if one of the following is true:

- Your spooled files are directed to a 4224, 4230, 4234, 4247, or 64xx printer that is configured as an AFP printer.
- Your spooled files have a host-resident code page (stored on the system) specified.

This table can help you determine what printer resident code page is substituted when your spooled file specifies a host resident code page instead of a registered code page identifier (ID).

This code page substitution is necessary because these printers do not support the downloading of 240-pel host resident fonts. Depending on the host resident code page name that is requested for a particular font reference, the appropriate registered code page ID value is selected to match (as closely as possible) your font request.

The Map fidelity column indicates whether the substituted printer resident code page is considered to be an exact match to the code page that is requested in your spooled file.

Table 15. Mapping host-resident to printer-resident code pages

Host-resident code page name	Registered code page ID	Map fidelity
T1GDP256	256	Exact
T1V10037	37	Exact
T1V10273	273	Exact
T1V10274	274	Exact
T1V10275	275	Exact
T1V10277	277	Exact
T1V10278	278	Exact
T1V10280	280	Exact
T1V10281	281	Exact

Table 15. Mapping host-resident to printer-resident code pages (continued)

Host-resident code page name	Registered code page ID	Map fidelity
T1V10282	282	Exact
T1V10284	284	Exact
T1V10285	285	Exact
T1000290	290	Exact
T1L02773	2102	Exact
T1L02774	2103	Exact
T1S0AE10	2108	Exact
T1V10297	297	Exact
T1000361	361	Exact
T1GI0361	2065	Exact
T1000382	382	Exact
T1GI0382	2067	Exact
T1GI0383	2068	Exact
T1000384	384	Exact
T1GI0384	2069	Exact
T1000385	385	Exact
T1GI0385	2070	Exact
T1000386	386	Exact
T1GI0386	2071	Exact
T1000387	387	Exact
T1GI0387	2072	Exact
T1000388	388	Exact
T1GI0388	2073	Exact
T1000389	389	Exact
T1GI0389	2074	Exact
T1000390	390	Exact
T1GI0390	2075	Exact
T1000391	391	Exact
T1GI0391	2076	Exact
T1000392	392	Exact
T1GI0392	2077	Exact
T1000393	393	Exact
T1GI0393	2077	Exact
T1000394	394	Exact
T1GI0394	2078	Exact
T1000395	395	Exact
T1GI0395	2079	Exact
T1000420	420	Exact
T1000424	424	Exact
T1GPI363	2066	Exact

Table 15. Mapping host-resident to printer-resident code pages (continued)

Host-resident code page name	Registered code page ID	Map fidelity
T1000437	437	Exact
T1V10500	500	Exact
T1000819	819	Exact
T1000850	850	Exact
T1000852	852	Exact
T1000857	857	Exact
T1000863	863	Exact
T1000870	870	Exact
T1V10871	871	Exact
T1000912	912	Exact
T1000920	920	Exact
T1001002	1002	Exact
T1D0BASE	1002	Exact
T1001003	1003	Exact
T1DCDCFS	1003	Exact
T1001004	1004	Exact
T1001026	1026	Exact

### Mapping printer-resident to host-resident font character sets

The table in this topic can help you determine what host-resident character sets (font character sets stored on the system) are downloaded to a 3820, 3825, 3827, 3828, 3829, 3831, 3835, or 3900 printer when your spooled file refers to a registered font identifier (font ID) instead of a host-resident font character set.

Often, when a spooled file is created on i5/OS, a font global identifier (FGID) or font ID is specified to be used when the spooled file is printed.

In the case of the 3820, 3825, 3827, 3828, 3829, 3831, 3835, and 3900 printers, font IDs are not supported. Therefore, when the spooled file is directed to these printers, the system substitutes font character sets that allow the spooled file to print on these printers.

In the case of \*CONTENT fidelity, depending on the registered font ID value, the font width value, and the font attributes that are requested for a particular font reference, the appropriate host-resident font character set is selected to match (as closely as possible) your font request.

In the case of \*ABSOLUTE fidelity, depending on the registered font ID value, the font width value, and the font attributes that are requested for a particular font reference, the appropriate host-resident font character set is selected to match exactly your font request. The system also ensures, for \*ABSOLUTE fidelity, that the character set is compatible with the code page when mapping from printer resident fonts to host-resident fonts.

If the FGID has an asterisk next to it, the appropriate identifier (CHRID) must be specified to match this entry in the font table regardless of what fidelity value has been specified.

The font width specifies the width of the blank character in 1440ths of an inch. This is an indicator of how many characters fit per inch of space on the paper.

Some FGIDs, such as 416, have multiple widths associated with them. The Font width column of the table is blank for these FGIDs. Also, the primary (first choice column in the table) and secondary (second choice column in the table) character set names of these FGIDs have an XX in the last two positions, which identify the size of the font. For uniformly spaced fonts, there are six widths: 84, 96, 120, 144, 168, and 240. For typographic fonts, there are 14 widths: 40, 47, 53, 60, 67, 73, 80, 93, 107, 120, 133, 160, 200 and 240. The system determines whether a uniformly spaced font or a typographic font is needed and then selects the host-resident character set name based on the width provided.

The first choice is used if it is present on your system. The second choice is used if the first choice cannot be found. The Map fidelity column indicates whether the first choice is considered to be an exact match to the printer resident font (font ID) that is requested in your spooled file. As a rule, the second choice is not considered to be an exact match.

If the first choice contains a metric-only font character set name, the system uses the second choice regardless of the fidelity setting. A metric-only font character set name begins with the characters *COE*.

If code page (CPGID) 259 (which is the symbol font code page) has been specified, this table is not used. Instead, if the FGID specified is a 10-pitch font, the C0S0SYM2 character set is substituted; otherwise, if the FGID specified is anything other than a 10-pitch font, the C0S0SYM0 character set is substituted.

*Table 16. Mapping printer-resident to host-resident font character sets*

Registered font ID	Font width	Font attributes	Map fidelity	Font character set name (first choice)	Font character set name (second choice)
2	144	Normal	Exact	C0E0DE10	C0S0CR12
2	144	Normal	Exact	C0E0DE0R	C0S0SYM0
3	144	Normal	Exact	C0L00BOA	
3	144	Normal	Exact	C0S0SYM2	
5	144	Normal	Exact	C0S0OR10	
5	144	Bold	Exact	C0S0OB10	
5	144	Normal	Exact	C0E0OR10	C0S0OR10
5	144	Normal	Exact	C0S0SYM2	
10	144	Normal	Exact	C0E0CY10	C04203B0
10	144	Normal	Exact	C0E0CY0R	C0S0SYM0
11	144	Normal	Exact	C0S0CR10	
11	144	Bold	Exact	C0S0CB10	
11	144	Italic	Exact	C0S0CI10	
11	144	Normal	Exact	C0S0SYM2	
11	144	Normal	Exact	C0E0CR0K	C04203B0
11	144	Normal	Exact	C0E0CR0F	C0S0CR10
11	144	Normal	Exact	C0E0CR0N	C04203B0
11	144	Normal	Exact	C0E0CR0Q	C04203B0
11	144	Normal	Exact	C0E0CR0G	C04202B0
11	144	Normal	Exact	C0E0CR0H	C04202B0
12	144	Normal	Exact	C0S0PR10	
12	144	Normal	Exact	C0S0SYM2	
12	144	Normal	Exact	C0E0PR0G	C04202B0
12	144	Normal	Exact	C0E0PR0H	C04202B0

Table 16. Mapping printer-resident to host-resident font character sets (continued)

Registered font ID	Font width	Font attributes	Map fidelity	Font character set name (first choice)	Font character set name (second choice)
13	144	Normal	Not exact	C0S0CR10	
13	144	Normal	Exact	C0S0SYM2	
13	144	Bold	Not exact	C0S0CB10	
18	144	Italic	Exact	C0S0CI10	
18	144	Normal	Exact	C0S0SYM2	
19	144	Normal	Exact	C0L00AOA	
19	144	Normal	Exact	C0S0SYM2	
20	144	Normal	Not exact	C0S0CR10	
20	144	Bold	Not exact	C0S0CB10	
20	144	Normal	Exact	C0S0SYM2	
21	144	Normal	Exact	C0E0KA10	C0L0KATA
25	144	Normal	Exact	C0E0PS10	C0S0CR10
26	144	Normal	Exact	C0L0KATA	
26	144	Normal	Exact	C0S0AE10	
26	144	Normal	Exact	C0D0GT10	
26	144	Bold	Exact	C0D0GB10	
30	144	Normal	Exact	C0S0S198	
30	144	Normal	Exact	C0S0SYM2	
36	144	Normal	Exact	C0E0LR10	C0S0CR10
38	144	Bold	Exact	C0S0OB10	
38	144	Bold	Exact	C0S0SYM2	
39	144	Bold	Exact	C0D0GB10	
39	144	Bold	Exact	C0S0SYM2	
40	144	Normal	Exact	C0D0GT10	
40	144	Normal	Exact	C0S0SYM2	
41	144	Normal	Exact	C0D0RT10	
41	144	Normal	Exact	C0S0SYM2	
42	144	Normal	Exact	C0D0ST10	
42	144	Normal	Exact	C0S0SYM2	
43	144	Italic	Exact	C0D0SI10	
43	144	Italic	Exact	C0S0SYM2	
44	144	Normal	Exact	C0L0KATA	
44	144	Normal	Exact	C0S0SYM2	
45	144	Normal	Exact	C0S0AE10	
46	144	Bold	Exact	C0S0CB10	
46	144	Normal	Exact	C0S0SYM2	
49	144	Normal	Exact	C0E0HR10	C04205B0
50	144	Bold	Exact	C0H0HB10	C04405B0



Table 16. Mapping printer-resident to host-resident font character sets (continued)

Registered font ID	Font width	Font attributes	Map fidelity	Font character set name (first choice)	Font character set name (second choice)
50	144	Bold	Exact	C0E0HB10	C04405B0
50	144	Normal	Exact	C0S0SYM2	
51	144	Normal	Not exact	C0S0CR10	
51	144	Bold	Not exact	C0S0CB10	
52	144	Normal	Not exact	C0S0CR10	
52	144	Bold	Not exact	C0S0CB10	
61	144	Normal	Exact	C0E0NR10	C04204B0
62	144	Italic	Exact	C0E0NI10	C04304B0
63	144	Bold	Exact	C0E0NB10	C04404B0
64	144	Bold italic	Exact	C0E0NM10	C04504B0
66	120	Normal	Exact	C0D0GT12	
66	120	Bold	Exact	C0D0GB12	
66	120	Normal	Exact	C0S0SYM0	
68	120	Italic	Exact	C0D0GI12	
68	120	Italic	Exact	C0S0SYM0	
69	120	Bold	Exact	C0D0GB12	
69	120	Bold	Exact	C0S0SYM0	
70	120	Normal	Exact	C0D0ST12	
70	120	Normal	Exact	C0S0SYM0	
71	120	Italic	Exact	C0D0SI12	
71	120	Italic	Exact	C0S0SYM0	
72	120	Bold	Exact	C0D0SB12	
72	120	Bold	Exact	C0S0SYM0	
74	120	Normal	Not exact	C0S0CR12	
74	120	Bold	Not exact	C0S0CB12	
75	120	Normal	Not exact	C0S0CR12	
75	120	Bold	Not exact	C0S0CB12	
76	120	Normal	Exact	C0E0AP12	C0S0AE10
78	120	Normal	Exact	C0E0KA12	C0L0KN12
80	120	Normal	Not exact	C0S0CR12	
80	120	Bold	Not exact	C0S0CB12	
80	120	Normal	Exact	C0S0SYM0	
84	120	Normal	Exact	C0S0SR12	
84	120	Normal	Exact	C0E0SR12	C0S0SR12
84	120	Normal	Exact	C0S0SYM0	
85	120	Normal	Exact	C0S0CR12	
85	120	Normal	Exact	C0S0CE12	
85	120	Bold	Exact	C0S0CB12	

Table 16. Mapping printer-resident to host-resident font character sets (continued)

Registered font ID	Font width	Font attributes	Map fidelity	Font character set name (first choice)	Font character set name (second choice)
85	120	Italic	Exact	C0S0CI12	
85	120	Normal	Exact	C0S0SYM0	
85	120	Normal	Exact	C0E0CREF	C0420200
85	120	Normal	Exact	C0E0CREQ	C0420300
85	120	Normal	Exact	C0E0CREG	C0420200
85	120	Normal	Exact	C0E0CREH	C0420200
86	120	Normal	Exact	C0S0PR12	
86	120	Bold	Exact	C0S0PB12	
86	120	Normal	Exact	C0S0SYM0	
86	120	Normal	Exact	C0E0PREF	C0420200
86	120	Normal	Exact	C0E0PREQ	C0420300
86	120	Normal	Exact	C0E0PREG	C0420200
86	120	Normal	Exact	C0E0PREH	C0420200
87	120	Normal	Exact	C0S0LR12	
87	120	Bold	Exact	C0S0LB12	
87	120	Italic	Exact	C0D0GI12	
87	120	Normal	Exact	C0S0SYM0	
87	120	Normal	Exact	C0E0LRSR	C0S0LR12
87	120	Normal	Exact	C0E0LREK	C0420300
87	120	Normal	Exact	C0E0LREF	C0420200
87	120	Normal	Exact	C0E0LREN	C0420300
87	120	Normal	Exact	C0E0LREQ	C0420300
87	120	Normal	Exact	C0E0LREG	C0420200
87	120	Normal	Exact	C0E0LREH	C0420200
91	120	Italic	Not exact	C0S0CR12	
91	120	Bold italic	Not exact	C0S0CB12	
91	120	Italic	Exact	C0S0SYM0	
92	120	Italic	Exact	C0S0CI12	
92	120	Normal	Exact	C0E0CIER	C0S0SYM2
95	120	Normal	Exact	C0E0AJ12	C0S0CR12
95	120	Normal	Exact	C0E0AJER	C0S0SYM2
96	120	Bold	Exact	C0E0WB12	C0S0CR12
98	120	Normal	Exact	C0E0HR12	C0420500
103	120	Normal	Exact	C0E0NR12	C0420400
104	120	Italic	Exact	C0E0NI12	C0430400
108	120	Bold	Exact	C0S0CB12	
108	120	Normal	Exact	C0E0CBER	C0S0SYM2
109	120	Italic	Exact	C0E0LI12	C0S0CR12

