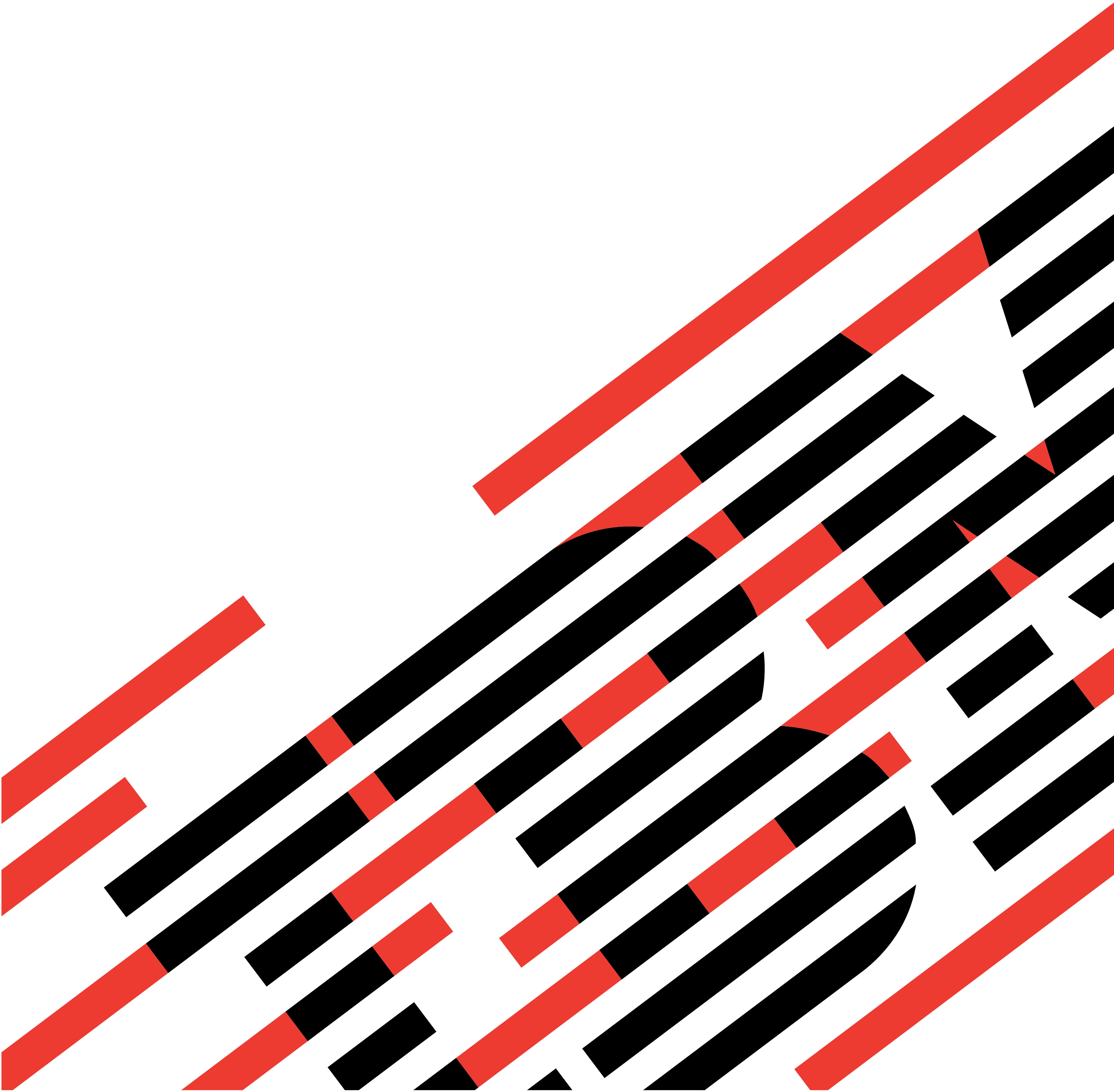
iSeries

Basic Printing

Version 5 Release 3



IBM

eserver

iSeries

Basic Printing

Version 5 Release 3

Note

Before using this information and the product it supports, be sure to read the information in "Notices," on page 141.

Third Edition (August 2004)

This edition applies to version 5, release 3, modification 0 of OS/400 (product number 5722–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.

Contents

Basic Printing

IBM® iSeries™ servers have powerful printing and display functions. You can present information using overlays, bar codes, graphics, images, and more. iSeries servers support a variety of industrial-strength printing and presentation solutions.

The following information illustrates some of the core iSeries printing functions, and helps you plan for and configure these functions.

“What’s new for V5R3” on page 2
Information about the changes and improvements made to Basic printing since the last release.

“Print this topic” on page 2
A PDF version of this information topic.

“Printing concepts” on page 3
Use this information to understand how printing works on the iSeries server.

“Plan for printing” on page 80
Use this information to select the printing method that best meets your needs. This topic presents an overview of several common printing protocols and their hardware requirements.

“Scenarios: Select a print protocol” on page 82
See examples of printing solutions that illustrate the advantages and requirements of common printing methods, and provides example configurations. This topic includes SNMP, PJL, IPP, IPDS™ (PSF/400), and LPR/LPD printing.

“Configure printing” on page 87
Refer to this procedure for instructions on setting up different aspects of printing configuration including the communication between the iSeries server and the printers.

“Manage printing” on page 99
Find resources for administering and working with your printing environment after successful configuration.

“Troubleshoot printing” on page 138
Find help for resolving problems.

“Printing reference” on page 112
Find additional information on printing related CL commands and DDS.

“Related information for printing” on page 139
Get detailed information about iSeries printing from the manuals, IBM Redbooks™, and Web sites listed here.

Note: Read the “Code disclaimer information” for important legal information.

Code 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. 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 INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO SOME OR ALL OF THE ABOVE LIMITATIONS OR EXCLUSIONS MAY NOT APPLY TO YOU.

What's new for V5R3

For V5R3, iSeries basic printing has the following new functions:

- **Spooled files and output queues supported in independent ASPs**
  Spooled files and output queues can now be placed into an independent auxiliary storage pool (independent ASP).
- **Improved TrueType font support**
  Improvements to the processing of TrueType and TrueType linked fonts. See the Printer Device Programming book for more information.

Enhancements to Basic printing information

An extensive amount of information is added to the Basic printing topic (formerly known as the Printing topic) this release. Most of this information is not new, but is moved from the Printer Device Programming book.

To find other information about what is new or changed this release, see the "Memo to Users" topic.

Print this topic

To view or download the PDF version of this document, select Basic Printing (about 3,000 KB).

Other information

You can view or print PDFs of related manuals and IBM Redbooks from "Related information for printing" on page 139.

Saving PDF files

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

- Right-click the PDF in your browser (right-click the link above).
- Click **Save Target As...** if you are using Internet Explorer. Click **Save Link As...** if you are using Netscape Communicator.
- Navigate to the directory in which you would like to save the PDF.
- Click **Save**.

Downloading Adobe Acrobat Reader

You need Adobe Acrobat Reader to view or print these PDFs. You can download a copy from the Adobe Web site (www.adobe.com/products/acrobat/readstep.html) .

Printing concepts

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

“Basic printing terminology”
Provides a description of some of the common printing terms used in this topic.

“Overview of the printing process” on page 4
Provides a high-level overview of the printing process.

“Printer file” on page 6
Provides a description of a printer file.

“Spooled files and output queues” on page 18
Provides a description of spooled files, output queues, and how they relate.

“Printer writer program” on page 32
Provides a description of printer writer programs.

“Printer data streams” on page 33
Provides a description of printer data streams.

“Printer device description” on page 46
Provides a description of printer device descriptions.

“User profile” on page 46
Provides a description of how user profiles relate to printing.

“Job description” on page 46
Provides a description of how job descriptions relate to printing.

“Workstation description” on page 46
Provides a description of how workstation descriptions relate to printing.

“System values” on page 47
Provides a description of how system values relate to printing.

“Controlling printing to output queue or printer” on page 47
Provides a description of the various elements that control how printer output is routed to its destination.

“Remote system printing” on page 60
Provides a description of sending spooled files to remote systems using remote output queues.

“Considerations” on page 64
Provides a description of additional considerations related to specific situations.

For more information, refer to the Printer Device Programming manual .

Basic printing terminology

The following is a list of some terminology that you may find in this topic:

auxiliary storage pool (ASP)
One or more storage units that are defined from the storage devices or storage device subsystems that make up auxiliary storage. An ASP provides a way of organizing data to limit the impact of storage-device failures and to reduce recovery time. See also disk pool.

disk pool
An auxiliary storage pool that contains only disk units.

independent ASP
See independent disk pool.

independent disk pool
One or more storage units that are defined from the disk units or disk-unit subsystems that make up addressable disk storage. An independent disk pool contains objects, the directories that contain the objects, and other object attributes such as authorization ownership attributes. An independent disk pool can be made available (varied on) and made unavailable (varied off) without restarting the system. An independent disk pool can be either a) switchable among multiple systems in a clustering environment or b) privately connected to a single system. Contrast with basic disk pool.

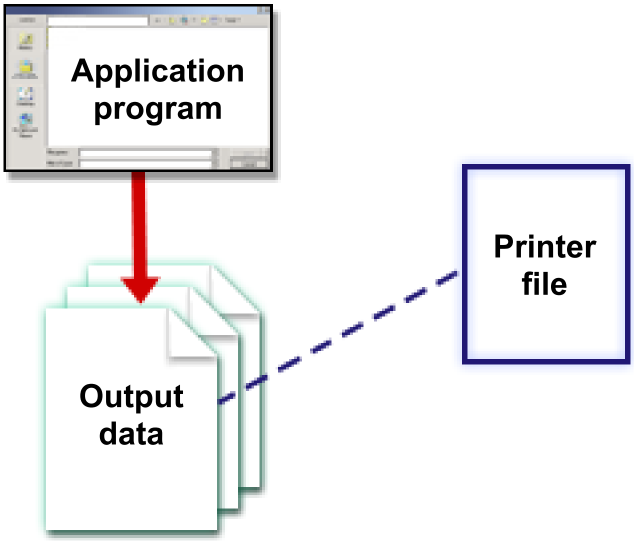
printer output
A file that holds output data that is waiting to be processed for printing. See also spooled file.

spooled file
A file that holds output data that is waiting to be processed for printing. See also printer output.

Overview of the printing process

The following steps provide a high-level view of the OS/400® 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. For more information about printer files, see “Printer file” on page 6.



2. If print spooling is selected, the output data is placed in a spooled file (see “Spooled file” on page 19) and the spooled file is placed in an output queue (see “Output queue” on page 19). If direct printing is selected, the output data is sent directly to the printer.