Table 16. Mapping printer-resident to host-resident font character sets (continued)

Registered font ID	Font width	Font attributes	Map fidelity	Font character set name (first choice)	Font character set name (second choice)
109	120	Normal	Exact	C0E0LISR	C0S0SYM2
110	120	Bold	Exact	C0S0LB12	
110	120	Bold	Exact	C0E0LBEK	C0440300
110	120	Bold	Exact	C0E0LBEN	C0440300
110	120	Normal	Exact	C0S0SYM0	
111	120	Bold	Exact	C0S0PB12	
111	120	Normal	Exact	C0S0SYM0	
112	120	Italic	Exact	C0S0PI12	
112	120	Normal	Exact	C0S0SYM0	
154	120	Normal	Exact	C0S0ESTR	
154	120	Bold	Exact	C0S0EBTR	
155	120	Bold italic	Exact	C0S0BITR	
155	120	Bold italic	Exact	C0E0BIRK	C0450300
155	120	Bold italic	Exact	C0E0BIRN	C0450300
155	120	Bold	Exact	C0S0SYM0	
157	120	Bold	Exact	C0E0TBTR	C0S0ESTR
157	120	Bold	Exact	C0E0TBRF	C0420200
157	120	Normal	Exact	C0E0TBRR	C0S0SYM2
158	120	Normal	Not exact	C0S0ESTR	
158	120	Bold	Not exact	C0S0EBTR	
158	120	Normal	Exact	C0S0SYM0	
159	120	Bold	Exact	C0S0BRTR	
159	120	Bold	Exact	C0E0BRRK	C0440300
159	120	Bold	Exact	C0E0BRRN	C0440300
159	120	Bold	Exact	C0E0BRRQ	C0440300
159	120	Bold	Exact	C0E0BRRG	C0440200
159	120	Bold	Exact	C0E0BRRH	C0440200
159	120	Bold	Exact	C0S0SYM0	
160	120	Normal	Exact	C0S0ESTR	
160	120	Bold	Exact	C0S0EBTR	
160	120	Italic	Exact	C0S0EITR	
160	120	Normal	Exact	C0S0SYM0	
162	120	Italic	Exact	C0S0EITR	
162	120	Normal	Exact	C0S0SYM0	
163	120	Bold	Exact	C0S0EBTR	
163	120	Bold	Exact	C0S0SYM0	
164	120	Normal	Exact	C0E0PRTR	C0S0ESTR
164	120	Normal	Exact	C0E0PRRR	C0S0SYM2

Table 16. Mapping printer-resident to host-resident font character sets (continued)

Registered font ID	Font width	Font attributes	Map fidelity	Font character set name (first choice)	Font character set name (second choice)
167	120	Bold	Exact	C0E0BKTR	C0440500
168	120	Bold	Exact	C0H0BRK2	
168	120	Bold	Exact	C0S0SYM0	
173	120	Normal	Exact	C0S0ELTR	
173	120	Normal	Exact	C0S0SYM0	
174	120	Normal	Exact	C0D0GP12	
175	120	Normal	Exact	C0S0DOTR	
175	120	Normal	Exact	C0S0SYM0	
186	120	Normal	Exact	C0E0RRTR	C0S0ESTR
186	120	Normal	Exact	C0E0RRRR	C0S0SYM2
187	120	Bold	Exact	C0E0RBTR	C0S0ESTR
187	120	Normal	Exact	C0E0RBRR	C0S0SYM2
188	120	Italic	Exact	C0E0RITR	C0S0ESTR
188	120	Normal	Exact	C0E0RIRR	C0S0SYM2
189	120	Bold italic	Exact	C0E0RMTR	C0S0ESTR
189	120	Normal	Exact	C0E0RMRR	C0S0SYM2
190	120	Normal	Exact	C0E0FRTR	C0S0ESTR
190	120	Normal	Exact	C0E0FRRR	C0S0SYM2
191	120	Bold	Exact	C0E0FBTR	C0S0ESTR
191	120	Normal	Exact	C0E0FBRR	C0S0SYM2
194	120	Italic	Exact	C0E0FITR	C0S0ESTR
194	120	Normal	Exact	C0E0FIRR	C0S0SYM2
195	120	Bold italic	Exact	C0E0FMTR	C0S0ESTR
195	120	Normal	Exact	C0E0FMRR	C0S0SYM2
201	108	Bold	Exact	C0S0D226	C0D0GT13
202	108	Italic	Exact	C0S0D227	C0D0GT13
203	108	Normal	Exact	C0S0D224	C0D0GT13
203	108	Normal	Exact	C0S0D225	C0D0GT13
204	108	Normal	Exact	C0S0D224	C0D0GT13
204	108	Normal	Exact	C0S0D225	C0D0GT13
204	108	Bold	Exact	C0S0D226	C0D0GT13
204	108	Italic	Exact	C0S0D227	C0D0GT13
204	96	Normal	Not exact	C0S0CR15	
204	96	Bold	Not exact	C0S0CB15	
204	111	Normal	Exact	C0S0SYM0	
205	96	Normal	Not exact	C0S0CR15	C0D0GT13
205	96	Bold	Not exact	C0S0CB15	C0D0GT13
213	96	Normal	Exact	C0E0NR15	C0420480

Table 16. Mapping printer-resident to host-resident font character sets (continued)

Registered font ID	Font width	Font attributes	Map fidelity	Font character set name (first choice)	Font character set name (second choice)
214	96	Bold	Exact	C0S0CB15	
215	96	Italic	Exact	C0S0CI15	
217	96	Double wide	Exact	C0S0CD15	
218	96	Double wide italic	Exact	C0S0CW15	
221	96	Normal	Not exact	C0S0CR15	
221	96	Bold	Not exact	C0S0CB15	
221	96	Normal	Exact	C0E0PR15	C0S0CR15
221	96	Normal	Exact	C0S0SYM0	
222	96	Normal	Exact	C0D0GT15	
222	96	Bold	Not exact	C0S0CB15	
222	96	Normal	Exact	C0E0LR15	C0S0LR15
222	96	Normal	Exact	C0E0LR5R	C0S0SYM2
223	96	Normal	Exact	C0S0CR15	
223	96	Double wide	Exact	C0S0CD15	
223	96	Bold	Exact	C0S0CB15	
223	96	Italic	Exact	C0S0CI15	
223	96	Double wide italic	Exact	C0S0CW15	
223	96	Normal	Exact	C0E0CR15	C0S0CR15
223	96	Normal	Exact	C0S0SYM0	
223	96	Normal	Exact	C0E0CR5K	C0420380
223	96	Normal	Exact	C0E0CR5N	C0420380
223	96	Normal	Exact	C0E0CR5G	C0420280
223	96	Normal	Exact	C0E0CR5H	C0420280
225	96	Normal	Not exact	C0S0CR15	
225	96	Bold	Not exact	C0S0CB15	
225	96	Normal	Exact	C0S0SYM0	
226	96	Normal	Exact	C0E0HR15	C0420580
229	96	Normal	Exact	C0D0ST15	C0S0CR15
229	96	Normal	Exact	C0S0SYM0	
230	96	Normal	Exact	C0D0GT15	C0S0CR15
230	96	Normal	Exact	C0S0SYM0	
232	96	Normal	Not exact	C0S0CR15	
232	96	Bold	Not exact	C0S0CB15	
233	96	Normal	Exact	C0S0CD15	
244	288	Normal	Exact	C0S0SYM0	
245	144	Normal	Not exact	C0S0CR10	
245	144	Bold	Not exact	C0S0CB10	

Table 16. Mapping printer-resident to host-resident font character sets (continued)

Registered font ID	Font width	Font attributes	Map fidelity	Font character set name (first choice)	Font character set name (second choice)
245	288	Bold	Exact	C0S0SYM2	
248	84	Normal	Exact	C0420580	
249	84	Normal	Exact	C0E0KA17	C0L0KN20
252	78	Normal	Not exact	C0D0GT18	
252	84	Normal	Exact	C0D0GT18	
252	84	Normal	Exact	C0S0SYM0	
253	84	Bold	Not exact	C0D0GT18	
253	84	Bold	Exact	C0S0SYM0	
254	78	Normal	Not exact	C0D0GT18	
254	84	Normal	Exact	C0E0CR7F	C0D0GT18
254	84	Normal	Exact	C0E0CR7G	C0420270
254	84	Normal	Exact	C0E0CR7H	C0420270
254	84	Normal	Exact	C0E0CR17	C0D0GT18
254	84	Normal	Exact	C0S0SYM0	
255	84	Normal	Exact	C0E0LR17	C0D0GT18
255	84	Normal	Exact	C0E0LR7R	C0S0SYM2
256	84	Normal	Exact	C0E0PR17	C0D0GT18
256	84	Normal	Exact	C0E0PR7R	C0S0SYM2
258	78	Normal	Exact	C0D0GT18	
259	78	Normal	Not exact	C0D0GT18	
266	177	Bold	Exact	C0E0NB08	C04404D0
267	177	Bold italic	Exact	C0E0NM08	C04504D0
275	78	Normal	Exact	C0D0GT18	
279	84	Bold	Exact	C0E0NR17	C0440470
280	72	Normal	Exact	C0S0AE20	
281	72	Normal	Exact	C0D0GT20	
281	72	Normal	Exact	C0E0LR20	C0D0GT20
281	72	Normal	Exact	C0S0SYM0	
282	72	Normal	Exact	C0E0LV20	C0420570
283	72	Bold	Exact	C0E0GN20	C0440470
285	58	Normal	Exact	C0E0LR25	C0D0GT20
290	54	Normal	Exact	C0D0GT24	
290	53	Normal	Not exact	C0D0GT24	
290	53	Normal	Exact	C0S0SYM0	
300	54	Normal	Exact	C0D0GT18	
304*	54	Normal	Exact	C0620050	C0D0GT20
304*	72	Normal	Exact	C0620060	C0D0GT20
304*	84	Normal	Exact	C0620070	C0D0GT18

Table 16. Mapping printer-resident to host-resident font character sets (continued)

Registered font ID	Font width	Font attributes	Map fidelity	Font character set name (first choice)	Font character set name (second choice)
304*	96	Normal	Exact	C0620080	C0D0GT15
304*	108	Normal	Exact	C0620090	C0D0GT13
304*	120	Normal	Exact	C0620000	C0D0GT12
304*	144	Normal	Exact	C06200B0	C0D0GT10
304*	168	Normal	Exact	C06200D0	C0D0GT10
304*	240	Normal	Exact	C06200N0	C0D0GT10
304*	72	Normal	Exact	C0620860	C0L0KN20
304*	84	Normal	Exact	C0620870	C0L0KN20
304*	96	Normal	Exact	C0620880	C0L0KN20
304*	120	Normal	Exact	C06208B0	C0L0KATA
304*	168	Normal	Exact	C06208D0	C0L0KATA
304*	240	Normal	Exact	C06208J0	C0L0KATA
305*	144	Normal	Exact	C0920AB0	C0L00AOA
306*	144	Normal	Exact	C0920BB0	C0L00BOA
307*	144	Normal	Exact	C0420P00	C050AE10
318*		Bold	Exact	C07400XX	C050CBXX
319*		Italic	Exact	C07300XX	C050CIXX
322*	144	Normal	Exact	C0440P00	C050AE10
323*	120	Normal	Exact	C0BPOSA0	
323*	144	Normal	Exact	C0BPOS91	
323*	168	Normal	Exact	C0BPOSB0	
323*	240	Normal	Exact	C0BPOSBN	
326	96	Bold	Exact	C0T40680	
326	120	Bold	Exact	C0T40600	
326	144	Bold	Exact	C0T406B0	
326	180	Bold	Exact	C0T406E0	
327	96	Italic	Exact	C0T30680	
327	120	Italic	Exact	C0T30600	
327	144	Italic	Exact	C0T306B0	
327	180	Italic	Exact	C0T306E0	
328	96	Normal	Exact	C0T20680	
328	120	Normal	Exact	C0T20600	
328	144	Normal	Exact	C0T206B0	
328	180	Normal	Exact	C0T206E0	
335		Normal	Exact	C0B200XX	
335*		Normal	Exact	C0B20CXX	
336		Bold	Exact	C0B400XX	
336*		Bold	Exact	C0B40CXX	

Table 16. Mapping printer-resident to host-resident font character sets (continued)