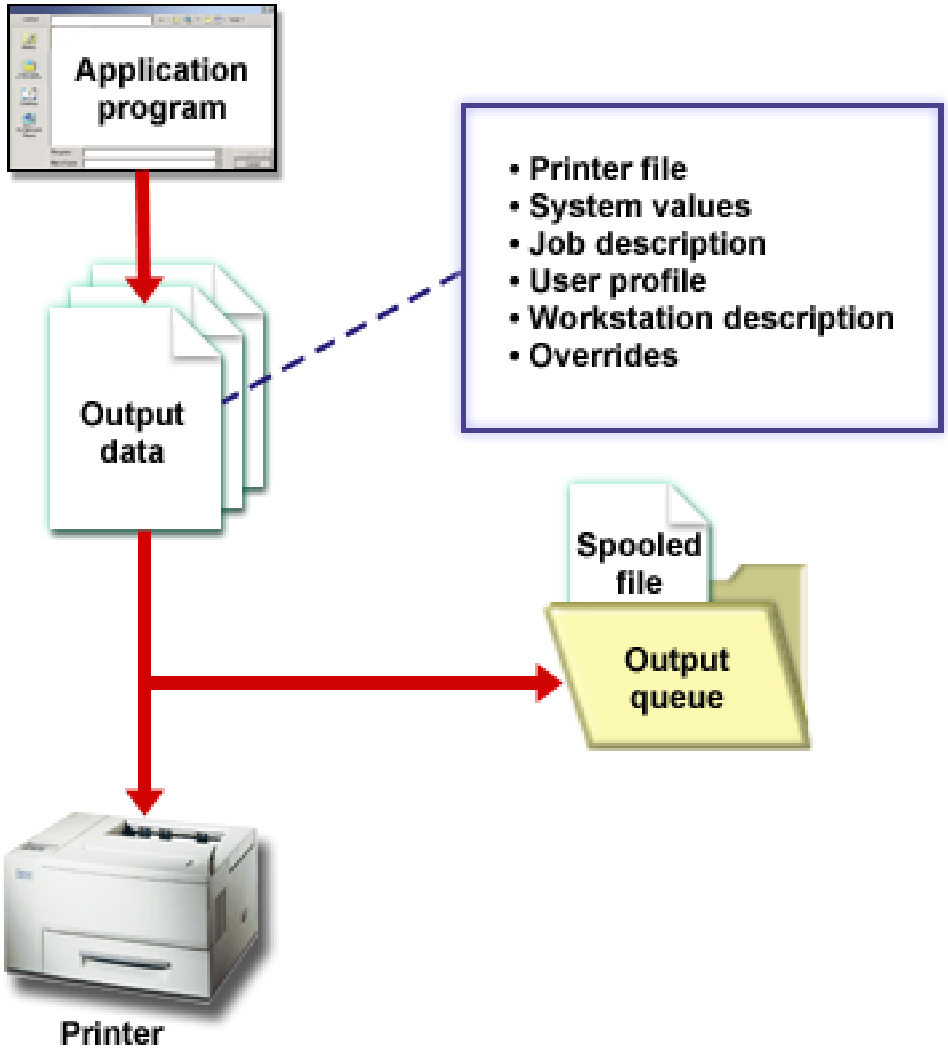
   The destination of the output data (see “Controlling printing to output queue or printer” on page 47) is based on values stored in several printing elements:

- “Job description” on page 46
- “User profile” on page 46
- “Workstation description” on page 46
- “Printer file” on page 6
- “System values” on page 47

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 OS/400 and non-OS/400 systems.

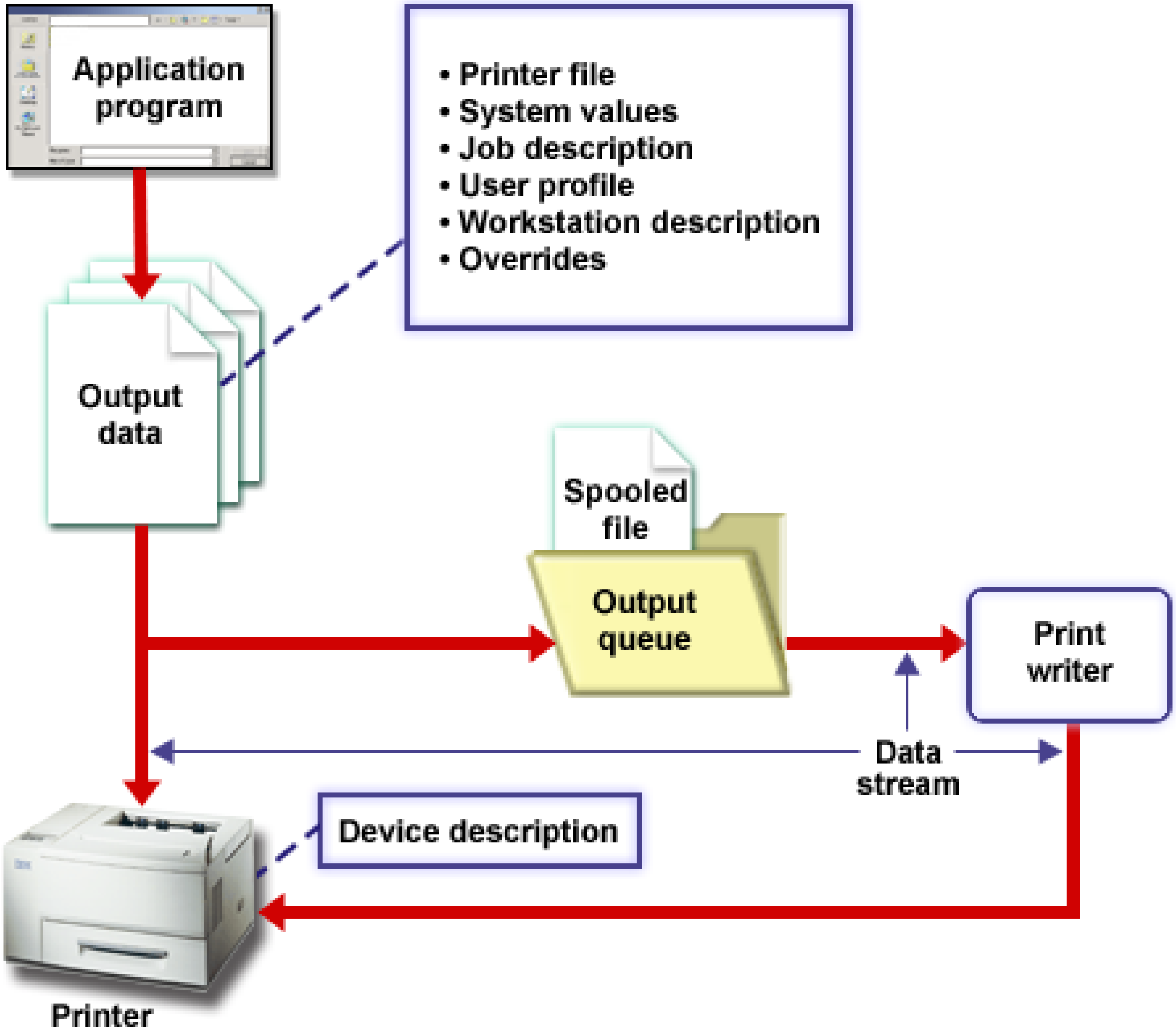


3. The printer writer program (see “Printer writer program” on page 32) interacts between the output queue and the printer and can be used to convert the printer data stream. The printer writer program

shipped with OS/400 supports a variety of printer data streams. Print Services Facility™ for OS/400 provides additional function that provides support for the Advanced Function Presentation™ (AFP™) Intelligent Printer Data Stream™ (IPDS). For more information about printer data streams, see “Printer data streams” on page 33.

Each printer must have a printer device description (see “Printer device description” on page 46). 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. For more information about remote writers and remote system printing, see “Remote system printing” on page 60.



Printer file

Read the following for more information on printer file concepts.

“Printer file overview” on page 7
Provides a description of printer files.

“Example: Using a program-described printer file with an application program” on page 8
Provides an example that shows how an application program calls and uses a program-described printer file to control how your printed output will look.

“Example: Using an externally described printer file with an application program” on page 12
Provides an example that shows how an application program calls and uses an externally described printer file to control how your printed output will look.

“Printer file overrides” on page 16
Provides more detailed information about the use of overrides for printer files.

Printer file overview

Printers attached to the iSeries server are supported by the operating system through printer files. Printer files describe how the system is to operate on the data as it passes between your application program and a printer.

A printer file handles every request for printing. There is one exception to this. 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 screen. The remote workstation controller, not the iSeries server, 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 could 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 could 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 server job debugging purposes, you might want the current job to own all service dumps, instead of the current user. To accomplish this, you could set the parameter to the special value of *JOB for the QPSRVDMP 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. For more information, see “Example: Using a program-described printer file with an application program.”
- **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. For more information about DDS keywords, see DDS Reference: Printer files in the Programming topic. For more information about using externally described printer files, see “Example: Using an externally described printer file with an application program” on page 12.

You can create your own printer files by using the Create Printer File (CRTPRTF), or you can use system-provided printer files. See “Printer file parameter considerations” on page 112 for more detailed information on 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.

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 will look.

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: Read the “Code disclaimer information” on page 1 for important legal information.

```
*********************************************************************
                  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
                          MOVEACSAR      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 OVRPRTF command). See "Printer file overrides" on page 16 for more information on 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 would be 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 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
PAGRTT
PRTTXT
JUSTIFY
DUPLEX
IPDSPASTHR
USRRSCLIBL
CORNERSTPL
EDGESTITCH
SADLSTITCH
FNTRSL
SPOOL
SCHEDULE
USRDTA
SPLFOWN
USRDFNOPT
USRDFNDTA
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). 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
```

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 will look.

This example will consist 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: Read the “Code disclaimer information” on page 1 for important legal information.

```
************************************************************************
                   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
                     MOVEACSAR      CTSTZP 30
                     MOVEACSAR      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, the high-level language program, and in printer file overrides (through the OVRPRTF command). See “Printer file overrides” on page 16 for more information on 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 would be 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 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
PAGRTT
PRTTXT
JUSTIFY
DUPLEX
IPDSPASTHR
USRRSCLIBL
CORNERSTPL
EDGESTITCH
SADLSTITCH
FNTRSL
SPOOL
SCHEDULE
USRDTA
SPLFOWN
USRDFNOPT
USRDFNDTA
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
DFRWRT
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 on DDS source files, see DDS Reference: Printer files in the Programming topic.

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
```

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 page 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 may be entered interactively from a display station or as part of a batch job. They may be included in a control language (CL) program, or they may be issued from other programs through a call to the program QCMDEXC. 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 may 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 may 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:

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

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 may 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.

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.