Registered font ID	Font width	Font attributes	Map fidelity	Font character set name (first choice)	Font character set name (second choice)
337		Italic	Exact	C0B300XX	
337*		Italic	Exact	C0B30CXX	
338		Bold italic	Exact	C0B500XX	
338*		Bold italic	Exact	C0B50CXX	
339		Reverse	Exact	C0B600XX	
339*		Reverse	Exact	C0B60CXX	
400	80	Normal	Exact	C0D0GT18	
400*		Normal	Exact	C05200XX	C0D0GTXX
404*		Normal	Exact	C05400XX	C0D0GTXX
416		Normal	Exact	C04200XX	C0S0CRXX
416	115	Normal	Exact	C0420000	C0S0CR15
416		Normal	Exact	C04202XX	
416		Normal	Exact	C04203XX	
416		Normal	Exact	C04204XX	
416		Normal	Exact	C04205XX	
416*		Normal	Exact	C04201XX	
416*		Normal	Exact	C04207XX	
417	96	Double wide	Exact	C0S0CD15	
420		Bold	Exact	C04400XX	C0S0CBXX
420	115	Normal	Exact	C0440000	C0S0CB15
420		Bold	Exact	C04402XX	
420		Bold	Exact	C04403XX	
420		Bold	Exact	C04404XX	
420		Bold	Exact	C04405XX	
420*		Bold	Exact	C04401XX	
420*		Bold	Exact	C04407XX	
424		Italic	Exact	C04300XX	C0S0CIXX
424	115	Italic	Exact	C0430000	C0S0CI15
424		Italic	Exact	C04302XX	
424		Italic	Exact	C04303XX	
424		Italic	Exact	C04304XX	
424		Italic	Exact	C04305XX	
424*		Italic	Exact	C04307XX	
425	96	Double wide italic	Exact	C0S0CW15	
428		Bold italic	Exact	C04500XX	C0S0CIXX
428	115	Bold italic	Exact	C0450000	C0S0CI15
428		Bold italic	Exact	C04502XX	
428		Bold italic	Exact	C04503XX	



Table 16. Mapping printer-resident to host-resident font character sets (continued)

Registered font ID	Font width	Font attributes	Map fidelity	Font character set name (first choice)	Font character set name (second choice)
428		Bold italic	Exact	C04504XX	
428		Bold italic	Exact	C04505XX	
428*		Bold italic	Exact	C04507XX	
432*		Normal	Exact	C07200XX	C0S0CRXX
434	177	Bold	Exact	C0E0OB08	C0S0CB10
435	221	Bold	Exact	C0E0OB06	C0S0CB10
751	53	Normal	Exact	C0T05580	C0D0GT24
751	53	Normal	Exact	C0S0SYM0	
752	80	Normal	Exact	C0E20NB0	C0N204B0
753	120	Bold	Exact	C0E0BNTR	C0N404B0
753	80	Bold	Exact	C0E40NB0	C0N404B0
754	120	Bold	Exact	C0E40NH0	C0N404H0
755	160	Bold	Exact	C0E40NN0	C0N404N0
756	80	Italic	Exact	C0E30NB0	C0N304B0
757	80	Bold italic	Exact	C0E50NB0	C0N504B0
758	120	Bold italic	Exact	C0E50NH0	C0N504H0
759	160	Bold italic	Exact	C0E50NN0	C0N504N0
1051	67	Normal	Exact	C0T05500	C0D0GT20
1051	67	Normal	Exact	C0S0SYM0	
1053	67	Bold	Exact	C0T07500	C0D0GT20
1053	67	Bold	Exact	C0S0SYM0	
1056	67	Italic	Exact	C0T15500	C0D0GT20
1056	67	Italic	Exact	C0S0SYM0	
1351	80	Normal	Exact	C0T055B0	C0D0GT18
1351	80	Normal	Exact	C0S0SYM2	
1653	107	Bold	Exact	C0T075F0	C0S0CB15
1653	107	Bold	Exact	C0S0SYM2	
1803	120	Bold	Exact	C0T075H0	C0S0CB10
2103	160	Bold	Exact	C0T075N0	C0S0CB10
2304		Normal	Exact	C0H200XX	C0S0CRXX
2304	96	Normal	Exact	C0H20080	C0D0GT24
2304	115	Normal	Exact	C0H200H0	C0S0CR12
2304	144	Normal	Exact	C0H200B0	C0D0GT18
2304	169	Normal	Exact	C0H200D0	C0S0CR15
2304	221	Normal	Exact	C0H200J0	C0S0CR10
2304	288	Normal	Exact	C0H200Z0	C0S0CR10
2304		Normal	Exact	C0H202XX	
2304		Normal	Exact	C0H203XX	

Table 16. Mapping printer-resident to host-resident font character sets (continued)

Registered font ID	Font width	Font attributes	Map fidelity	Font character set name (first choice)	Font character set name (second choice)
2304		Normal	Exact	C0H204XX	
2304		Normal	Exact	C0H205XX	
2304*		Normal	Exact	C0H201XX	
2304*		Normal	Exact	C0H207XX	
2305		Bold	Exact	C0H400XX	C0S0CBXX
2305	96	Bold	Exact	C0H40080	C0D0GT24
2305	115	Bold	Exact	C0H400H0	C0S0CB12
2305	144	Bold	Exact	C0H400B0	C0D0GT18
2305	169	Bold	Exact	C0H400D0	C0S0CB15
2305	221	Bold	Exact	C0H400J0	C0S0CB10
2305	288	Bold	Exact	C0H400Z0	C0S0CB10
2305		Bold	Exact	C0H402XX	
2305		Bold	Exact	C0H403XX	
2305		Bold	Exact	C0H404XX	
2305		Bold	Exact	C0H405XX	
2305*		Bold	Exact	C0H401XX	
2305*		Bold	Exact	C0H407XX	
2306		Italic	Exact	C0H300XX	C0S0CIXX
2306	96	Italic	Exact	C0H30080	C0D0GT24
2306	115	Italic	Exact	C0H300H0	C0S0CI12
2306	144	Italic	Exact	C0H300B0	C0D0GT18
2306	169	Italic	Exact	C0H300D0	C0S0CI15
2306	221	Italic	Exact	C0H300J0	C0S0CI10
2306	288	Italic	Exact	C0H300Z0	C0S0CI10
2306		Italic	Exact	C0H302XX	
2306		Italic	Exact	C0H303XX	
2306		Italic	Exact	C0H304XX	
2306		Italic	Exact	C0H305XX	
2306*		Italic	Exact	C0H307XX	
2307		Bold italic	Exact	C0H500XX	C0S0CIXX
2307	96	Bold italic	Exact	C0H50080	C0D0GT24
2307	115	Bold italic	Exact	C0H500H0	C0S0CI12
2307	144	Bold italic	Exact	C0H500B0	C0D0GT18
2307	169	Bold italic	Exact	C0H500D0	C0S0CI15
2307	221	Bold italic	Exact	C0H500J0	C0S0CI10
2307	288	Bold italic	Exact	C0H500Z0	C0S0CI10
2307		Bold italic	Exact	C0H502XX	
2307		Bold italic	Exact	C0H503XX	

Table 16. Mapping printer-resident to host-resident font character sets (continued)

Registered font ID	Font width	Font attributes	Map fidelity	Font character set name (first choice)	Font character set name (second choice)
2307		Bold italic	Exact	C0H504XX	
2307		Bold italic	Exact	C0H505XX	
2307*		Bold italic	Exact	C0H507XX	
2308		Normal	Exact	C0N200XX	C0S0CRXX
2308	96	Normal	Exact	C0N20080	C0D0GT24
2308	115	Normal	Exact	C0N200H0	C0S0CR12
2308	144	Normal	Exact	C0N200B0	C0D0GT18
2308	169	Normal	Exact	C0N200D0	C0S0CR15
2308	221	Normal	Exact	C0N200J0	C0S0CR10
2308	288	Normal	Exact	C0N200Z0	C0S0CR10
2308		Normal	Exact	C0N202XX	
2308		Normal	Exact	C0N203XX	
2308		Normal	Exact	C0N204XX	
2308		Normal	Exact	C0N205XX	
2308*		Normal	Exact	C0N201XX	
2308*		Normal	Exact	C0N207XX	
2309		Bold	Exact	C0N400XX	C0S0CBXX
2309	96	Bold	Exact	C0N40080	C0D0GT24
2309	115	Bold	Exact	C0N400H0	C0S0CB12
2309	144	Bold	Exact	C0N400B0	C0D0GT18
2309	169	Bold	Exact	C0N400D0	C0S0CB15
2309	221	Bold	Exact	C0N400J0	C0S0CB10
2309	288	Bold	Exact	C0N400Z0	C0S0CB10
2309		Bold	Exact	C0N402XX	
2309		Bold	Exact	C0N403XX	
2309		Bold	Exact	C0N404XX	
2309		Bold	Exact	C0N405XX	
2309*		Bold	Exact	C0N401XX	
2309*		Bold	Exact	C0N407XX	
2310		Italic	Exact	C0N300XX	C0S0CIXX
2310	96	Italic	Exact	C0N30080	C0D0GT24
2310	115	Italic	Exact	C0N300H0	C0S0CI12
2310	144	Italic	Exact	C0N300B0	C0D0GT18
2310	169	Italic	Exact	C0N300D0	C0S0CI15
2310	221	Italic	Exact	C0N300J0	C0S0CI10
2310	288	Italic	Exact	C0N300Z0	C0S0CI10
2310		Italic	Exact	C0N302XX	
2310		Italic	Exact	C0N303XX	

Table 16. Mapping printer-resident to host-resident font character sets (continued)

Registered font ID	Font width	Font attributes	Map fidelity	Font character set name (first choice)	Font character set name (second choice)
2310		Italic	Exact	C0N304XX	
2310		Italic	Exact	C0N305XX	
2310*		Italic	Exact	C0N307XX	
2311		Bold italic	Exact	C0N500XX	C0S0CIXX
2311	96	Bold italic	Exact	C0N50080	C0D0GT24
2311	115	Bold italic	Exact	C0N500H0	C0S0CI12
2311	144	Bold italic	Exact	C0N500B0	C0D0GT18
2311	169	Bold italic	Exact	C0N500D0	C0S0CI15
2311	221	Bold italic	Exact	C0N500J0	C0S0CI10
2311	288	Bold italic	Exact	C0N500Z0	C0S0CI10
2311		Bold italic	Exact	C0N502XX	
2311		Bold italic	Exact	C0N503XX	
2311		Bold italic	Exact	C0N504XX	
2311		Bold italic	Exact	C0N505XX	
2311		Bold italic	Exact	C0N507XX	
4407		Normal	Exact	C0T055XX	C0S0CRXX
4407	42	Normal	Exact	C0T05560	C0D0GT24
4407	54	Normal	Exact	C0T05580	C0D0GT24
4407	66	Normal	Exact	C0T05500	C0D0GT20
4407	72	Normal	Exact	C0T055A0	C0D0GT20
4407	78	Normal	Exact	C0T055B0	C0D0GT18
4427		Bold	Exact	C0T075XX	C0S0CBXX
4427	66	Bold	Exact	C0T07500	C0D0GT20
4427	96	Bold	Exact	C0T075D0	C0S0CB15
4427	108	Bold	Exact	C0T075F0	C0S0CB15
4427	132	Bold	Exact	C0T075J0	C0S0CB10
4427	162	Bold	Exact	C0T075N0	C0S0CB10
4535		Italic	Exact	C0T155XX	C0S0CIXX
4535	66	Italic	Exact	C0T15500	C0D0GT20
4535	72	Italic	Exact	C0T155A0	C0D0GT20
4555		Bold italic	Exact	C0T175XX	C0S0CIXX
4555	66	Bold italic	Exact	C0T17500	C0D0GT20
4555	78	Bold italic	Exact	C0T175B0	C0D0GT18
4555	132	Bold italic	Exact	C0T175J0	C0S0CI10
4919	40	Normal	Exact	C0E20G60	C0D0GT18
4919	53	Normal	Exact	C0E20G80	C0S0CR15
4919	67	Normal	Exact	C0E20G00	C0S0CR12
4919	80	Normal	Exact	C0E20GB0	C0S0CR10

Table 16. Mapping printer-resident to host-resident font character sets (continued)

Registered font ID	Font width	Font attributes	Map fidelity	Font character set name (first choice)	Font character set name (second choice)
4939	67	Bold	Exact	C0E40G00	C0S0CB12
4939	93	Bold	Exact	C0E40GD0	C0S0CB10
4939	120	Bold	Exact	C0E40GH0	C0S0CB10
5047	67	Italic	Exact	C0E30G00	C0S0CI12
5067	67	Bold italic	Exact	C0E50G00	C0S0CI12
5687	80	Normal	Exact	C0E20TB0	C0S0CR10
5687	67	Normal	Exact	C0E20T00	C0S0CR12
5687	53	Normal	Exact	C0E20T80	C0S0CR15
5687	40	Normal	Exact	C0E20T60	C0D0GT18
5707	160	Bold	Exact	C0E40TN0	C0S0CB10
5707	120	Bold	Exact	C0E40TH0	C0S0CB10
5707	93	Bold	Exact	C0E40TD0	C0S0CB10
5707	80	Bold	Exact	C0E40TB0	C0S0CB10
5707	67	Bold	Exact	C0E40T00	C0S0CB12
5815	80	Italic	Exact	C0E30TB0	C0S0CI10
5815	67	Italic	Exact	C0E30T00	C0S0CI12
5835	80	Bold italic	Exact	C0E50TB0	C0S0CI10
5835	67	Bold italic	Exact	C0E50T00	C0S0CI12
5943	120	Normal	Exact	C0E20MH0	C0S0CR10
5943	93	Normal	Exact	C0E20MD0	C0S0CR10
5943	80	Normal	Exact	C0E20MB0	C0S0CR10
6199	80	Normal	Exact	C0E20PB0	C0S0CR10
6199	67	Normal	Exact	C0E20P00	C0S0CR12
6199	53	Normal	Exact	C0E20P80	C0S0CR15
6199	40	Normal	Exact	C0E20P60	C0D0GT18
6219	120	Bold	Exact	C0E40PH0	C0S0CB10
6219	93	Bold	Exact	C0E40PD0	C0S0CB12
6219	67	Bold	Exact	C0E40P00	C0S0CB15
6327	67	Italic	Exact	C0E30P00	C0S0CI12
6347	67	Bold italic	Exact	C0E50P00	C0S0CI12
8503	80	Normal	Exact	C0E20BB0	C0S0CR10
8503	67	Normal	Exact	C0E20B00	C0S0CR10
8503	53	Normal	Exact	C0E20B80	C0S0CR15
8503	40	Normal	Exact	C0E20B60	C0D0GT18
8523	120	Bold	Exact	C0E40BH0	C0S0CB10
8523	93	Bold	Exact	C0E40BD0	C0S0CB10
8523	67	Bold	Exact	C0E40B00	C0S0CB12
8631	67	Italic	Exact	C0E30B00	C0S0CI12

Table 16. Mapping printer-resident to host-resident font character sets (continued)

Registered font ID	Font width	Font attributes	Map fidelity	Font character set name (first choice)	Font character set name (second choice)
8651	67	Bold italic	Exact	C0E50B00	C0S0CI12
12855	80	Normal	Exact	C0E20KB0	C0S0CR10
12855	67	Normal	Exact	C0E20K00	C0S0CR12
12855	53	Normal	Exact	C0E20K80	C0S0CR15
12875	160	Bold	Exact	C0E40KN0	C0S0CB10
12875	120	Bold	Exact	C0E40KH0	C0S0CB10
12875	67	Bold	Exact	C0E40K00	C0S0CB12
12875	53	Bold	Exact	C0E40K80	C0S0CB15
12875	80	Bold	Exact	C0E40KB0	C0S0CB10
16951	80	Normal	Exact	C0E20CB0	C0S0CR10
16951	67	Normal	Exact	C0E20C00	C0S0CR10
16951	53	Normal	Exact	C0E20C80	C0S0CR15
16951	40	Normal	Exact	C0E20C60	C0D0GT18
16971	120	Bold	Exact	C0E40CH0	C0S0CB10
16971	93	Bold	Exact	C0E40CD0	C0S0CB10
16971	67	Bold	Exact	C0E40C00	C0S0CB12
17079	67	Italic	Exact	C0E30C00	C0S0CI12
17099	67	Bold italic	Exact	C0E50C00	C0S0CI12
33079		Normal	Exact	C0A055XX	C0S0CRXX
33099		Bold	Exact	C0A075XX	C0S0CBXX
33207		Italic	Exact	C0A155XX	C0S0CIXX
33227		Bold italic	Exact	C0A175XX	C0S0CIXX
33335	80	Normal	Exact	C0E20OB0	C0S0CR10
33335	67	Normal	Exact	C0E20O00	C0S0CR12
33335	53	Normal	Exact	C0E20O80	C0S0CR15
33335	40	Normal	Exact	C0E20O60	C0D0GT18
33355	120	Bold	Exact	C0E40OH0	C0S0CB10
33355	93	Bold	Exact	C0E40OD0	C0S0CB10
33355	67	Bold	Exact	C0E40O00	C0S0CB12
33463	67	Italic	Exact	C0E30O00	C0S0CI10
33483	67	Bold italic	Exact	C0E50O00	C0S0CI12
33591	80	Normal	Exact	C0E20FB0	C0S0CR10
33591	67	Normal	Exact	C0E20F00	C0S0CR12
33591	53	Normal	Exact	C0E20F80	C0S0CR15
33591	40	Normal	Exact	C0E20F60	C0D0GT18
33601	120	Bold	Exact	C0E40FH0	C0S0CB10
33601	93	Bold	Exact	C0E40FD0	C0S0CB10
33601	67	Bold	Exact	C0E40F00	C0S0CB12

Table 16. Mapping printer-resident to host-resident font character sets (continued)

Registered font ID	Font width	Font attributes	Map fidelity	Font character set name (first choice)	Font character set name (second choice)
33719	67	Italic	Exact	C0E30F00	C0S0CI12
33729	67	Bold italic	Exact	C0E50F00	C0S0CI12
34103	80	Normal	Exact	C0E20HB0	C0S0CR10
34103	67	Normal	Exact	C0E20H00	C0S0CR10
34103	53	Normal	Exact	C0E20H80	C0S0CR15
34103	40	Normal	Exact	C0E20H60	C0D0GT18
34123	120	Bold	Exact	C0E40HH0	C0S0CB10
34123	93	Bold	Exact	C0E40HD0	C0S0CB10
34123	67	Bold	Exact	C0E40H00	C0S0CB12
34231	67	Italic	Exact	C0E30H00	C0S0CI12
34251	67	Bold italic	Exact	C0E50H00	C0S0CI10
37431	120	Normal	Exact	C0E20EH0	C0S0CR10
37431	93	Normal	Exact	C0E20ED0	C0S0CR10
37431	80	Normal	Exact	C0E20EB0	C0S0CR10
41783	80	Italic	Exact	C0E30SB0	C0S0CI10
41803	120	Bold italic	Exact	C0E50SH0	C0S0CI10
41803	93	Bold italic	Exact	C0E50SD0	C0S0CI10
49719	54	Normal	Exact	C0P05580	C0D0GT18

## Mapping printer-resident to host-resident code pages

The table in this topic can help you determine what host resident code page will be downloaded to a 3820, 3825, 3827, 3829, 3831, 3835, or 3900 printer when your spooled file refers to a registered code page identifier (ID) instead of a host-resident code page.

This font substitution is necessary because these printers do not support printer resident fonts. Depending on the registered code page ID value that is requested for a particular font reference, the appropriate host-resident code page is selected to match (as closely as possible) your font request.

The first choice is used if it is present on your system. The second choice is used if the first choice cannot be found.

The Map fidelity column indicates whether the first choice is considered to be an exact match to the printer-resident font that is requested in your spooled file. As a rule, the second choice is not considered to be an exact match.

Table 17. Mapping printer-resident to host-resident code pages

Registered code page ID	Host-resident code page name (first choice)	Host-resident code page name (second choice)	Map fidelity
29	T1V10871		Exact
37	T1V10037		Exact
38	T1V10500		Exact
256	T1GDP256		Exact
259	T1000259		Exact

Table 17. Mapping printer-resident to host-resident code pages (continued)

Registered code page ID	Host-resident code page name (first choice)	Host-resident code page name (second choice)	Map fidelity
260	T1V10037		Exact
273	T1V10273		Exact
274	T1V10274		Exact
275	T1V10275		Exact
277	T1V10277		Exact
278	T1V10278		Exact
280	T1V10280		Exact
281	T1V10281		Exact
282	T1V10282		Exact
283	T1V10284		Exact
284	T1V10284		Exact
285	T1V10285		Exact
286	T1V10273		Exact
287	T1V10277		Exact
288	T1V10278		Exact
289	T1V10284		Exact
290	T1V10290		Exact
293	T1000293	T1S0AE10	Exact
297	T1V10297		Exact
310	T1000310	T1S0AE10	Exact
340	T1L0OCR1	T1V10500	Not exact
361	T1000361	T1GI0361	Exact
363	T1GPI363		Exact
382	T1000382	T1GI0382	Exact
383	T1000383	T1GI0383	Exact
384	T1000384	T1GI0384	Exact
385	T1000385	T1GI0385	Exact
386	T1000386	T1GI0386	Exact
387	T1000387	T1GI0387	Exact
388	T1000388	T1GI0388	Exact
389	T1000389	T1GI0389	Exact
390	T1000390	T1GI0390	Exact
391	T1000391	T1GI0391	Exact
392	T1000392	T1GI0392	Exact
393	T1000393	T1GI0393	Exact
394	T1000394	T1GI0394	Exact
395	T1000395	T1GI0395	Exact
396	T1GI0396		Exact
420	T1000420	T1V10500	Not exact



Table 17. Mapping printer-resident to host-resident code pages (continued)

Registered code page ID	Host-resident code page name (first choice)	Host-resident code page name (second choice)	Map fidelity
423	T1000423		Exact
424	T1000424	T1V10500	Not exact
437	T1000437	T1V10500	Not exact
500	T1V10500		Exact
803	T1000803		Exact
813	T1000813		Exact
819	T1000819		Exact
829	T1M00829		Exact
831	T1V10282		Exact
838	T1000838		Exact
850	T1000850		Exact
851	T1000851		Exact
852	T1000852		Exact
853	T1000853		Exact
855	T1000855		Exact
856	T1000856		Exact
857	T1000857		Exact
860	T1000860		Exact
861	T1000861		Exact
862	T1000862		Exact
863	T1000863		Exact
864	T1000864		Exact
865	T1000865		Exact
866	T1000866		Exact
869	T1000869		Exact
870	T1000870	T1V10500	Not exact
871	T1V10871		Exact
874	T1V10874		Exact
875	T1000875		Exact
880	T1000880		Exact
890	T1000890	T1V10500	Not exact
892	T1L0OCR1	T1V10500	Not exact
893	T1L0OCRB	T1V10500	Not exact
897	T1000897		Exact
899	T1000899		Exact
905	T1000905		Exact
912	T1000912		Exact
914	T1000914		Exact
915	T1000915		Exact

Table 17. Mapping printer-resident to host-resident code pages (continued)

Registered code page ID	Host-resident code page name (first choice)	Host-resident code page name (second choice)	Map fidelity
916	T1000916		Exact
920	T1000920		Exact
1002	T1001002	T1D0BASE	Exact
1003	T1DCDCFS		Exact
1004	T1001004		Exact
1008	T1001008		Exact
1025	T1001025		Exact
1026	T1001026		Exact
1027	T1001027		Exact
1028	T1001028		Exact
1029	T1001029		Exact
1038	T1001038		Exact
1039	T1001039		Exact
1041	T1001041		Exact
1046	T1001046		Exact
1068	T1001068		Exact
1069	T1001069		Exact
1070	T1GDP037		Exact
1071	T1GDP273		Exact
1072	T1GDP274		Exact
1073	T1GDP275		Exact
1074	T1GDP277		Exact
1075	T1GDP278		Exact
1076	T1GDP280		Exact
1077	T1GDP281		Exact
1078	T1GDP282		Exact
1079	T1GDP284		Exact
1080	T1GDP285		Exact
1081	T1GDP279		Exact
1087	T1001087		Exact
1091	T1001091		Exact
1092	T1001092		Exact
2063	T1D0BASE		Exact
2064	T1GDP276		Exact
2065	T1GI0361		Exact
2066	T1GPI363		Exact
2067	T1GI0382		Exact
2068	T1GI0383		Exact
2069	T1GI0384		Exact

Table 17. Mapping printer-resident to host-resident code pages (continued)

Registered code page ID	Host-resident code page name (first choice)	Host-resident code page name (second choice)	Map fidelity
2070	T1GI0385		Exact
2071	T1GI0386		Exact
2072	T1GI0387		Exact
2073	T1GI0388		Exact
2074	T1GI0389		Exact
2075	T1GI0390		Exact
2076	T1GI0391		Exact
2077	T1GI0392		Exact
2078	T1GI0394		Exact
2079	T1GI0395		Exact
2081	T1GE0200		Exact
2082	T1GE0300		Exact
2086	T1L0OCRB		Exact
2087	T1L0OCR1		Exact
2092	T1S0S193		Exact
2093	T1S0S198		Exact
2102	T1L02773		Exact
2103	T1L02774		Exact
2108	T1S0AE10		Exact

### CHRID values supported

The table in this topic lists all the character identifiers (CHRID), the related national language groups, the correct code page, and which printers support which character identifier.