See the following topics for more information:

- "Spooling overview"
- "Spooled file" on page 19
- "Output queue" on page 19
- "Multiple output queues" on page 20
- "Output queue recovery" on page 20
- "Spooled file recovery after an abnormal IPL or an abnormal vary on of an independent ASP group" on page 21
- "Default output queues for printers" on page 21
- "Order of spooled files on an output queue" on page 22
- "Data queue support for spooled files" on page 23
- "Spooled file names" on page 27
- "Spooled file security" on page 28
- "Output queue security" on page 29
- "QPRTJOB job" on page 29
- "Spooled file subsystem" on page 30
- "Spooled file library" on page 30
- "Spooled files in independent disk pools" on page 31

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 with the iSeries server. Specifying SPOOL = *YES or SPOOL = *NO on the SPOOL parameter of a printer file determines if 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.

For information on how to work with spooled files, see “Manage printing” on page 99.

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 DEVD 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 could be set from 08:00:00 to 17:30:00 hours. During this time, only spooled files of 100 or fewer pages would print. 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 "Control printing by spooled file size" on page 105 for more information on how to configure deferred printing.

Multiple output queues

You may want to create multiple output queues for:

- 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 could be given a separate output queue.
- Output of special system files

  You may want to consider separate queues for the following system-supplied files:
  - QPJOBLOG: You may want all job logs sent to a separate queue.
  - QPPGMDMP: You may want all program dumps sent to a separate queue so you can review and print them if needed or clear them daily.
  - QPSRVDMP: You may 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 may 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 is performed by the QSPLMAINT system job, which 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 QRCL*xxxxx*, where *xxxxx* is the independent ASP number of the primary independent ASP of the independent ASP group (for example, QRCL00033 if the primary independent ASP number is 33). The move is performed by the QSPMN*xxxxx* job, where *xxxxx* 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 QSPLMAINT system job, which issues a completion message to the QSYSOPR message queue when all spooled files have been moved.

Spooled file recovery after an abnormal IPL or an abnormal vary on of an independent ASP group

System spooled file recovery starts immediately following an abnormal IPL. Spooled file recovery is done under the system job QSPLMAINT. 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 an abnormal vary on of an independent ASP group. Spooled file recovery is done under the system job QSPMN*xxxxx*, where *xxxxx* 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 QRCL*xxxxx*, where *xxxxx* is the independent ASP number of the primary independent ASP.

Default output queues for printers

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 (DEVD) 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 may 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 may 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 for spooled files

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” on page 25 for a description of the information contained in record type 01.

- **Environment variable data queue support**

  Using the ADDENVVAR or 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 created by a job (such as a remote command server job) and the spooled file was stored under a QPRTJOB.

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 would be:

```
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)
```

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.

The entry in the data queue has a format of record type 02. See “Record type 02 data queue entry format” on page 26 for a description of the information contained in record type 02.

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 iSeries server tries to add entries to a data queue that does not exist or 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.

Record type 01 data queue entry format: Following is 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: **01** 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. **CHAR(10)** Job name **CHAR(10)** User name **CHAR(6)** 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. **CHAR(10)** Output queue name **CHAR(10)** Library of the output queue

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

Decimal Offset	Hex Offset	Type	Description
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 create date. Identifies the date on which the spooled file was created in CYYMMDD format.
87	57	CHAR(1)	Reserved
88	58	CHAR(6)	Spooled file create time. Identifies the time that the spooled file was created in HHMMSS format.
94	5E	CHAR(34)	Reserved

Record type 02 data queue entry format: Following is 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: **02** 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. **CHAR(10)** Job name **CHAR(10)** User name **CHAR(6)** 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.

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

Decimal Offset	Hex Offset	Type	Description
52	34	CHAR(20)	Qualified output queue name Identifies the qualified name of the output queue on which the spooled file was placed. **CHAR(10)** Output queue name **CHAR(10)** 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. **CHAR(10)** Job name **CHAR(10)** User name **CHAR(6)** Job number
98	62	CHAR(10)	User data Identifies the user specified data for the spooled file that was created.
108	6C	BINARY(4)	Thread ID Identifies the thread of the job that created the spooled file.
112	70	CHAR(10)	System name Identifies the name of the system on which the spooled file was generated.
122	7A	CHAR(7)	Creation date Identifies the date on which the spooled file was created in CYYMMDD format.
129	81	CHAR(6)	Creation time Identifies the time that the spooled file was created in HHMMSS format.
135	87	CHAR(9)	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 would be called QSYSPRT because QSYSPRT is the printer file used by the Print key operation.

There are several ways in which the spooled file could 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 may 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 four 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 the user has *EXECUTE authority. This authority should only be granted 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 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.

  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.

For the Copy Spooled File (CPYSPLF), Display Spooled File (DSPSPLF), and Send Network Spooled File (SNDNETSPLF) commands, in addition to the four 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. The specific authority required is *READ data authority.

If the user is authorized to control the file by one of the four ways already listed above, 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) or a user with SPCAUT(*SPLCTL) may 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 the *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 the *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.

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 on spooled file and output queue security, see the “Security” topic.

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 iSeries server, 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: Using SPLFOWN parameter allows you to specify who owns the spooled file.

QPRTJOB jobs are created automatically by the system. There may 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 iSeries server.
- Sending a spooled file from VM or MVS™ through a VM/MVS bridge to an iSeries server.
- 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, you can create a spooled file for a different, current, or group user profile when SPLFOWN is set to *CURGRPPRF.
- Using the UNIX SETUID API, you can 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.

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 submit 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 QSPL*xxxx*, 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 may take 24 KB of storage.

It is best to keep the QSPL or QSPLxxxx library small by periodically deleting old spooled files with the DLTSPLF 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.

See “Reclaim spooled file storage” on page 103 for more information on how to remove spooled database members. This is the only allowable way to remove spooled database members from the QSPL or QSPLxxxx library. Any other way can cause severe problems.

Displaying the data in the QSPL or QSPLxxxx library may 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 server spooling functions can access them. Changing the library or files could cause some server spooling functions to work incorrectly and destroys the integrity of the spooled file security scheme.

Spooled files in independent disk pools

Spooled files may 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 the spooled file is being created on gets varied off (this would occur if a name space change was made and the reservation on the independent ASP was gone), then put and close errors could occur. This could also contribute to data inaccuracies in spool internal information. These inaccuracies would be fixed when the independent ASP is varied back on. Since the recovery of this condition is done in a background job, users will see some inconsistencies for those spooled files until the QSPMN*xxxxx* server 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 QSPMN*xxxxx* job is responsible for clearing unused DB members for deleted spooled files, automatic removal of DB members that have not been reused within the days specified on the Automatically clean up unused printer output storage (QRCLSPLSTG) system value, and movement of stranded spooled files onto the output queue QSPRCLOUTQ in the QRCL*xxxxx* library in the primary ASP when a damaged output queue is deleted by a user. There is one QSPMN*xxxxx* system server job for each ASP group that is varied on.

For more information about reclaiming spooled storage space, see “Reclaim spooled file storage” on page 103.

If a printer writer or job ends abnormally and causes a spooled file or output queue to become unusable or left in an unstable state where some operations are not permitted, the independent ASP must be varied off and then back on again. Calling QSPFIXUP will not fix spooled files or output queues on independent ASPs.

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.

For more information on working with independent ASPs, see the “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. See the CL command for more information.

- CHGJOB
- CHGWTR
- CPYSPLF
- HLDJOB
- RCLSPLSTG

- RLSJOB
- WRKJOB
- WRKSPLF

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 shipped with 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 iSeries server does not automatically assign a different printer to continue printing the jobs in the output queue that your printer was assigned to. You have to 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 OS/400 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 iSeries server. 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). See “Remote system printing” on page 60 for more information on remote writers.

Printer data streams

The iSeries server supports different data streams. See the following for more information:

- “SNA character string (SCS)”
- “Advanced Function Presentation Data Stream (AFPDS)” on page 34
- “Intelligent Printer Data Stream (IPDS)” on page 35
- “American National Standard Code for Information Interchange (ASCII)” on page 45

SNA character string (SCS)

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.

The following 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 acomplished 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 could be repeated several times.

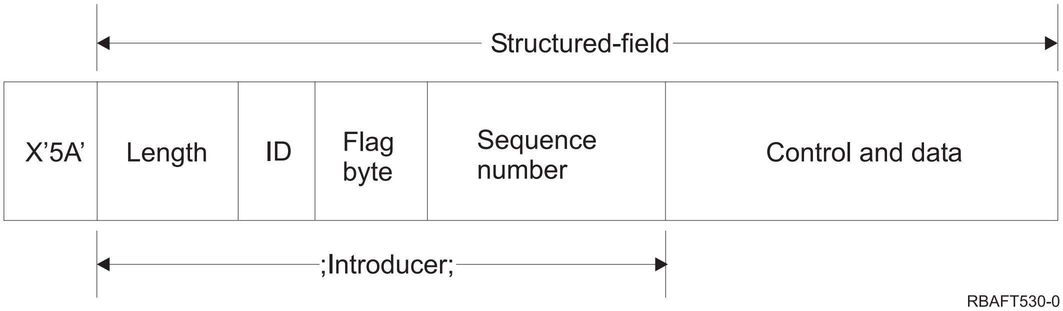
Page eject
The FF (forms feed) control code is inserted.

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

Advanced Function Presentation Data Stream (AFPDS)

AFPDS 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.

AFPDS 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.



RBAFT530-0

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 AFPDS

The following IBM licensed programs generate AFPDS data streams:

- Operating System/400® (OS/400)
- Advanced Function Printing™ Utilities for iSeries (AFP Utilities for iSeries)
- Document Composition Facility (DCF)
- DisplayWrite/390 (DW/390)
- Graphical Data Display Manager (GDDM®)
- zSeries® Advanced Function Presentation utilities:

- – 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

AFPDS 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 iSeries server uses the following objects when processing AFPDS:

AFPDS spool buffers
The AFPDS print file 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.

AFPDS 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 zSeries to an iSeries server or loaded from tape into space objects using OS/400 commands. Overlays and page segments can be created by AFP Utilities for iSeries.

Messages
Messages generated during the processing of AFPDS print files are placed in the printer writer job log.

Intelligent Printer Data Stream (IPDS)

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 features:

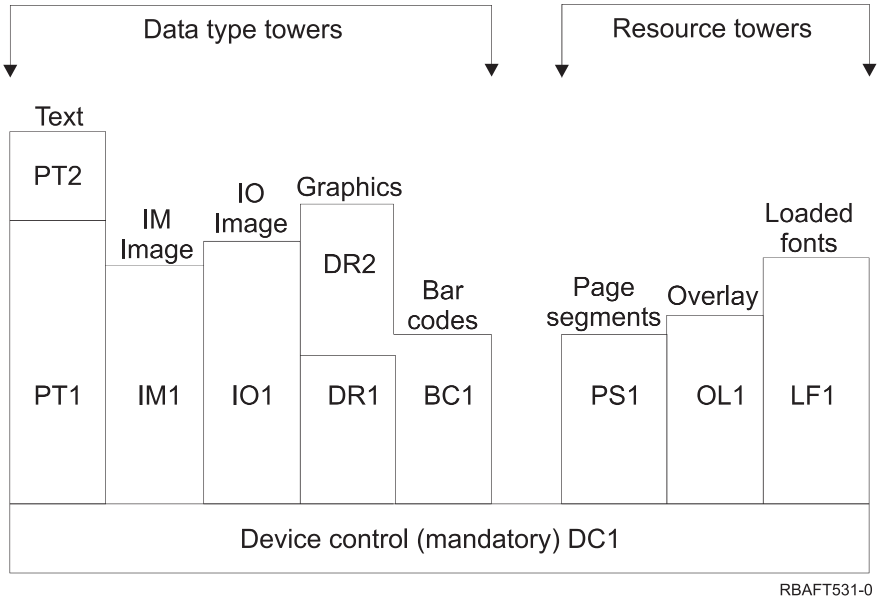
- Different applications may create source data (graphics, image, bar code, and text), independently of one another. IPDS makes it possible for the output of these independent applications to be merged at print time resulting in an integrated mixed data page.