	Code Pages		Printers <sup>1</sup>							
	CHRID Code Page xxx yyy <sup>2,3</sup>	Sub-stitute Code Page yyy <sup>2,4</sup>								
Language Groups			3812 <sup>5</sup> 3816 <sup>5</sup>	4214 <sup>5</sup>	4224 <sup>5</sup> 4230 <sup>5</sup> 4247 <sup>5</sup>	4234 <sup>5</sup> 6400 <sup>9</sup> 6408 <sup>9</sup> 6412 <sup>9</sup>	5219	5224 5225	3112 3116 3912 3916 4028 4312 4317 4324 Infoprint 20 Infoprint 32	3130 3160 3935 Infoprint 3000 Infoprint 4000
<i>Major groups</i>										
International (and U.S. ASCII)	103 038	500	Yes				Yes		Yes	Yes
Multinational	697 500		Yes	Yes	Yes	Yes			Yes	Yes
	337 256	500	Yes				Yes	Yes	Yes	Yes
	697 256	500	Yes		Yes	IPDS <sup>7</sup>			Yes	
United States	101 037		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	697 037		Yes	Yes		Yes			Yes	Yes
Australia, Brazil, Canada, Netherlands, New Zealand, Portugal, U.S. <sup>10</sup>	695 1140	697 037								

Language Groups	Code Pages		Printers <sup>1</sup>							
	CHRID Code Page xxx yyy <sup>2,3</sup>	Sub-stitute Code Page yyy <sup>2,4</sup>	3812 <sup>5</sup> 3816 <sup>5</sup>	4214 <sup>5</sup>	4224 <sup>5</sup> 4230 <sup>5</sup> 4247 <sup>5</sup>	4234 <sup>5</sup> 6400 <sup>9</sup> 6408 <sup>9</sup> 6412 <sup>9</sup>	5219	5224 5225	3112 3116 3912 3916 4028 4312 4317 4324 Infoprint 20 Infoprint 32	3130 3160 3935 Infoprint 3000 Infoprint 4000
<i>Individual regions or languages</i>										
Arabic	697 361		Yes		Yes				Yes	
Arabic X/B	235 420	500	Yes		Yes	IPDS <sup>7</sup>			Yes	Yes
	697 420		Yes		4224-No 4230-Yes 4247-Yes	IPDS <sup>7</sup>			Yes	
Arabic <sup>10</sup>	1461 420									
Austria/ Germany <sup>6</sup>	265 273		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	697 273		Yes	Yes	Yes	Yes			Yes	Yes
Austria/ Germany	697 286	273	Yes		Yes	IPDS <sup>7</sup>				Yes
	317 286		Yes		Yes	IPDS <sup>7</sup>				Yes
Austria, Germany <sup>10</sup>	695 1141	697 273								
Belgium <sup>6</sup>	697 500			Yes	Yes	Yes	Yes	Yes		Yes
	269 274			Yes	Yes	Yes	Yes	Yes		Yes
	697 274			Yes	Yes	Yes				Yes
Belgium, Canada, Switzerland <sup>10</sup>	695 1148	697 500								
Brazil <sup>6</sup>	273 275		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	697 275		Yes	Yes	Yes	Yes			Yes	Yes
Bulgaria, FYR Macedonia, Serbia (Cyrillic) <sup>10</sup>	1381 1154	1150 1025								
Canada-Bilingual	038 256		Yes				Yes		Yes	
	039 256		Yes				Yes		Yes	
Canada-English	037 256		Yes				Yes		Yes	
Canadian French <sup>6</sup>	277 276	297 037	Yes				Yes	Yes	Yes	Yes
	341 260		Yes		Yes	IPDS <sup>7</sup>			Yes	Yes
	697 260		Yes			IPDS <sup>7</sup>			Yes	
Chinese (Hong Kong S.A.R.)	119 256		Yes				Yes		Yes	Yes
Chinese-Simplified	1174 836									
Chinese- Traditional	1175 037									
Chinese-Traditional <sup>10</sup>	32000 1159	697 37								
Cyrillic	960 880				Yes	IPDS <sup>7</sup>			Yes	Yes
Cyrillic multilingual	1150 1025				4224-No 4230-Yes 4247-Yes	IPDS <sup>7</sup>				Yes
Czechoslovakia/ Czech	083 257						Yes			
Czechoslovakia/ Slovak	085 257						Yes			
Czech Republic, Hungary, Poland <sup>10</sup>	1375 1153	959 870								
Denmark/ Norway <sup>6</sup>	281 277		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	697 277		Yes	Yes	Yes	Yes			Yes	Yes

Language Groups	Code Pages		Printers <sup>1</sup>							
	CHRID Code Page xxx yyy <sup>2,3</sup>	Sub-stitute Code Page yyy <sup>2,4</sup>	3812 <sup>5</sup> 3816 <sup>5</sup>	4214 <sup>5</sup>	4224 <sup>5</sup> 4230 <sup>5</sup> 4247 <sup>5</sup>	4234 <sup>5</sup> 6400 <sup>9</sup> 6408 <sup>9</sup> 6412 <sup>9</sup>	5219	5224 5225	3112 3116 3912 3916 4028 4312 4317 4324 Infoprint 20 Infoprint 32	3130 3160 3935 Infoprint 3000 Infoprint 4000
Denmark/ Norway	697 287	277	Yes		Yes	IPDS <sup>7</sup>			Yes	
	321 287		Yes		Yes	IPDS <sup>7</sup>			Yes	Yes
Denmark, Norway <sup>10</sup>	695 1142	697 277								
Estonia	1307 1122				4224-No 4230-Yes 4247-Yes					
Estonia <sup>10</sup>	1391 1157	1307 1122								
Farsi	1219 1097				Yes	IPDS <sup>7</sup>				
Finland/ Sweden <sup>6</sup>	285 278		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	697 278		Yes	Yes	Yes	Yes			Yes	Yes
Finland/ Sweden	697 288	278	Yes		Yes	IPDS <sup>7</sup>			Yes	
	325 288		Yes		Yes	IPDS <sup>7</sup>			Yes	Yes
Finland, Sweden <sup>10</sup>	695 1143	697 278								
France (1977) <sup>6</sup>	289 279	297	Yes					Yes	Yes	
France (1980) <sup>6</sup>	288 297		Yes		Yes	Yes	Yes		Yes	Yes
	697 297		Yes		Yes	Yes			Yes	Yes
France	251 256		Yes				Yes		Yes	Yes
France <sup>10</sup>	695 1147	697 297								
France/ Belgium	031 256		Yes				Yes		Yes	
Germany/ Austria	028 256		Yes				Yes		Yes	
	029 256		Yes				Yes		Yes	
Greek	218 423				Yes	IPDS <sup>7</sup>			Yes	Yes
	925 875				Yes	IPDS <sup>7</sup>			Yes	Yes
Greek <sup>10</sup>	1371 875	218 423								
Hebrew	941 424		Yes		Yes	IPDS <sup>7</sup>			Yes	Yes
	697 424		Yes		4224-No 4230-Yes 4247-Yes	IPDS <sup>7</sup>			Yes	
	1147 803				4224-No 4230-Yes 4247-Yes	IPDS <sup>7</sup>				Yes
Hebrew <sup>10</sup>	1356 424									
Hungary	091 257						Yes			
Iceland <sup>10</sup>	695 1149	697 871								
Icelandic	697 871		Yes		Yes	IPDS <sup>7</sup>			Yes	Yes
	697 029		Yes						Yes	
Italy <sup>6</sup>	293 280		Yes	Yes	Yes	IPDS <sup>7</sup>	Yes	Yes	Yes	Yes
	697 280		Yes	Yes	Yes	Yes			Yes	Yes
Italy	041 256		Yes				Yes		Yes	
Italy <sup>10</sup>	695 1144	697 280								

Language Groups	Code Pages		Printers <sup>1</sup>							
	CHRID Code Page xxx yyy <sup>2,3</sup>	Sub-stitute Code Page yyy <sup>2,4</sup>	3812 <sup>5</sup> 3816 <sup>5</sup>	4214 <sup>5</sup>	4224 <sup>5</sup> 4230 <sup>5</sup> 4247 <sup>5</sup>	4234 <sup>5</sup> 6400 <sup>9</sup> 6408 <sup>9</sup> 6412 <sup>9</sup>	5219	5224 5225	3112 3116 3912 3916 4028 4312 4317 4324 Infoprint 20 Infoprint 32	3130 3160 3935 Infoprint 3000 Infoprint 4000
Japan-English <sup>6</sup>	297 281		Yes	Yes	Yes	IPDS <sup>7</sup>	Yes	Yes	Yes	Yes
	697 281		Yes	Yes	Yes	Yes			Yes	Yes
	068 256		Yes				Yes		Yes	
	069 256		Yes				Yes		Yes	
Japan- Katakana <sup>6</sup>	332 290		Yes		Yes	Yes		Yes	Yes	Yes
Japan- Katakana	1172 290									
Japan- Katakana <sup>10</sup>	1398 290									
Japan-Latin	1172 1027									
Japan-Latin <sup>10</sup>	1398 1027									
Korean	1173 833									
Korean	933 833				4230-Yes 4247-Yes 4224-	IPDS <sup>7</sup>				
	697 290		Yes			IPDS <sup>7</sup>			Yes	
Latin	959 870			Yes	IPDS <sup>7</sup>			Yes	Yes	
Latin America/ Puerto Rico	025 256		Yes				Yes	Yes		
Latvia/ Lithuania	1305 1112				4224-No 4230-Yes 4247-Yes					
Latvia/ Lithuania <sup>10</sup>	1393 1156	1305 1112								
Lao	1341 1132									
Netherlands	043 256		Yes				Yes		Yes	
Norway/ Denmark	055 256		Yes				Yes		Yes	
Poland	093 257						Yes			
Portugal <sup>6</sup>	301 282		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	697 282		Yes	Yes	Yes	Yes			Yes	Yes
Portugal	697 831	282	Yes		Yes				Yes	
	063 256		Yes				Yes		Yes	
Romania	087 258						Yes			
South Africa	081 258						Yes			
Spain <sup>6</sup>	305 283	284 284	Yes		Yes	Yes	Yes	Yes	Yes	Yes
	697 283		Yes		Yes				Yes	Yes
	697 289	284	Yes		Yes	IPDS <sup>7</sup>			Yes	
	329 289		Yes		Yes	IPDS <sup>7</sup>			Yes	Yes
	045 256		Yes				Yes		Yes	
Spain, Latin America (Spanish) <sup>10</sup>	695 1145	697 284								
Spanish Speaking <sup>6</sup>	309 284		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	697 284		Yes	Yes	Yes	Yes	Yes	Yes	Yes	
	1149 284						Yes			Yes

Language Groups	Code Pages		Printers <sup>1</sup>							
	CHRID Code Page xxx yyy <sup>2,3</sup>	Sub-stitute Code Page yyy <sup>2,4</sup>	3812 <sup>5</sup> 3816 <sup>5</sup>	4214 <sup>5</sup>	4224 <sup>5</sup> 4230 <sup>5</sup> 4247 <sup>5</sup>	4234 <sup>5</sup> 6400 <sup>9</sup> 6408 <sup>9</sup> 6412 <sup>9</sup>	5219	5224 5225	3112 3116 3912 3916 4028 4312 4317 4324 Infoprint 20 Infoprint 32	3130 3160 3935 Infoprint 3000 Infoprint 4000
Sweden/ Finland	052 256		Yes				Yes		Yes	Yes
	053 256		Yes				Yes			
Switzerland/ French	048 256		Yes				Yes		Yes	
Switzerland/ German	049 256		Yes				Yes		Yes	
Thai	1102 889				Yes	IPDS <sup>7</sup>				
	938 838				4230-Yes 4247-Yes 4224-	IPDS <sup>7</sup>				
Thai <sup>10</sup>	1395 1160	938 838								
Turkish	965 905				4230-Yes 4247-Yes 4224-Yes	IPDS <sup>7</sup>		Yes		Yes
	1152 1026				4230-Yes 4247-Yes 4224-	IPDS <sup>7</sup>				Yes
Turkish <sup>10</sup>	1378 1155	1152 1026								
Ukraine	1326 1123									
Ukraine <sup>10</sup>	1388 1158	1326 1123								
United Kingdom <sup>6</sup>	313 285		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	697 285		Yes	Yes	Yes	Yes		Yes		Yes
U.K./ Israel	066 256		Yes				Yes		Yes	
U.K./ Israel-Latin	067 256		Yes				Yes		Yes	
United Kingdom <sup>10</sup>	695 1146	697 285								
USA- Accounting	017 256		Yes				Yes		Yes	
USA/Australia	001 256		Yes				Yes		Yes	
Vietnamese	1336 1130									
Vietnamese <sup>10</sup>	1397 1164	1336 1130								
Countries of the former Yugoslavia	410 890				Yes	IPDS <sup>7</sup>				
Countries of the former Yugoslavia-Latin	095 257						Yes			
<i>Languages not related to country or region</i>										
APL	697 293		Yes			IPDS <sup>7</sup>			Yes	
	380 293		Yes		4224- 4230- 4247-	IPDS <sup>7</sup>			Yes	Yes

Language Groups	Code Pages		Printers <sup>1</sup>							
	CHRID Code Page xxx yyy <sup>2,3</sup>	Sub-stitute Code Page yyy <sup>2,4</sup>	3812 <sup>5</sup> 3816 <sup>5</sup>	4214 <sup>5</sup>	4224 <sup>5</sup> 4230 <sup>5</sup> 4247 <sup>5</sup>	4234 <sup>5</sup> 6400 <sup>9</sup> 6408 <sup>9</sup> 6412 <sup>9</sup>	5219	5224 5225	3112 3116 3912 3916 4028 4312 4317 4324 Infoprint 20 Infoprint 32	3130 3160 3935 Infoprint 3000 Infoprint 4000
APL Alternate	697 310		Yes		4224-No 4230-Yes 4247-Yes	IPDS <sup>7</sup>			Yes	Yes
	963 310		Yes		4224-No 4230-Yes 4247-Yes				Yes	Yes
ASCII	103 256		Yes				Yes		Yes	Yes
DCF Compatibility	1132 1002		Yes		4230-Yes 4247-Yes 4224-No	IPDS <sup>7</sup>			Yes	Yes
DCF US Text	1133 1003				4230- 4247- 4224-					Yes
DCF text with numeric space	1259 1068				4230- 4247- 4224-					Yes
EBCDIC	101 256		Yes				Yes		Yes	Yes
GML List Symbols	1258 1039									Yes
International Typographic	697 361		Yes						Yes	
OCR (unregistered)	697 340	500	Yes		Yes	IPDS <sup>7</sup>			Yes	
OCR A	697 892	500	Yes		Yes	IPDS <sup>7</sup>			Yes	
	968 892		Yes		Yes	IPDS <sup>7</sup>			Yes	Yes
OCR A (unregistered)	580 340	892	Yes		Yes	IPDS <sup>7</sup>			Yes	
OCR B	697 893	500	Yes		Yes	IPDS <sup>7</sup>			Yes	
	969 893		Yes		Yes	IPDS <sup>7</sup>			Yes	Yes
OCR B (unregistered)	590 340	893	Yes		Yes	IPDS <sup>7</sup>			Yes	
Personal Computer	697 437		Yes		4224-No 4247-Yes 4230-Yes				Yes	
Symbols	340 259		Yes						Yes	Yes
Symbol- Selectric	201 259	500	Yes				Yes		Yes	
Symbol-6640	202 259	500	Yes	Yes			Yes		Yes	
Symbol-6670	203 259		Yes				Yes		Yes	
Symbols, Adobe	1257 1087									Yes
Symbols Set 7	697 259		Yes						Yes	
Symbols Mod Set 7	1191 1091									Yes
Symbols Set 8	630 363									Yes



Language Groups	Code Pages		Printers <sup>1</sup>							
	CHRID Code Page xxx yyy <sup>2,3</sup>	Sub-stitute Code Page yyy <sup>2,4</sup>	3812 <sup>5</sup> 3816 <sup>5</sup>	4214 <sup>5</sup>	4224 <sup>5</sup> 4230 <sup>5</sup> 4247 <sup>5</sup>	4234 <sup>5</sup> 6400 <sup>9</sup> 6408 <sup>9</sup> 6412 <sup>9</sup>	5219	5224 5225	3112 3116 3912 3916 4028 4312 4317 4324 Infoprint 20 Infoprint 32	3130 3160 3935 Infoprint 3000 Infoprint 4000

**Notes:**

- <sup>1</sup> The 4245, 5256, and 5262 work station printers do not support the hardware function required for alternative CHRID processing. If a nondefault character set and code page is selected for these printers, a diagnostic message is sent and processing continues using the default character set.
- <sup>2</sup> If the printer supports the code page specified (the second part (yyy) of the CHRID parameter) but not the character set (xxx), then the character set supported by the printer is used along with the specified code page. For example, if 337 037 (extended character set for displays) is specified for the 5224 and 5225 Printers, the print file is printed with character set 101, code page 037.
- <sup>3</sup> In some cases, the printer will substitute a supported code page for an unsupported code page. Consult the various printer reference guides for defaults on the code page mapping.
- <sup>4</sup> If the printer does not support or map the code page specified, an attempt is made by the system to find a satisfactory substitute. This column shows the code page substitutes that are made if the specified printer supports the substitute.
- <sup>5</sup> The 3812, 3816, 4214, 4224, 4230, 4234, and 4247 Printers support character set 697 (full character set). This character set contains all the characters in the limited character sets. For example, 697 037 contains all the characters in 101 037 or 337 037 (extended character set for displays).
- <sup>6</sup> This language is considered a primary language group. All other entries, if any, under the primary language group are considered as alternative language groups.
- <sup>7</sup> This function is supported by 4234 IPDS version and 64xx with IPDS feature only.
- <sup>8</sup> This function is supported by 4234 SCS version only.
- <sup>9</sup> The 64xx SCS emulation mode must be set to 4234.
- <sup>10</sup> This code page provides support for the Euro currency symbol.

## LPI values supported

Lines per inch means the number of characters that can be printed vertically within an inch.

Each entry in the following table shows the valid range of values for lines per page for each printer type and for each value of lines per inch (LPI) valid for the printer.

**Note:** Because of slight adjustments made for position checks, it is recommended to not print on line 1 when specifying 8 or 9 LPI on an IPDS printer.

*Table 18. Lines per inch (LPI parameter)*

Printer	3 lines per inch	4 lines per inch	6 lines per inch	7.5 lines per inch	8 lines per inch	9 lines per inch	12 lines per inch
3287		1-104	1-104		1-104		
3812 IPDS		2-56	2-84		2-112	2-112	2-168
3812 SCS		1-56	1-84		1-112	1-126	1-168
3816 IPDS		2-56	2-84		2-112	2-112	2-168
3816 SCS		1-56	1-84		1-112	1-126	1-168
3820		1-56	1-84		1-112	1-126	1-168
3825		1-56	1-84		1-112	1-126	1-168
3827		1-56	1-84		1-112	1-126	1-168
3835		2-91	2-136		2-182	2-204	2-273

Table 18. Lines per inch (LPI parameter) (continued)

Printer	3 lines per inch	4 lines per inch	6 lines per inch	7.5 lines per inch	8 lines per inch	9 lines per inch	12 lines per inch
3935		1-68	1-102		1-136	1-153	1-204
4028		2-56	2-84		1-112	1-112 or 2-126	2-168
4214		1-255	1-255		1-255	1-255	
4224, 4234 IPDS		2-91	2-136		2-182	2-204	2-273
4230		2-91	2-136		2-182	2-204	2-273
4234 SCS		1-255	1-255		1-255		
4245 Models T12 and T20			1-255		1-255		
4247		2-91	2-136		2-182	2-204	2-273
5211			2-84		2-112		
5219 Continuous Forms		2-255	2-255		2-255		2-255
5219 Cut Sheet		57	86		114		172
5224		1-255	1-255		1-255	1-255	
5225		1-255	1-255		1-255	1-255	
5256 (set manually)			1-255		1-255		
5262			1-255		1-255		
5553	1-255	1-255	1-255	1-255	1-255		1-255
5583	1-255	1-255	1-255	1-255	1-255		
6252		1-255	1-255		1-255	1-255	

## CPI values supported

Characters per inch means the number of characters printed horizontally within an inch across a page. Each entry in the table in this topic shows the valid range of values for the characters per line for each printer type and for each value of characters per inch (CPI) for the printer.

Table 19. Characters per inch (CPI parameter)

Printer	5 characters per inch	10 characters per inch	12 characters per inch	13.3 characters per inch	15 characters per inch	16.7 characters per inch	18 characters per inch	20 characters per inch
3112 <sup>1</sup>	1-42	1-85	1-102		1-127			
3116 <sup>1</sup>	1-42	1-85	1-102		1-127			
3130 <sup>1</sup>		1-132	1-158		1-198			
3160 <sup>1</sup>		1-132	1-158		1-198			
3287		1-132						
3812 <sup>1</sup>	1-42	1-85	1-102		1-127			
3812 <sup>1</sup> Rotated Form	1-70	1-140	1-168		1-210			

Table 19. Characters per inch (CPI parameter) (continued)

Printer	5 characters per inch	10 characters per inch	12 characters per inch	13.3 characters per inch	15 characters per inch	16.7 characters per inch	18 characters per inch	20 characters per inch
3816 <sup>1</sup>	1-42	1-85	1-102		1-127			
3816 <sup>1</sup> Rotated Form	1-70	1-140	1-168		1-210			
3820 <sup>1</sup>		1-85	1-102		1-127			
3825 <sup>1</sup>		1-85	1-102		1-127			
3827 <sup>1</sup>		1-85	1-102		1-127			
3835 <sup>1</sup> , 3935 <sup>1</sup>		1-132	1-158		1-198			
3912 <sup>1</sup>	1-42	1-85	1-102		1-127			
3916 <sup>1</sup>	1-42	1-85	1-102		1-127			
4028 <sup>1</sup>	1-42	1-85	1-102		1-127			
4028 <sup>1</sup> Rotated Form	1-70	1-140	1-168		1-210			
4214 Continuous Forms	1-66	1-132	1-158		1-198	1-220		
4214 Cut Sheet	1-60	1-120	1-144		1-180	1-200		
4224 <sup>1</sup>		1-132	1-158		1-198	1-220		
4230 <sup>1</sup>		1-132	1-158		1-198	1-220		
4234 IPDS <sup>1</sup>	1-66	1-132	1-158		1-198	1-238		
4234 SCS <sup>1</sup>		1-132			1-198			
4245		1-132						
4247 <sup>1</sup>		1-132	1-158		1-198	1-220		
5219		1-132	1-158		1-198			
5224		1-132			1-198			
5225		1-132			1-198			
5256 Model 3		1-132						
5262		1-132						
5553		1-136	1-163	1-181	1-204		1-244	1-272
5583		1-132	1-158	1-176	1-198		1-236	1-264
6252		1-132			1-198			
6408 SCS <sup>2</sup>		1-132			1-198			
6408 IPDS <sup>3</sup>	1-66	1-132	1-158		1-198	1-238		

Table 19. Characters per inch (CPI parameter) (continued)

Printer	5 characters per inch	10 characters per inch	12 characters per inch	13.3 characters per inch	15 characters per inch	16.7 characters per inch	18 characters per inch	20 characters per inch
<b>Notes:</b>								
<sup>1</sup>	Many character per inch values (implied by the pitch of the font) are supported in addition to the ones listed here. For more information, see the FONT parameter. To find the maximum characters per line, multiply the implied characters per inch value listed in the font table by maximum page width supported (in inches). The maximum page width supported by the 3812 and 3816 Printers is 8.5 inches for nonrotated forms and 14.0 inches for rotated forms.							
<sup>2</sup>	This printer emulates 4234 SCS or 5225.							
<sup>3</sup>	This printer emulates 4234 IPDS.							

### 4019 printer information

The i5/OS operating system supports the 4019 printer by treating it as an emulated version of another device. In some respects, the result achieved with the 4019 printer is not identical to the emulated device.

The following table expresses capabilities in terms of the emulated printers, but indicates some situations in which the 4019 result exceeds that of the emulated device. See QWP4019 program for information about how to work with your emulated 4019 printer to make the 4019 resident fonts available.

Note that image, graphics, and bar codes are not supported under any emulation or means of attachment.

Pay special attention to the treatment of fonts and the footnotes relating to page length and width.

The following table lists the ways that you can attach the 4019 printer, the emulation method used, and the function provided by the combination of the attachment and emulation methods. In this table a supported function is indicated by an X.

Table 20. 4019 Printer-System Functions

Attached with	3477	3197	AWSC	WSF	WSE	E5250	R5250	OS/2 WSF
Emulating	5219	4214	3812	3812	5219	5219	5219	5219
<b>Printer file commands</b>								
Page length <sup>1</sup>	X	X	X	X	X	X	X	X
Page width <sup>1</sup>	X	X	X	X	X	X	X	X
LPI (4.0)	X	X	X	X	X	X	X	X
LPI (6.0)	X	X	X	X	X	X	X	X
LPI (8.0)	X	X	X	X	X	X	X	X
LPI (9.0) <sup>2</sup>	X	X	X	X				
FONT(*CPI) <sup>3</sup> CPI(5.0) <sup>4</sup>	X	X	X	X				
FONT(*CPI) <sup>3</sup> CPI(10.0)	X	X	X	X	X	X	X	X
FONT(*CPI) <sup>3</sup> CPI(12.0)	X	X	X	X	X	X	X	X
FONT(*CPI) <sup>3</sup> CPI(15.0)	X	X	X	X	X	X	X	X
FONT(*CPI) <sup>3</sup> CPI(16.7)		X	X	X	X	X	X	X

Table 20. 4019 Printer-System Functions (continued)

Attached with	3477	3197	AWSC	WSF	WSE	E5250	R5250	OS/2 WSF
Emulating	5219	4214	3812	3812	5219	5219	5219	5219
Fold records	X	X	X	X	X	X	X	X
Truncate records	X	X	X	X	X	X	X	X
Paper drawer (1)	X	X	X	X	X	X	X	X
Paper drawer (2)	X		X	X		X		
Paper drawer (E1)	X		X	X	X	X		
<b>Non-typographical fonts (See font table below for details.)</b>								
Typo and user-defined Fonts								
Form Feed (*CUT)		X		X	X		X	X
Form Feed (*AUTOCUT)	X	X	X	X	X	X	X	X
Print Quality (*Draft) with PAGRTT(*DEV D) automatically gives PAGRTT (*COR)			X	X				
Change character set/code page ID	X		X	X				
Rotation 0	X	X	X	X	X	X	X	X
Rotation 90								
Rotation 180								
Rotation 270	X		X	X				
Rotation *COR			X	X				
Print text	X	X	X	X	X	X	X	X
Hardware justification 0	X	X	X	X	X	X	X	X
Hardware justification 50	X			X	X	X	X	X
Hardware justification 100	X		X	X	X	X	X	X
Duplex								
Copies	X	X	X	X	X	X	X	X
File separators	X	X	X	X	X	X	X	X
<b>Additional DDS keywords</b>								
BARCODE								
CHRSIZ								
COLOR								
FONT (Changing Type Styles)								

Table 20. 4019 Printer-System Functions (continued)

Attached with	3477	3197	AWSC	WSF	WSE	E5250	R5250	OS/2 WSF
<b>Emulating</b>	<b>5219</b>	<b>4214</b>	<b>3812</b>	<b>3812</b>	<b>5219</b>	<b>5219</b>	<b>5219</b>	<b>5219</b>
HIGHLIGHT			X					
SKIPA	X	X	X	X	X	X	X	X
SKIPB	X	X	X	X	X	X	X	X
SPACEA	X	X	X	X	X	X	X	X
SPACEB	X	X	X	X	X	X	X	X
UNDERLINE	X	X	X	X	X	X	X	X
<b>Other functions</b>								
Graphics								
Image								
Symbols code page 259	X		X	X				

**Notes:**

<sup>1</sup> Existing applications or documents might not fit on the 4019 page since there is an unprintable border around the outside edge. You can change the margins and lines per page (and re-paginate in some cases) to obtain the output that you want.