  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 the following:
  - 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.



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 do the following:

- 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.

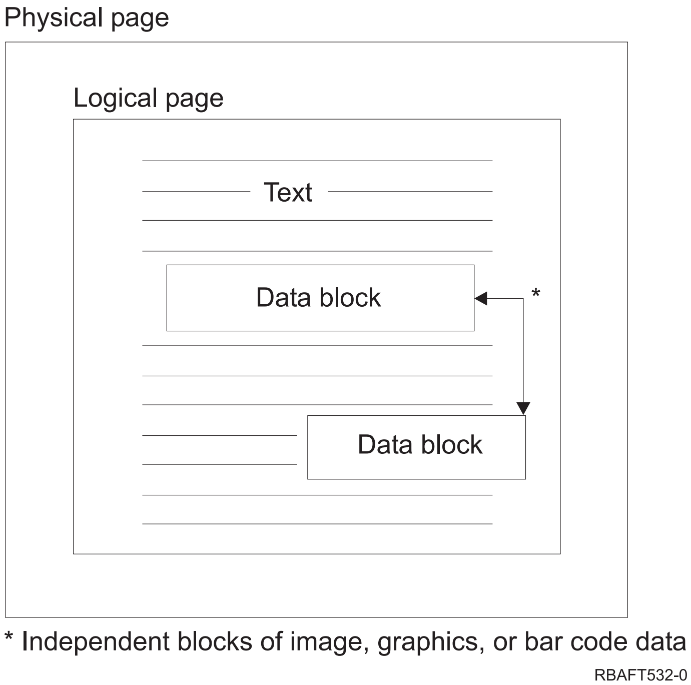
Logical page
The electronic representation of the page that is sent to the printer. The logical page is a rectangular area that may or may not be the same size as the physical page on which it is placed. Printing can only occur where the current logical page intersects the physical page (valid printable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



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 could be produced on an editor, image data could be the output of a scanner stored in a folder, and graphics data could 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 may 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 may 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.

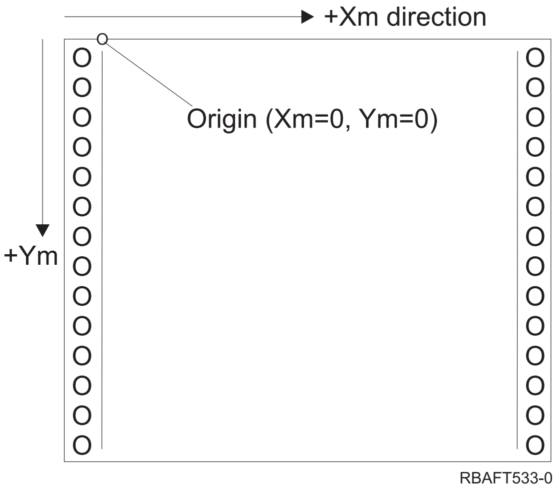
Coordinate systems

Xm, Ym 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 Xm, Ym 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.



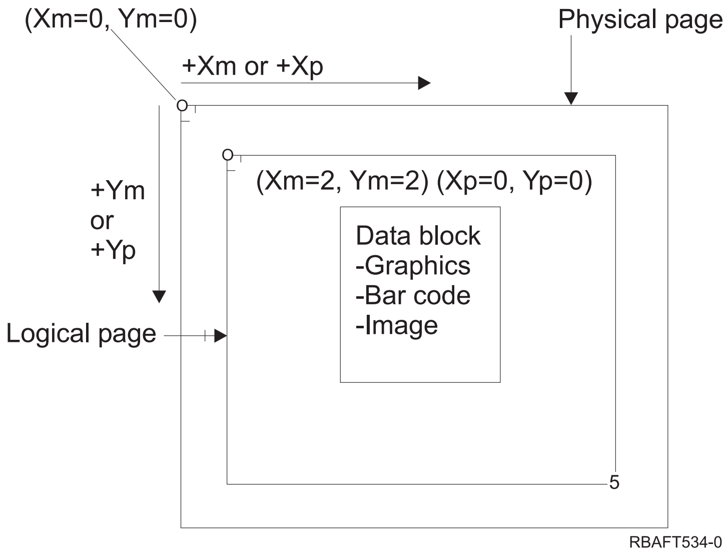
The Xm, Ym coordinate system is fixed for each media size. IPDS commands cannot change the orientation of these coordinates.

Xp, Yp Coordinate System (Logical Page)

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

The size of the logical page in the Xp dimension is called the *Xp extent*. The size of the logical page in the Yp dimension is called the *Yp extent*. The Xp, Yp coordinate system is used to locate data blocks on the logical page.

The coordinate system for overlays is the same as the Xp, Yp coordinate system for logical pages.



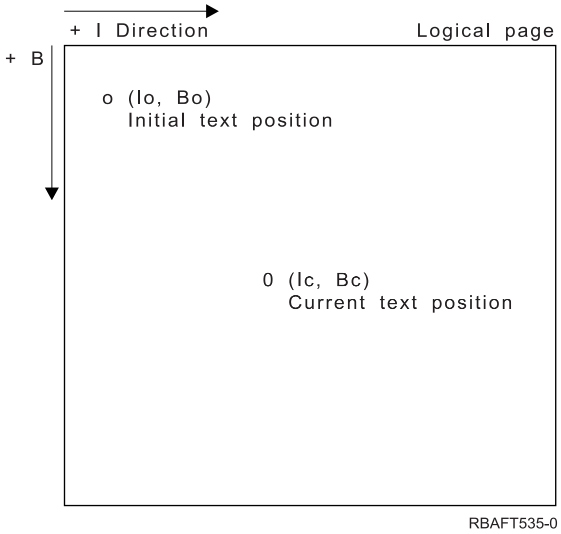
I, B Coordinate System (Text)

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 may actually place characters or lines on a page in various directions (as in bidirectional printing).

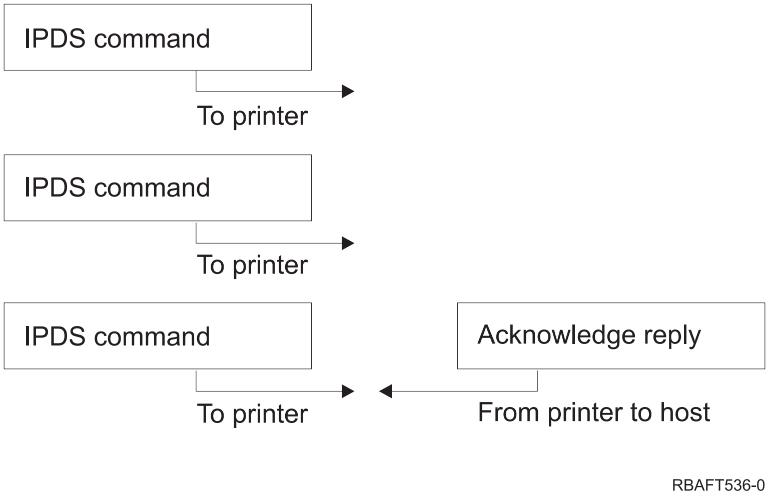
The coordinates of the first text position on the logical page are called the initial inline text coordinate (Io) and the initial baseline text coordinate (Bo). The coordinates of the current position on the logical page are called the current inline text coordinate (Ic) and the current baseline text coordinate (Bc).



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:



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.

Mixed Object: Document Content Architecture (MO:DCA)

The ability to print documents with consistent output, independent of either operating system or printer, is extremely important to the user of printed data. In order to help achieve this goal, IBM has defined a single object-oriented data stream—**Mixed Object Document Content Architecture (MO:DCA)**. (An object is a collection of data that can be treated as a unit.) This architecture has been developed in order to meet several objectives:

- 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 strategic architecture for the interchange of revisable and presentation form of documents and objects used as resources is MO:DCA, which has evolved from Revisable Form Text: Document Content Architecture (RFT:DCA).

The data stream for an MO:DCA document consists of various objects, such as text, images, and graphics, 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 layout structure defines the way the data should be presented.

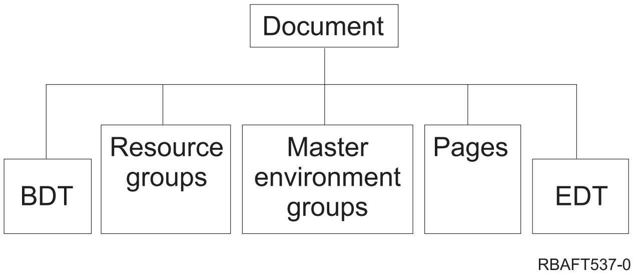


Figure 1. MO:DCA Document Structure

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 could 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 could 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 may 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

American National Standard Code for Information Interchange (ASCII)

There is no formal structure controlling the use of the 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 down 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 OS/400, ASCII printing support is provided by translating iSeries server EBCDIC characters to the ASCII equivalents.

Printer device description

Printer device descriptions have to 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.

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

Job description

A job description is 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)

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 would use the same printer device or output queue that is currently specified in the workstation description.

System values

System values are objects supplied by IBM and shipped with 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.