This unprintable area applies to both envelopes and paper of any size. The unprintable area is 6.35 mm (0.25 in) from the sides and 4.23 mm (0.17 in) from the top and bottom. This results in an 8-inch writing line on 8.5 by 11 inch paper and a 7.7-inch writing line on A4 paper. With 6 LPI for example, this yields 64 lines on an 11-inch page or 68 lines on A4 paper.

Consideration should be given to the effect of this unprintable area when a document is formatted to assure that it prints correctly. If data is formatted to print beyond the printable area on the sides, the excess will print as an additional short line.

<sup>2</sup> LPI(9.0) is not supported by the 5219 printer; therefore, LPI(9.0) is not supported for any of the 5219 emulations.

<sup>3</sup> On the Create Printer File (CRTPRTF), Change Printer File (CHGPRTF), and Override with Printer File (OVRPRTF) commands you can avoid direct specification of a font by using FONT(\*CPI). This allows the system to default to any font that supports the requested CPI value. However, the defaulted font might be one that is not supported on the 4019. An unsupported font will cause printing to stop and require operator intervention. To avoid this, you can specify FONT explicitly on these commands.

<sup>4</sup> CPI(5.0) is not supported by the 5219; therefore, CPI(5.0) is not supported for any of the 5219 emulations.

**Related reference:**

“QWP4019 program” on page 391

QWP4019 is an IBM-supplied program that you can call to set flags on and off in a printer device description.

**4234 compressed font substitution by lines-per-inch value**

The table in this topic lists the font substitution that takes place when printing on a 4234 printer configured as follows.

- Value of \*NO for the AFP parameter
- Value greater than or equal to 8 for the lines per inch (LPI) parameter

This substitution allows the use of fonts that are slightly shorter when the LPI value is greater than or equal to 8.

Table 21. 4234 Compressed Font Substitution by Lines Per Inch (LPI) Value

Font Used When LPI is 4 or 6	Font Substituted When LPI is Greater Than or Equal to 8
11	52
26	51
85	75
87	74
160	154
204	205
222	232
223	233
258	259
400	300

## QWP4019 program

QWP4019 is an IBM-supplied program that you can call to set flags on and off in a printer device description.

By setting the flags on, you enable functions that are not accessible through the Create Device Description (Printer) (CRTDEVPRT) or Change Device Description (Printer) (CHGDEVPRT) commands. For example, the following command tells the system that PRT01 has a continuous form feed device:

```
CALL QWP4019 (PRT01 *CNT)
```

Since the flags are stored in the device description, QWP4019 only needs to be run once for each printer and each function. The flags can be changed only by running QWP4019 or deleting the device description. To make sure the flag is set, it is recommended that the printer writer for that printer be restarted after the QWP4019 program has been called.

The QWP4019 program allows you to take advantage of functions available on an attached printer but not supported by the emulator you are using.

**Note:** The QWP4019 program was designed to make the 4019 fonts available for the IBM LaserPrinter 4019 printer using emulation. Additional parameters can be specified to enable functions in the SCS printers. Most of these parameters are only valid for printers considered as a 5219 or 3812.

## QWP4019 parameter names and functions

The list in this topic contains the QWP4019 parameter names and explains the function provided when they are called.

### Parameter

#### Function provided

- \*ON This parameter sets a flag in the printer device description that:
  - Indicates to i5/OS that the 4019 fonts are to be used instead of the 5219 or 3812 fonts. See Printer font support to view the font mapping and substitution for the 4019 printer.
  - Enables manual paper feed selection if the value in the form feed (FORMFEED) parameter is \*CUT.
  - Prevents the value of \*COR in the page rotation (PAGRTT) parameter from being sent to a 3477 InfoWindow display that has an attached printer configured as a 5219. This is important because the 3477 does not support computer output reduction (COR). Without this flag, the PAGRTT(\*COR) value in the printer file cannot be used for these printers.

**\*OFF** This parameter sets the \*ON flag off. Also, if the \*SIC or \*COR flags were set on, using the \*OFF parameter sets them off.

**\*CHECK**

This parameter asks how the printer appears to the system (as a 3812, 4019, or 5219).

If the value 4019 is returned, it indicates that the QWP4019 program has been called with the \*ON parameter.

**\*CNT** This parameter sets a flag in the printer device description that tells the system that the printer has a continuous form feed device. The flag is used by the system to determine whether forms alignment is possible for printers that are configured as a 3812 printer. The 3812 printer does not support continuous forms.

This flag can only be set for printers that are configured as a 3812 printer.

**\*CNTOFF**

This parameter sets the \*CNT flag off.

**\*IMP** This parameter sets a flag in the printer device description that allows a print quality selection other than draft when:

- The value in the page rotation (PAGRIT) parameter of the printer file is \*AUTO.
- The printer is attached to an emulator (for example, a 3477 InfoWindow display) that supports page rotation.

For automatic page rotation to be done by an actual 3812 SCS printer, the print quality control sent to the printer must specify draft quality. Therefore, without the \*IMP flag on, the system sends controls to the printer for draft selection. Because the 3812 SCS printer supports only one level of print quality, printed output is not affected by print quality selection.

With the \*IMP flag set on, the system sends the value in the print quality (PRTQLTY) parameter of the printer file directly to the emulator. This is done instead of changing the print quality to draft when the value of the page rotation (PAGRIT) parameter is \*AUTO.

**\*IMPOFF**

This parameter sets the \*IMP flag off.

**\*SIC** This parameter sets a flag in the printer device description that sends an ASCII Set Initial Conditions command to the printer. This command turns off the font intervention messages for a 4019 or 4029 printer.

The 4019 or 4029 printer must be attached to a 3477 InfoWindow display and be configured as a 5219 printer.

The \*ON flag must be set in the 4019 or 4029 device description.

**Note:** Do not attempt to turn this flag on for any device other than a 4019 or 4029 attached to a 3477 InfoWindow display.

**\*SICOFF**

This parameter sets the \*SIC flag off.

**\*COR** This parameter sets a flag in the printer device description that enables computer output reduction (value of \*COR in the page rotation (PAGRIT) parameter of the printer file.)

\*COR is only needed if the \*ON flag is set.

The printers must be attached to a 348x InfoWindow, configured as a 5219, and have the \*ON flag set in the printer device description.

To the system, printers configured as 5219 appear the same whether they are attached to a 3477 or to a 348x InfoWindow display.



The \*ON flag prevents computer output reduction from being done on printers attached to the 348x or 3477. This is because the 3477 does not support computer output reduction. Therefore, if the \*ON flag is set, the \*COR flag must be set on to enable computer output reduction for printers configured as 5219 and attached to a 348x InfoWindow display.

**\*COROFF**

This parameter sets the \*COR flag off.

**\*RST** This parameter sets a flag in the printer device description that causes the writer to reset the printer at the beginning of each spooled file. In most environments, this might cause a noticeable decrease in performance because of the SNA overhead involved in doing the reset. This flag can be set for any SCS printer.

**\*RSTOFF**

This parameter sets the \*RST flag off.

**\*ON5256**

This parameter causes the operating system to configure the printer as a 5256 printer.

**\*ON5262**

This parameter causes the operating system to configure the printer as a 5262 printer.

**\*OFF52**

This parameter turns off the \*ON5256 and \*ON5262 flags in the device description.

**\*ON4214**

This parameter causes the operating system to configure the printer as a 4214 printer.

**\*OF4214**

This parameter turns off the \*ON4214 flag in the device description.

**Related reference:**

“Printer font support” on page 317

This table contains font ID substitution information which identifies which fonts are supported by certain printers. For example, if your application specified a certain font ID that your printer did not support, you can find out which printers support that font and route your printed output to a printer that supports that font ID.

**QWP4019 program use**

These examples show how to use the QWP4019 program.

**Example 1**

Set the 4019 flag on and then off in the device description for PRT01.

QWP4019 CALL	Result
CALL QWP4019 (PRT01 *ON)	Sets the 4019 flag on in the device description for PRT01.
CALL QWP4019 (PRT01 *CHECK)	The system returns 4019 because the 4019 flag is set on.
CALL QWP4019 (PRT01 *OFF)	Sets the 4019 flag off in the device description for PRT01. <b>Note:</b> This CALL also sets flags *SIC and *COR off.
CALL QWP4019 (PRT01 *CHECK)	The system returns 5219 or 3812 because the 4019 flag is off.

**Example 2**

Enable COR and the 4019 fonts for a 4019 printer attached to a 348x InfoWindow display.

QWP4019 CALL	Result
CALL QWP4019 (PRT01 *ON)	Sets the 4019 flag on in the device description for PRT01. This supplies the 4019 fonts but disables computer output reduction.
CALL QWP4019 (PRT01 *COR)	Sets the *COR flag on in the device description. This enables computer output reduction for PRT01.

### Example 3

Enable near letter quality (NLQ) print quality for an IBM Personal Printer Series II 2390 printer attached to a 3477 InfoWindow display.

QWP4019 CALL	Result
CALL QWP4019 (PRT01 *IMP)	Turns the *IMP flag on in the device description for PRT01.
CALL QWP4019 (PRT01 *CHECK)	The system returns 5219 or 3812 because the 4019 flag is off. There is no check for *IMP, *SIC, and *COR flags.

### Example 4

The following table provides a summary of the QWP4019 parameters, the call used to set flags on, and the call used to set flags off.

Parameter name	Call to set flag on	Call to set flag off
*ON	CALL QWP4019 (PRT01 *ON)	CALL QWP4019 (PRT01 *OFF) <sup>1</sup>
*IMP	CALL QWP4019 (PRT01 *IMP)	CALL QWP4019 (PRT01 *IMPOFF)
*SIC	CALL QWP4019 (PRT01 *SIC)	CALL QWP4019 (PRT01 *SICOFF)
*COR	CALL QWP4019 (PRT01 *COR)	CALL QWP4019 (PRT01 *COROFF)
*CNT	CALL QWP4019 (PRT01 *CNT)	CALL QWP4019 (PRT01 *CNTOFF)
*RST	CALL QWP4019 (PRT01 *RST)	CALL QWP4019 (PRT01 *RSTOFF)
*ON5256	CALL QWP4019 (PRT01 *ON5256)	CALL QWP4019 (PRT01 *OFF52)
*ON5262	CALL QWP4019 (PRT01 *ON5262)	CALL QWP4019 (PRT01 *OFF52)
*ON4214	CALL QWP4019 (PRT01 *ON4214)	CALL QWP4019 (PRT01 *OF4214)
<b>Note:</b>		
<sup>1</sup> Using the CALL QWP4019 (PRT01 *OFF) command sets the *SIC, *COR, and *ON flags off.		

## QPQCHGCF program

The QPQCHGCF program provides a way for you to indicate whether a particular section of a double-byte coded font is resident in the printer or should be downloaded.

A particular section of a double-byte coded font can be in one of the following status:

- Resident in the printer and should not be downloaded
- Is not resident in the printer or has been changed (different version than the one in the printer) and needs to be downloaded by PSF to the printer.

## QPQCHGCF parameter names and functions

The QPQCHGCF program has these parameters.

### Parameters:

1	Coded font name	Input	Char(8)
2	Coded font library name	Input	Char(10)
3	Font character set name	Input	Char(10)
4	Resident font indicator	Input	Char(4)

### Coded font name:

Specifies the name of the coded font to be marked. This is an 8-character input parameter.

### Coded font library name:

Specifies the name of the library containing the coded font. This is a 10-character input parameter.

You can use the following special value for the library name:

**\*LIBL** This indicates that the job's current library list will be used to search for the coded font.

### Font character set name:

Specifies the name of the font character set to mark within the coded font. This is an 8-character input parameter.

The font character set name can be specified with the following special value:

**\*ALL** This indicates that all the font character set and code page pairs in the coded font are to be marked.

The font character set name can be a generic name. A generic name is a character string of one or more characters followed by an asterisk (\*); for example, C0S0\*. The asterisk substitutes for any valid characters. A generic name specifies all font character sets with names that begin with the generic prefix. If an asterisk is not included in the name, the system assumes it to be the complete font character set name. To change all the font character sets in a certain range, for example, C0G16F70 - C0G16F7F, you should specify C0G16F7\* for the font character set name. Specifying C0G16F\* can be the same as \*ALL and change all the font character sets in the coded font (assuming all font character set names started with C0G16F0).

### Resident font indicator:

Specifies whether the font character set is resident in the printer, or if it is not resident in the printer and needs to be downloaded by the system.

**\*NO:** The font character set is not resident in the printer and needs to be downloaded by the system to the printer. Also, the font character set might be resident in the printer, but has been modified. In that case, \*NO should be specified.