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. See the following for more details:

- "Routing process order"
- "Routing examples" on page 49

Routing process order

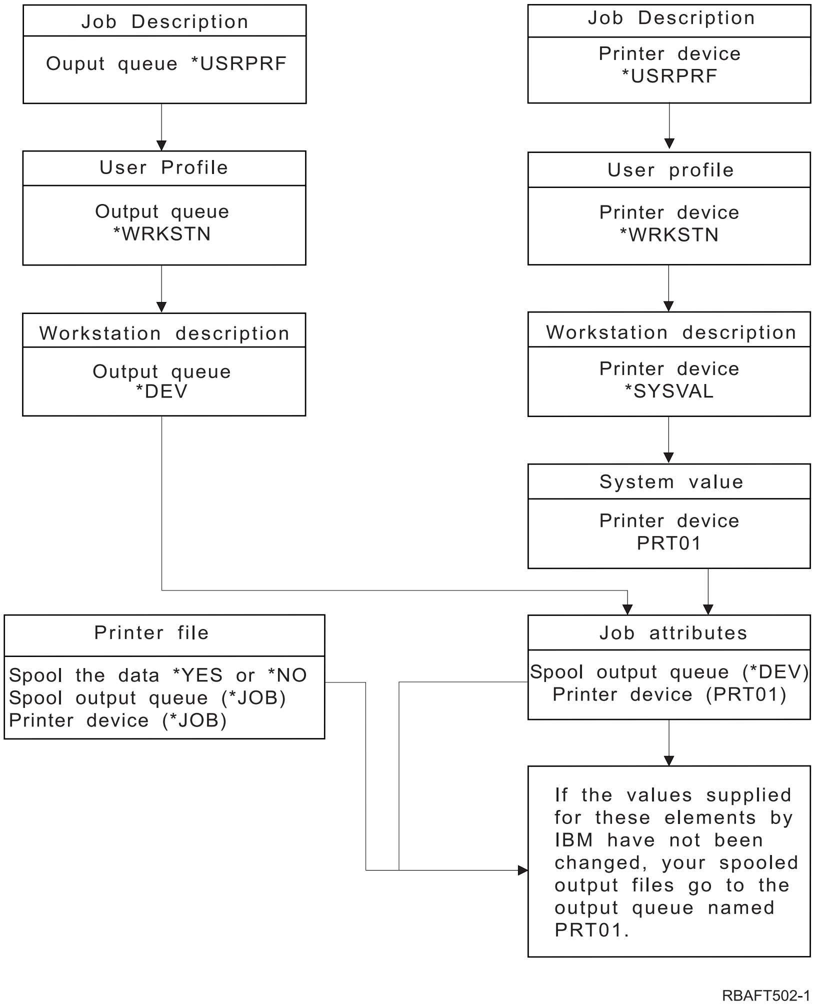
Assuming that the method used to start the job does not override the output queue or printer device values, the following order is followed:

1. The job initiation phase begins. The system looks at the following:
   a. The output queue value in the job description.
   b. The output queue value in the user profile.
   c. The output queue value in the workstation description.
   d. The printer device value in the job description.
   e. The printer device value in the user profile.
   f. The printer device value in the workstation description.
   g. 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.
2. The job run time phase begins. The system looks at the following:
   a. The output queue value in the printer file.
   b. The output queue value in the job attributes.
   c. The printer device value in the printer file.
   d. The printer device value in the job attributes.
   e. 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.



The CHGJOB CL command allows you to override the output queue and printer device values. The OVRPRTF CL command allows you to override many printer file values including the output queue and printer device values. See “Printer file overrides” on page 16 for more information on overriding printer file values.

Printing in a batch environment

When a user signs on to the server, an output queue (OUTQ) and a printer device (PRTDEV) 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.

Routing examples

The following demonstrate different examples of routing based on changing the values stored in the printing elements:

Example name	**Assumptions include**
"Example 1: Determine your output queue"	Default values
"Example 2: Determine your output queue" on page 50	OUTQ values set
"Example 3: Determine your output queue" on page 51	Current user has a group profile
"Example 4: Determine your output queue" on page 52	• Switch to alternate user profile • Printer file SPLFOWN set • Job description OUTQ set
"Example 5: Determine your output queue" on page 52	• Switch to alternate user profile • Data area • Job description OUTQ set
"Example 6: Determine your output queue" on page 53	• Switch to alternate user profile • Job description OUTQ set
"Example 7: Determine your output queue" on page 54	• Switch to alternate user profile • Group profile • Data area • Second data area
"Example 8: Determine your printer name" on page 55	• User profile PRTDEV set • Workstation description PRTDEV set
"Example 9: Determine your printer name" on page 56	Workstation description PRTDEV set
"Example 10: Determine your printer name when using batch" on page 57	• Job runs in batch • Default values
"Example 11: Determine your printer name when using batch" on page 57	• Job runs in batch • Submit Job PRTDEV set • Submit Job OUTQ set
"Example 12: Determine your printer name when using batch" on page 57	• Job runs in batch • Submit Job PRTDEV set • Submit Job OUTQ set

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

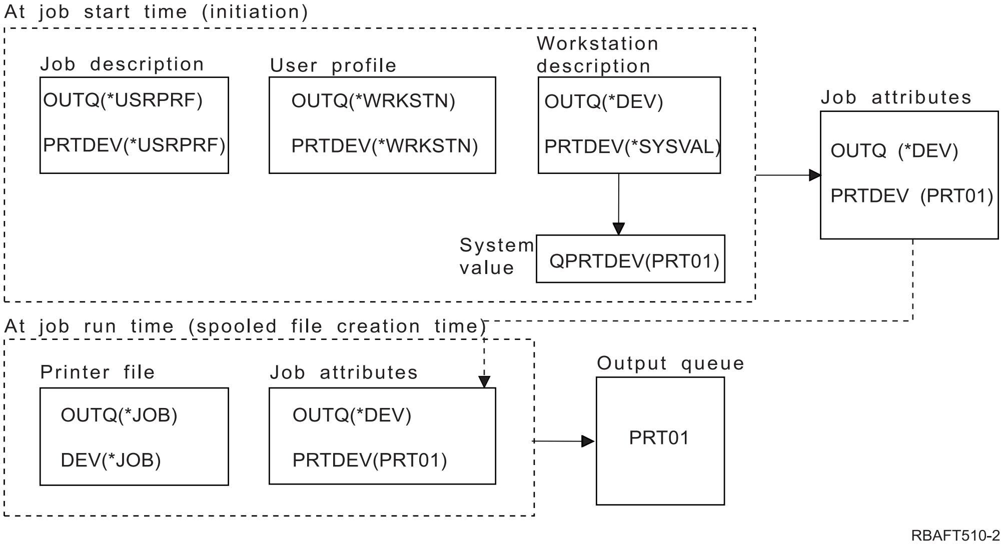
- "Self-test: Determining output queue and printer device" on page 58
- "Self-test answers" on page 59

Example 1: Determine 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.



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 a 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:

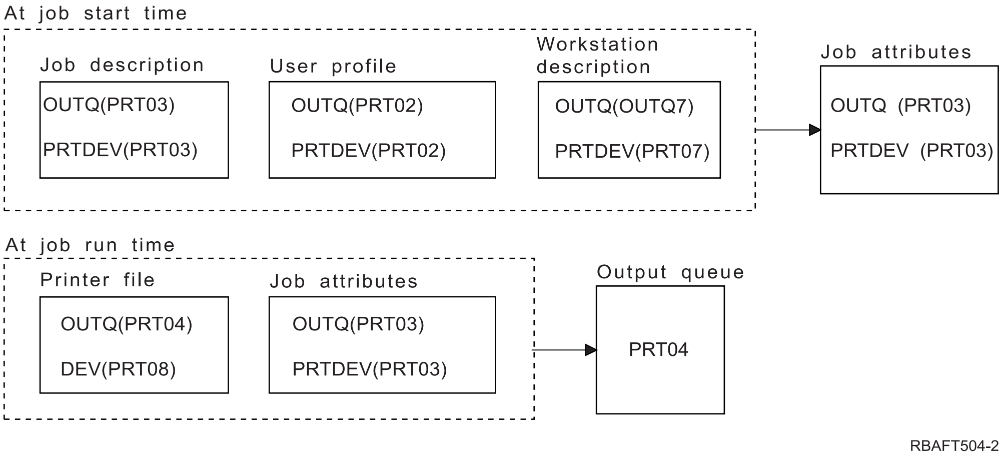
When a spooled file is to be created, the system looks at the OUTQ parameter in the printer file for the output queue name. In this example, that value is *JOB. This tells the system to look at the OUTQ attribute for the job. The OUTQ job attribute was set to *DEV during the job initiation stage. The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

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: Determine 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



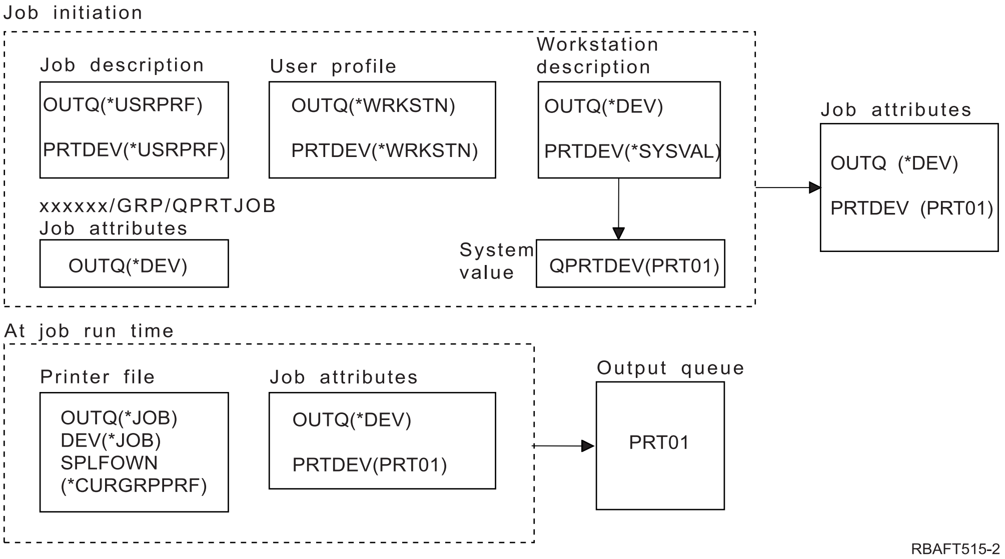
The output queue would be PRT04. The system found the output queue parameter value of PRT04 in the printer file rather than a value that would point it to your OUTQ job attribute.

Example 3: Determine your output queue: In the printer file, assume the following:

- 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.



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 takes 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.

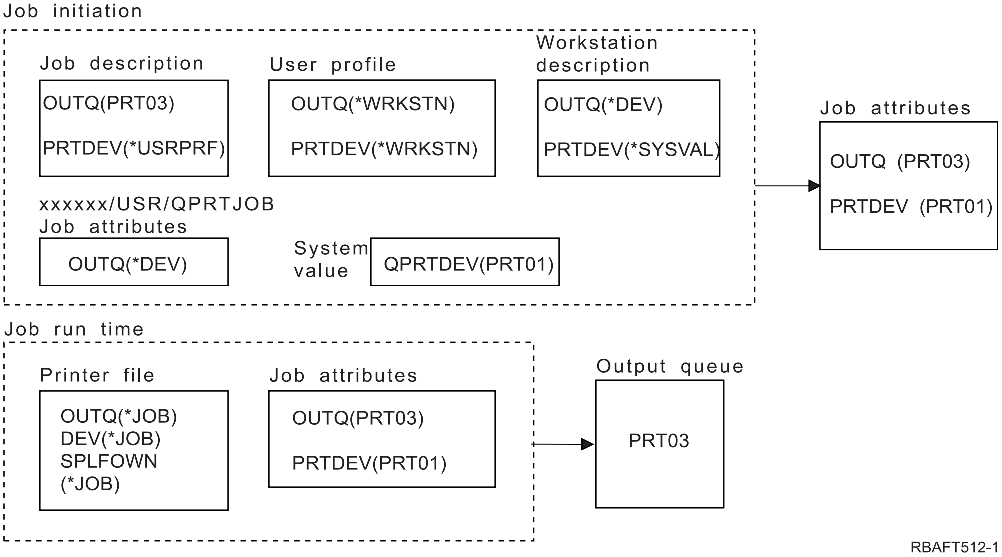
At job run time the following takes place:

> 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: Determine 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.



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.

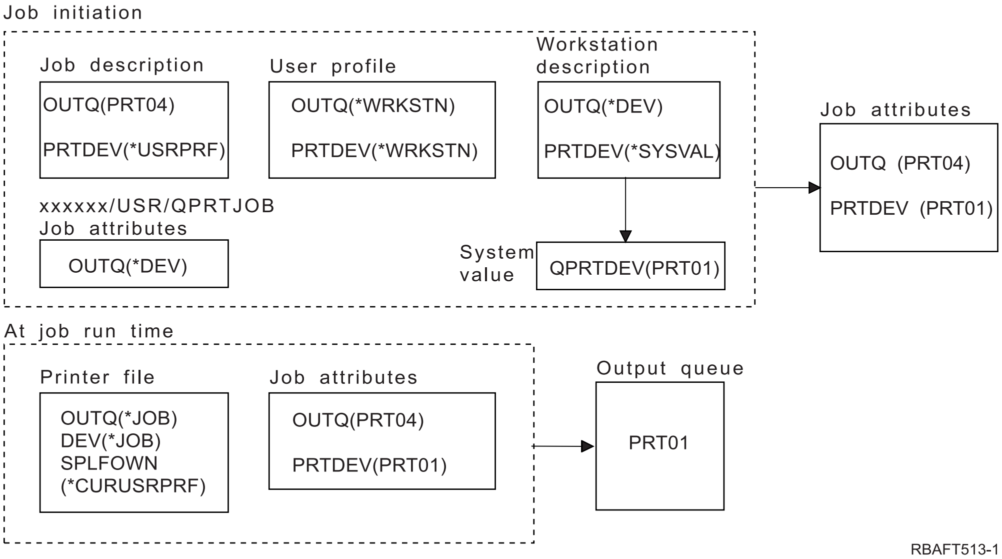
The system looks at the OUTQ parameter in the printer file for the output queue name. The value, in this example, *JOB, tells the system to look at the OUTQ job attribute. Because the SPFLOWN parameter is set to *JOB, the job attribute OUTQ of the current job is used. The value is PRT03. In this example,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

Example 5: Determine your output queue: In the printer file, assume the following:

- 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:

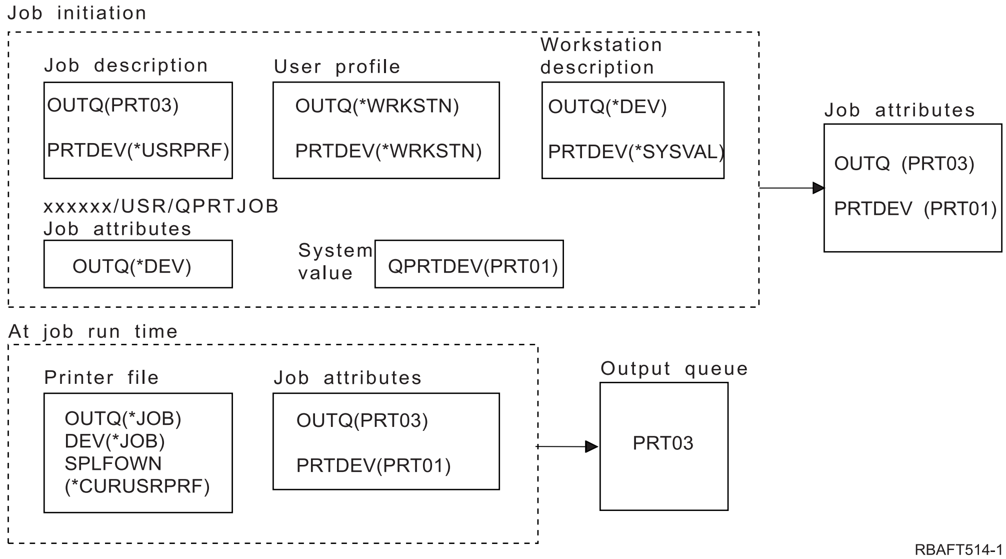
The system looks at the OUTQ parameter in the printer file for the output queue name. That value *JOB, tells the system to look at the OUTQ attribute of the job. There is a data area, QPRTJOB in QUSRSYS, that is owned by the user profile QSPL that has a logical value of false. Because of this data area, 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: Determine your output queue: In the printer file, assume the following:

- 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).

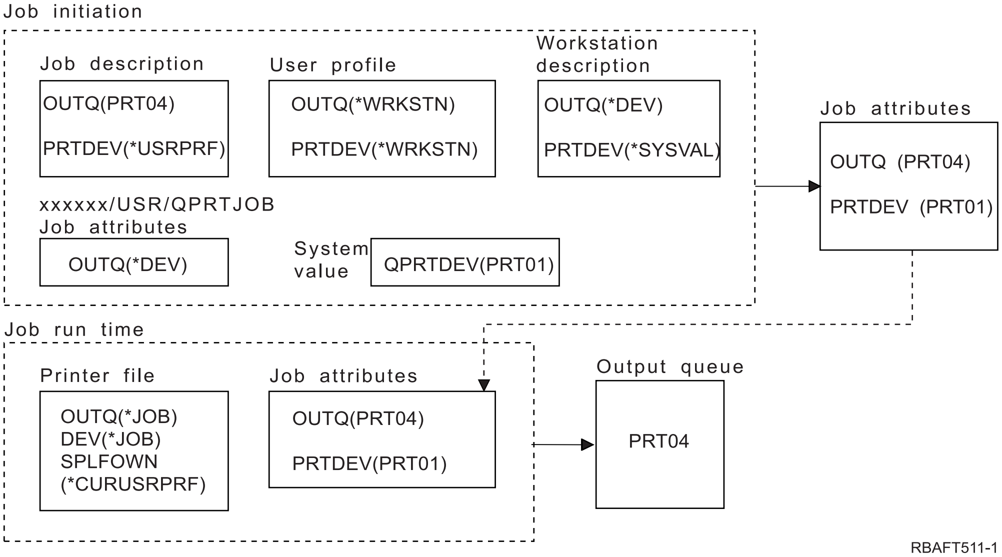
The system looks at the OUTQ parameter in the printer file for the output queue name. That value, *JOB, tells the system to look at the OUTQ job attribute. The system will look at the OUTQ job attribute of the current job, which is PRT03.

Example 7: Determine your output queue: In the printer file, assume the following:

- 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

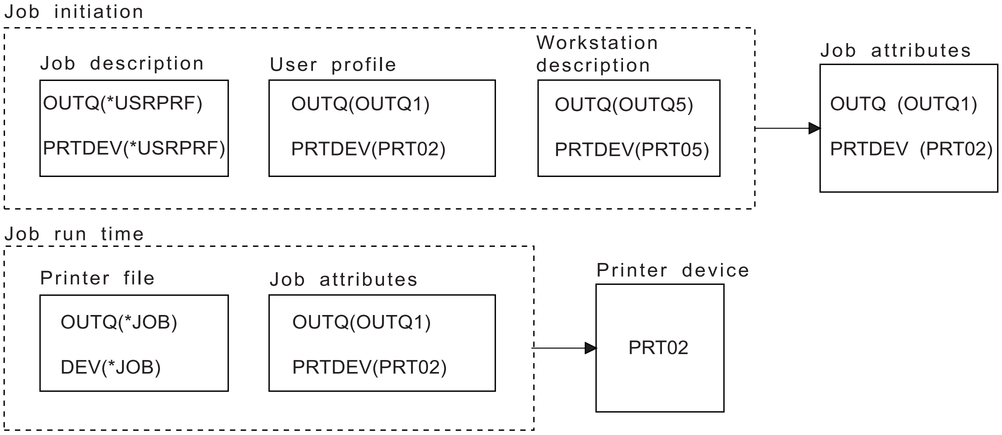
Note: The SPLFOWN parameter of the printer file is *CURUSRPRF, and the job has switched to alternate user profile USR. The spooled file will be created under job *xxxxxx*/USR/QPRTJOB (where *xxxxxx* is 000000-999999).

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: Determine 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 takes place:

Your printer device would be 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.

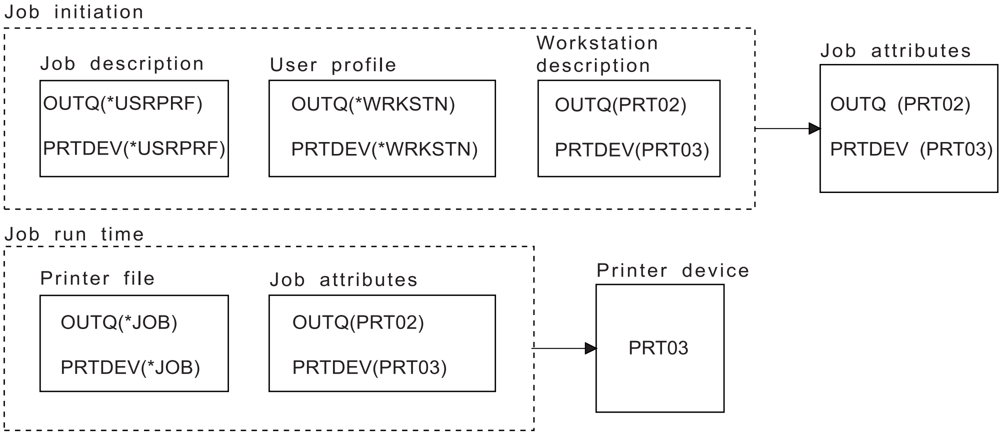
The job attribute PRTDEV value is PRT02.

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

Example 9: Determine 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 ithe job attributes.

At job run time, the following takes 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 would go directly to PRT03 for printing and no output queue would be 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: Determine your printer name when using batch:

Assume:

- 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 SBMJOB command is specified as *CURRENT.
- The printer device parameter on the SBMJOB 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 was used when printing, the output would go 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: Determine your printer name when using batch:

Assume:

- 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 SBMJOB command is specified as *USRPRF.
- The user profile has *WRKSTN as the value for the output queue parameter.
- The printer device parameter on the SBMJOB 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 was used when printing, the output would go to printer device PRT99.

Example 12: Determine your printer name when using batch:

Assume:

- 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 SBMJOB command is specified as *USRPRF.
- The user profile has *WRKSTN as the value for the output queue parameter.
- The printer device parameter on the SBMJOB 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.