:PK **\*YES:** The font character set is resident in the printer and does not need to be downloaded by the system to the printer.

### Notes:

- All IBM-supplied coded fonts are shipped with the resident font indicator turned off. That means the entire font will be downloaded unless the QPQCHGCF program is run to mark the IBM-supplied sections as resident.
- The font character set and code page pair are treated the same for marking them resident or needing to be downloaded. If the font character set has changed, then the corresponding code page will also be

marked as needing to be downloaded. If the code page has been changed, then the corresponding font character will also be marked as needing to be downloaded.

## QPQCHGCF program use

These examples show how to mark font-character-set or code-page pairs as resident using the QPQCHGCF program.

### Example 1:

The following example marks all the font-character-set and code-page pairs as resident in the printer for coded font X0G16F in library QFNT61. No user-defined sections will be downloaded.

```
CALL QPQCHGCF (X0G16F QFNT61 *ALL *YES)
```

### Example 2:

The following example marks all the font-character-set and code-page pairs in sections 41 - 4F as resident, and then marks sections 50 - 55 as resident.

```
CALL QPQCHGCF (X0G16B QFNT61 C0G16F4* *YES)
CALL QPQCHGCF (X0G16B QFNT61 C0G16F50 *YES)
CALL QPQCHGCF (X0G16B QFNT61 C0G16F51 *YES)
CALL QPQCHGCF (X0G16B QFNT61 C0G16F52 *YES)
CALL QPQCHGCF (X0G16B QFNT61 C0G16F53 *YES)
CALL QPQCHGCF (X0G16B QFNT61 C0G16F54 *YES)
CALL QPQCHGCF (X0G16B QFNT61 C0G16F55 *YES)
```

### Example 3:

The following example marks all the font-character-set and code-page pairs in sections 41 - 4F as resident. This then marks section 48 to be downloaded. Sections 50 - 5F are marked as resident, and sections 60 - 68 are marked as resident.

```
CALL QPQCHGCF (X0G16F QFNT61 C0G16F4* *YES)
CALL QPQCHGCF (X0G16F QFNT61 C0G16F48 *NO)
CALL QPQCHGCF (X0G16F QFNT61 C0G16F5* *YES)
CALL QPQCHGCF (X0G16F QFNT61 C0G16F60 *YES)
CALL QPQCHGCF (X0G16F QFNT61 C0G16F61 *YES)
CALL QPQCHGCF (X0G16F QFNT61 C0G16F62 *YES)
CALL QPQCHGCF (X0G16F QFNT61 C0G16F63 *YES)
CALL QPQCHGCF (X0G16F QFNT61 C0G16F64 *YES)
CALL QPQCHGCF (X0G16F QFNT61 C0G16F65 *YES)
CALL QPQCHGCF (X0G16F QFNT61 C0G16F66 *YES)
CALL QPQCHGCF (X0G16F QFNT61 C0G16F67 *YES)
CALL QPQCHGCF (X0G16F QFNT61 C0G16F68 *YES)
```

## Restrictions on using the QPQCHGCF program

These are restrictions when using the QPQCHGCF program.

- When marking fonts, the print writer must be stopped and restarted. If fonts are marked while the writer is active, unpredictable results will occur.
  - End PSF (ENDWTR), if it is active.
  - Use QPQCHGCF to mark the coded font.
  - Start PSF (STRPRTWTR).
- If a section of a font is modified, the modified section must contain all rotations that the original font contained. For example, if the coded font X0M16B is resident in the printer in rotations 0, 90, 180 and 270, and section 46 is modified, then the modified section 46 must also contain rotations 0, 90, 180, and 270.
- If IBM-supplied coded fonts are modified, you should not remove sections from the coded font as supplied. This might result in incorrect results when the modified font is referred to in a job being printed on device that does not support resident double byte raster fonts.

- PSF does not support referencing a double-byte resident raster font by its registered font ID. That is, you should not specify a double-byte font on the FONT parameter of the printer file, on the FONT DDS keyword, or with any other application that allows you to specify font with its registered font ID.

### Coded fonts whose font character sets reside in the 3130

The list in this topic shows the DBCS fonts that are resident in the 3130 printer.

Japanese (In QFNT61 library)

FONT	RESIDENT FONT				
FONT	SIZE	CODED FONT	CHARACTER SET	FONTID	WIDTH
Mincho	16x16	X0M16B/F	C0M16FXX	53559	096
Mincho	24x24	X0M24B/F	C0M24FXX	53559	140
Mincho	20x24	X0Z24B/F	C0Z24FXX	53559	144
Mincho	26x26	X0M26B/F	C0M26FXX	53559	156
Mincho	32x32	X0M32B/F	C0M32FXX	53559	180
Mincho	36x36	X0M36B/F	C0M36FXX	53559	216
Mincho	40x40	X0M40B/F	C0M40FXX	53559	240
Mincho	44x44	X0M44B/F	C0M44FXX	53559	264
Mincho	48x48	X0M48B/F	C0M48FXX	53559	288
Mincho	52x52	X0M52B/F	C0M52FXX	53559	312
Mincho	64x64	X0M64B/F	C0M64FXX	53559	384
Gothic	16x16	X0G16B/F	C0G16FXX	53815	100
Gothic	20x24	X0G20B/F	C0G20FXX	53813	144
Gothic	24x30	X0G24B/F	C0G24FXX	53813	140
Gothic	32x32	X0G32B/F	C0G32FXX	53815	192
Gothic	36x36	X0G36B/F	C0G36FXX	53815	216
Gothic	48x48	X0G48B/F	C0G48FXX	53815	288
Gothic	64x44	X0G64B/F	C0G64FXX	53815	384
R-Gothic	36x36	X0R36B/F	C0R36FXX	54071	216
R-Gothic	40x40	X0R40B/F	C0R40FXX	54071	240
R-Gothic	48x48	X0R48B/F	C0R48FXX	54071	288
R-Gothic	64x64	X0R64B/F	C0R64FXX	54071	384

Korean (In QFNT62 library)

FONT	RESIDENT FONT				
FONT	SIZE	CODED FONT	CHARACTER SET	FONTID	WIDTH
Mincho	24x24	X0M24K/L	C0HB00XX	53559	144
Mincho	32x32	X0M32K/L	C0HD00XX	53559	192
Mincho	36x36	X0M36K/L	C0HE00XX	53559	216
Mincho	40x40	X0M40K/L	C0HF00XX	53559	240
Mincho	48x48	X0M48K/L	C0HG00XX	53559	288
Mincho	64x64	X0M64K/L	C0HH00XX	53559	384
Gothic	16x16	X0G16K/L	C0HA00XX	53815	096
Gothic	24x30	X0G24K/L	C0HC00XX	53813	180

Traditional Chinese (In QFNT63 library)

FONT	RESIDENT FONT				
FONT	SIZE	CODED FONT	CHARACTER SET	FONTID	WIDTH
Ming	24x24	X0M24T	C0TB00XX	54583	144
Ming	32x32	X0M32T	C0TC00XX	54583	192
Ming	40x40	X0M40T	C0TD00XX	54583	240
Gothic	16x16	X0G16T	C0TA00XX	53815	096

Simplified Chinese (In QFNT64 library)

FONT	RESIDENT FONT				
FONT	SIZE	CODED FONT	CHARACTER SET	FONTID	WIDTH
Song	26x26	X0S26P	C0S26PXX	54327	144
Song	32x32	X0S32P	C0S32PXX	54327	192
Song	40x40	X0S40P	C0S40PXX	54327	240
Gothic	16x16	X0G16P	C0G16PXX	53815	096

Thai (In QFNT65 library)				
FONT	RESIDENT FONT			
FONT	SIZE	CODED FONT	CHARACTER SET	FONTID WIDTH
Official	24x40	X0040F	C0040FXX	57655 240
Official	24x60	X0060F	C0060FXX	57655 360
Italics	24x60	X0I60F	C0I60FXX	58039 360

## QPQCHGCF instructions for marking coded fonts

Included in the QPQCHGCF program is file QCDEFNT in library QGPL. This file contains five members: QFNT61, QFNT62, QFNT63, QFNT64, and QFNT65. Each member contains control language (CL) statements for marking the IBM-supplied sections of the fonts included in libraries QFNT61 - QFNT65.

To run a batch job to mark IBM-supplied sections as resident, you must edit the appropriate member in QCDEFNT.

- Edit the JOBDD parameter on the //BCHJOB statement. This JOBDD should have enough authority to change the coded fonts in the QFNT6X libraries.
- Edit this source file if you have changed any IBM-supplied sections and want these sections to be downloaded.

After the source file has been edited, you can run the batch job to mark the specified fonts. This is done with the Start Data Base Reader (STRDBRDR) command .

```
STRDBRDR FILE(QGPL/QCDEFNT)
MBR(QFNT6X)
```

where the member name (MBF) is QFNT61, QFNT62, QFNT63, QFNT64, OR QFNT65.

In order that the marked coded fonts take effect, you must perform the following tasks.

- End PSF (ENDWTR), if it is active.
- Run batch program to mark coded fonts.
- Start PSF (STRPRTWTR).

QCDEFNT in QUSRSYS contains the following five members:



```
QFNT61 - CL statements to mark Japanese coded fonts
QFNT62 - CL statements to mark Korean coded fonts
QFNT63 - CL statements to mark Traditional Chinese coded fonts
QFNT64 - CL statements to mark Simplified Chinese coded fonts
QFNT65 - CL statements to mark Thai coded fonts
```

---



## Related information for Basic printing

Product manuals, IBM Redbooks® publications, Web sites, and other information center topic collections contain information that relates to the Basic printing topic collection. You can view or print any of the PDF files.





### Manuals

- *About Type: IBM's Technical Reference for 240-Pel Digitized Type*
- Data Stream and Object Architectures: Graphics Object Content Architecture,  (1.1 MB)
- Font Object Content Architecture (FOCA) Reference  (1.9 MB)

## IBM Redbooks

- IBM AS/400 Printing V, SG24-2160 
- IBM AS/400 Printing VI, SG24-6250 

## Web sites

- InfoPrint Solutions Company - Software - i5/OS  ([www.infoprintsolutionscompany.com/internet/wwwsites.nsf/vwWebPublished/soft\\_i5oshome\\_us](http://www.infoprintsolutionscompany.com/internet/wwwsites.nsf/vwWebPublished/soft_i5oshome_us))
- IBM - Print Services Facility for i5/OS  ([www-03.ibm.com/systems/i/software/print/psfhome\\_i\\_ww.html](http://www-03.ibm.com/systems/i/software/print/psfhome_i_ww.html))
- InfoPrint Solutions Company : Support & Maintain  ([www.infoprintsolutionscompany.com/internet/wwwsites.nsf/vwWebPublished/sm\\_home\\_us](http://www.infoprintsolutionscompany.com/internet/wwwsites.nsf/vwWebPublished/sm_home_us))
- Support for IBM System i  ([www.ibm.com/support/](http://www.ibm.com/support/))

## Other information

- Advanced Function Presentation
- DDS Reference: Printer Files
- i5/OS NetServer
- Print APIs
- PrintManager APIs
- Reclaim spooled file storage Experience Report
- Spool performance considerations Experience Report
- System i Access for Web
- System i Access for Windows
- Work Management
- Working with printer output

---

## Code license and disclaimer information

IBM grants you a nonexclusive copyright license to use all programming code examples from which you can generate similar function tailored to your own specific needs.

SUBJECT TO ANY STATUTORY WARRANTIES WHICH CANNOT BE EXCLUDED, IBM, ITS PROGRAM DEVELOPERS AND SUPPLIERS MAKE NO WARRANTIES OR CONDITIONS EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT, REGARDING THE PROGRAM OR TECHNICAL SUPPORT, IF ANY.

UNDER NO CIRCUMSTANCES IS IBM, ITS PROGRAM DEVELOPERS OR SUPPLIERS LIABLE FOR ANY OF THE FOLLOWING, EVEN IF INFORMED OF THEIR POSSIBILITY:

1. LOSS OF, OR DAMAGE TO, DATA;
2. DIRECT, SPECIAL, INCIDENTAL, OR INDIRECT DAMAGES, OR FOR ANY ECONOMIC CONSEQUENTIAL DAMAGES; OR
3. LOST PROFITS, BUSINESS, REVENUE, GOODWILL, OR ANTICIPATED SAVINGS.

SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF DIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, SO SOME OR ALL OF THE ABOVE LIMITATIONS OR EXCLUSIONS MAY NOT APPLY TO YOU.





---

## Appendix. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing  
Legal and Intellectual Property Law  
IBM Japan, Ltd.  
3-2-12, Roppongi, Minato-ku, Tokyo 106-8711

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:** INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation

Software Interoperability Coordinator, Department YBWA  
3605 Highway 52 N  
Rochester, MN 55901  
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, IBM License Agreement for Machine Code, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

#### 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.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. \_enter the year or years\_.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

---

## Programming interface information

This Basic printing publication documents intended Programming Interfaces that allow the customer to write programs to obtain the services of IBM i.

---

## Trademarks

IBM, the IBM logo, and [ibm.com](http://ibm.com) are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at Copyright and trademark information at [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Windows is a trademark of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

---

## Terms and conditions

Permissions for the use of these publications is granted subject to the following terms and conditions.

**Personal Use:** You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative works of these publications, or any portion thereof, without the express consent of IBM.

**Commercial Use:** You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.







Printed in USA