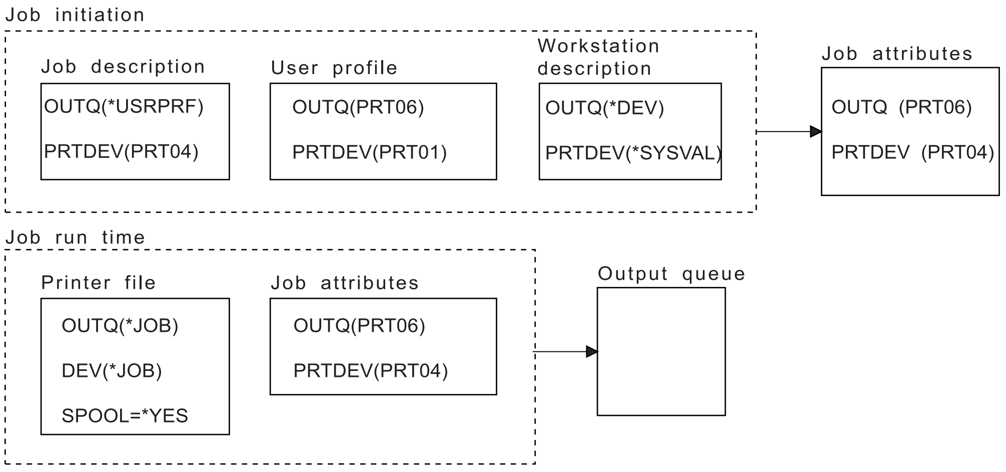
If no spooling was used when printing, the output would go to the printer defined as the system printer. This is the printer name assigned to the system value Default printer (QPRTDEV).

Self-test: Determining output queue and printer device: Below are diagrams 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 would be.

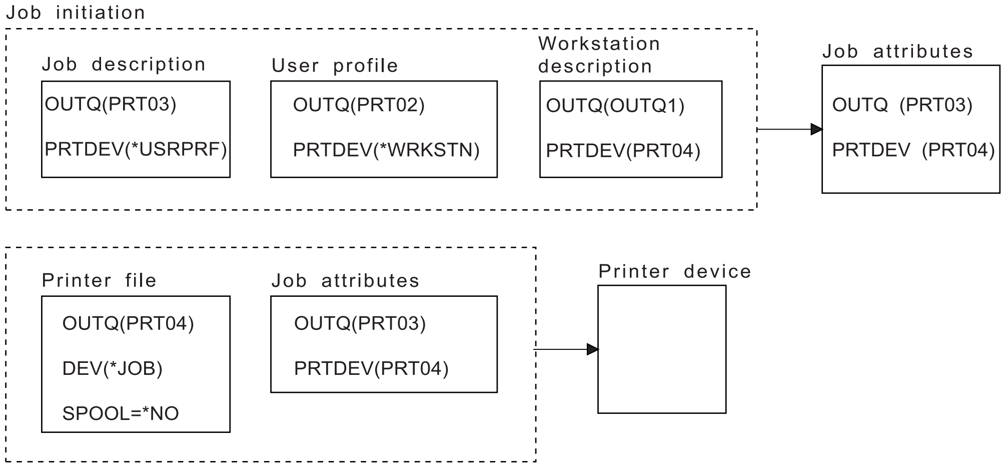
Note: Keep in mind the SPOOL parameter value when deciding on your answer.

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

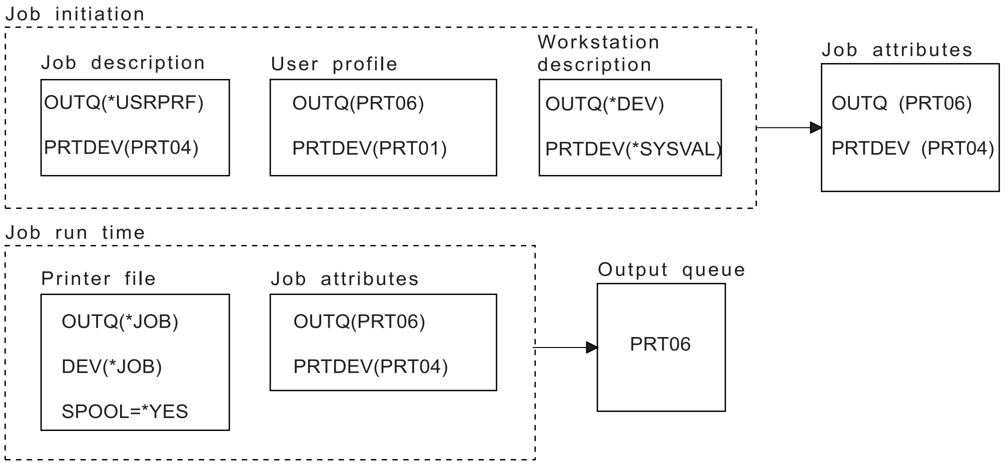


RBAFT507-1

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

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

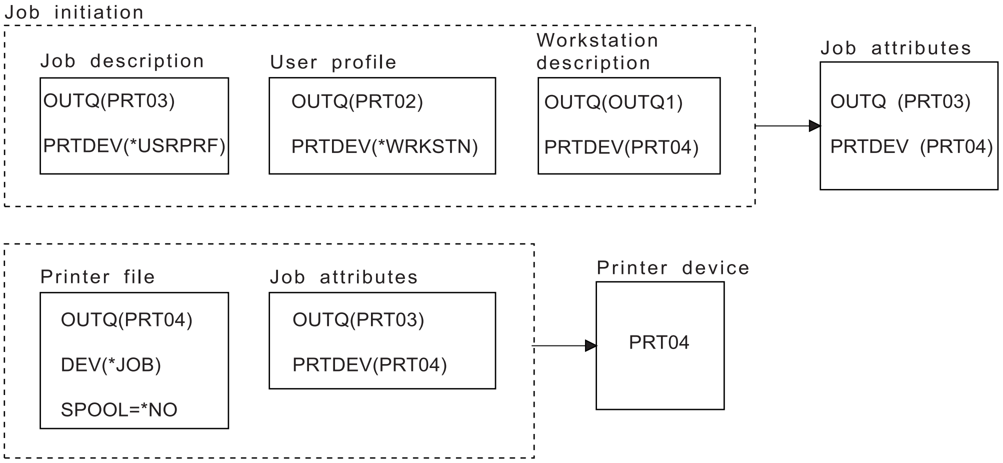
For the first diagram, the output queue name is PRT06.



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.

Remote system printing

Remote system printing allows spooled files created on an iSeries server 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).

See the following for more information:

“Benefits of using remote system printing”
Provides a description of the benefits of using remote system printing.

“How remote system printing works” on page 62
Provides a description of how remote printing works.

“Working with user print information” on page 62
Provides a description of user print information and how to change that information.

“Send and defer status of spooled output files” on page 63
Provides a description of send and defer status.

Benefits of using remote system printing

The benefits of using remote system printing include:

- 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 “Working with user print information” on page 62 for more detail on how to work with the display, retrieve, and change user print information commands.
- Send (SND) and Defer (DFR) status’ 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

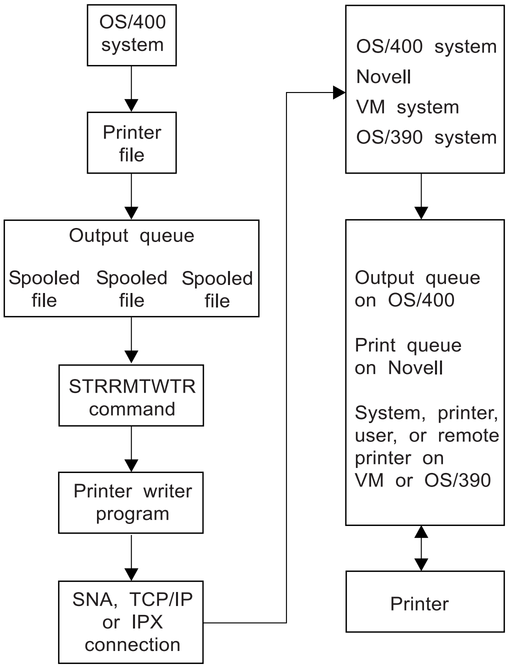
    The spooled output file has been deferred from being sent

  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.

See “Send and defer status of spooled output files” on page 63 for more detail on the send and defer attributes of a spooled file.

How remote system printing works

The following diagram illustrates the remote system printing function.



RBAFT518-1

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, we 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.

Working with 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 (RTVUSRPRTI) command.

User print information is not used when sending spooled output files to another iSeries server or an S/3X system. It is used only as information to pass to the VM/MVS bridge customer user 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 could 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 of spooled output files

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

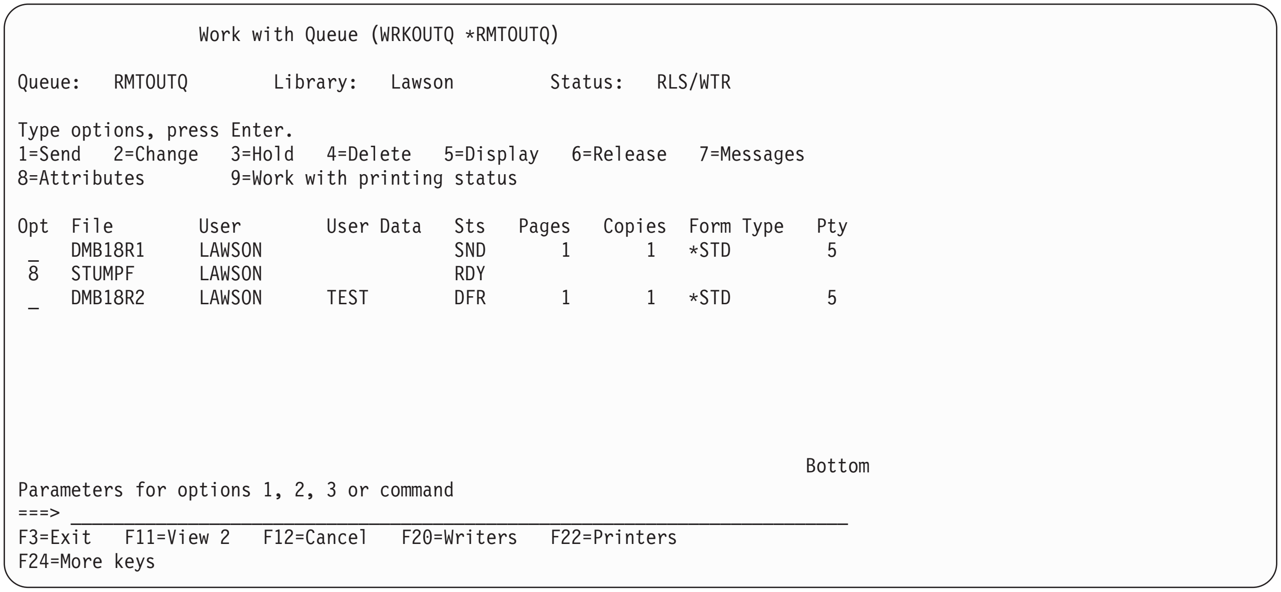
  When a spooled output file is being sent to a remote system, it has a status of SND. If the connection type is *SNA, the spooled output file may remain in SND status until a confirmation message from the remote system is received by the remote writer. At this time the spooled output file is deleted or 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 affect, 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 would print from 8:00 a.m. to 12:00 p.m. would be 40 pages. The largest spooled output file that would print from 12:00 p.m. to 12:30 p.m. would be 10 pages. The largest spooled output file that would print from 12:30 p.m. to 4:00 p.m. would be 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 could 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                  SND       1       1   *STD         5
 8   STUMPF      LAWSON                  RDY
 _   DMB18R2     LAWSON      TEST        DFR       1       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
```

Considerations

See the following for additional considerations:

- “Direct print considerations” on page 65
- “Open considerations” on page 65
- “Output considerations” on page 65
- “Close considerations” on page 65
- “First-character forms-control data considerations” on page 66
- “Printer font considerations” on page 67
- “Alternative character sets and code pages for printer output considerations” on page 70
- “Output field considerations” on page 71
- “Externally described printer file record format considerations” on page 72
- “Redirecting output considerations” on page 73

- “3812 and 3816 SCS printer considerations” on page 76
- “3835 printer considerations” on page 77
- “3912, 3916, and 4028 printer considerations” on page 77
- “Special printer file considerations for AFPDS” on page 77
- “Special DDS considerations for AFPDS” on page 78
- “Performance considerations” on page 80

Direct print considerations

When output data is written directly to the printer instead of output queue, the job is called a direct print job. When printing direct print job on an SCS printer, the file device type is changed to *SCS by OS/400. When printing on an IPDS printer that is configured with AFP(*NO), the file device type is changed to *IPDS by OS/400. If the print job has file device type *AFPDS, *USERASCII, *LINE, or *AFPDSLINE, the direct print job is not supported.

Open considerations

The following considerations apply to opening printer files for spooled output:

- The output queue should be created for the type of output your program produces so that system operator intervention can be kept to a minimum while a printer writer is producing output. You should consider the following 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, it is dependent on things such as:
  - 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

The following 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 lets you 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 containing 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 will not 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 itself. You can do this 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 DDS on the same file.)

For information about the machine data, see the Printer Device Programming manual .

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

Any other character in position 1 of a record defaults to 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 next record processed has a channel value forms-control number that is the same value as the line number the printer is on, the printer advances to that value (line number) on the next page. In the example in the paragraph above, if the printer had been on line 20 already, the printer would then have advanced 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.

Printer font considerations

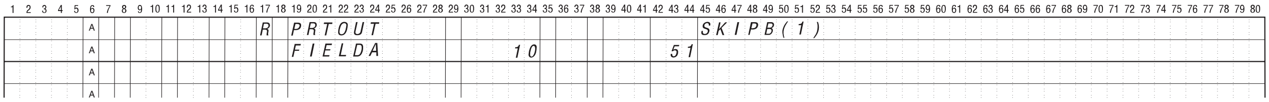
Characters per inch versus font

Most SCS printers use the CPI parameter on the CRTPRTF, CHGPRTF, or 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 converting the page size (PAGESIZE) 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 change in the appearance of the printed output. For example, a printer file that has FONT(222), Gothic font with 15 pitch, and CPI(15) could print on a 3812, 4224, or 5219 Printer (which uses the FONT parameter) or also could print on a 4214, 4230, 4234, 4247, 5224, or 5225 Printer (which supports 15 characters per inch). If this print job were directed to a printer that only supports 10 characters per inch, then printer file redirection would be used.

For more information about redirecting spooled files to SCS printers, go to “Redirecting output considerations” on page 73.

When using an SCS externally described printer file, normally the value specified in the CPI parameter is 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 would be 50 blanks to the left of FIELDA (50 blanks at 10 characters per inch is 5 inches).



```
     A          R PRTOUT                    SKIPB(1)
     A            FIELDA        10        51
     A
     A
```

RSLH114-0

For the IPDS externally described printer files, the pitch implied by the FONT parameter on the CRTPRTF, CHGPRTF, and OVRPRTF commands is used instead of the CPI parameter or the value of the FONT DDS keyword to determine the starting column of fields on a printed page. In the example above, if

CPI(10) and FONT(087) had been specified, and the printer specified was one that supports fonts, then FIELDA specified to start in column 51 would be 50 blanks at 12 characters per inch (implied CPI value for font 087) or 50/12 inches, which is 4.167 inches in from the left margin of the paper. The default for FONT is (*CPI). When *CPI is selected, the OS/400 program automatically selects a font of the pitch specified on the CPI parameter.

Proportionally spaced and typographic fonts

All IPDS printers support proportionally spaced fonts. For proportionally spaced fonts, characters vary in width depending on the character being printed (for example, **i** is a narrow character, and **W** is a wide character). See the Printer Device Programming manual for a list of all supported fonts. The implied characters-per-inch column in this table lists the value of the width of a blank character for the font selected.

When using proportionally spaced fonts, fold and truncation (FOLD parameter) may not work as intended. This is because the system does not keep track of the width of each individual character.

The following printers also support typographic fonts: the 3812, 3130, 3160, 3816, 3820, 3825, 3827, 3829, 3831, 3835, 3900, 3916, 3930, 3935, 4028, 4312, 4317, 4324, InfoPrint 3000, and the InfoPrint 4000. You can specify a typographic font by indicating point size (height of font). A point measures 1/72 of an inch. An 8-point font would be 1/9 of an inch high, and a 24-point font would be 1/3 of an inch high. When using fonts that contain tall characters, it may be necessary to double or triple space, to avoid having lines overlap when printing the page.

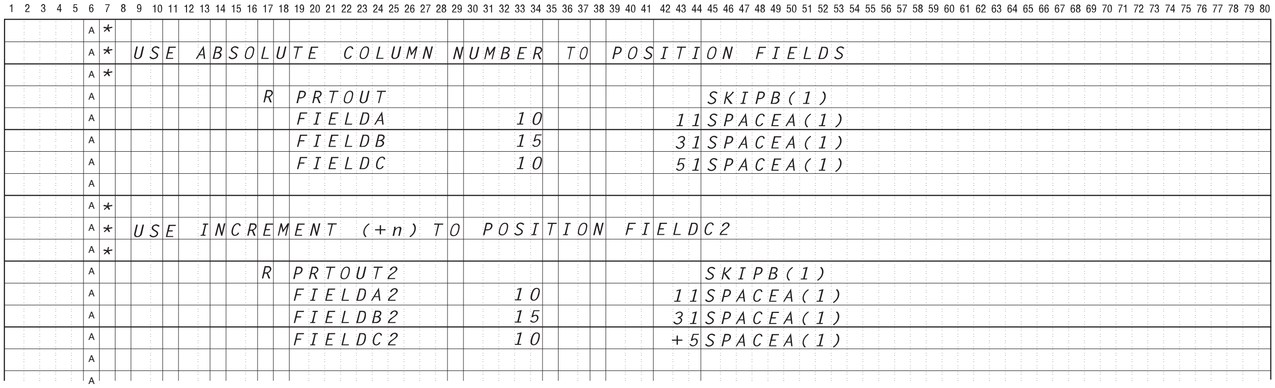
Because proportionally spaced and typographic fonts have characters of variable widths, care should be taken with the use of underlining and overstriking. The highlighting or underlining method of printing a line with a space after (SPACEA) value of 0 followed by printing another line may not work correctly.

Proportionally spaced and typographic fonts may be specified on the CRTPRTF, CHGPRTF, or OVRPRTF command when using an externally described printer file. The point size may be specified for typographic fonts. The point size is ignored for fonts that are not typographic. As noted above, the implied characters-per-inch value of the font identifier specified on the FONT parameter is used to position fields on a printed page. The same rule for positioning fields on a printed page is used with proportionally spaced and typographic fonts. The width of a blank character is used to position fields on a page. Output should be tested to see that using externally described printer files with proportionally spaced fonts produces satisfactory results because overprinting and gaps can occur in the output. 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.

Note: The amount of printed space for a field varies depending on which characters are in a field. Enough space should be left between fields to allow for the widest characters (uppercase characters) expected in that field.

In the previous example, if CPI(10) and FONT(1351) had been specified, and the printer used was a 3812 Printer, then FIELDA specified to start in column 51 would be 50 blanks at 17.14 characters per inch (implied CPI value for font 1351) or 50/17.14 inches, which is 2.975 inches from the left margin of the paper.

In this example, FIELDA would start 2.975 inches from the left margin of the paper regardless of how many fields were defined to its left. When using an increment value (+n) instead of a column number (positions 42 through 44 in DDS specification), fields are positioned the same for proportionally spaced fonts as they are for fixed pitch fonts. That is, the field is positioned based on the width of the blank for the specified font. The following example illustrates that using either absolute column numbers or relative increment numbers (+n) will supply the same result.



```
     A*
     A*  USE ABSOLUTE COLUMN NUMBER TO POSITION FIELDS
     A*
     A          R PRTOUT                    SKIPB(1)
     A            FIELDA        10        11SPACEA(1)
     A            FIELDB        15        31SPACEA(1)
     A            FIELDC        10        51SPACEA(1)
     A
     A*
     A*  USE INCREMENT (+n) TO POSITION FIELDC2
     A*
     A          R PRTOUT2                   SKIPB(1)
     A            FIELDA2       10        11SPACEA(1)
     A            FIELDB2       15        31SPACEA(1)
     A            FIELDC2       10        +5SPACEA(1)
     A
     A
```

RSLH115-3

Font substitution

If the font identifier specified on the FONT parameter is not supported by the printer being used, printer data management selects a substitute font that is supported by the printer (if possible). A substitute font is at the same pitch or a higher pitch whenever possible to ensure that as much data as possible fits on the printed page. A complete list of fonts supported and the substitute font selected for each printer that supports the FONT parameter is shown in the Printer Device Programming manual .

If a substitute font cannot be used, spool redirection is done. An inquiry message is sent to the message queue associated with the device or printer writer. The inquiry message gives you the option of holding or printing the file. If the print option is used, then the spooled file is reformatted with the print attributes of file QPSPLPRT. Output may not look as intended. Refer to the printer's reference manual for information on what conditions cause the substitute font to not be used.

For the SCS 3812, 3816 and 5219 Printers, font substitution can be made only at the file level. For font changes made in the document, font substitution is not done. In this case, spool redirection (described previously) is used.

IPDS printer FONT parameter considerations

When a printer has AFP(*YES) specified in the printer device description and the DEVTYPE specified for the printer file being used is *AFPDS, the FNTCHRSET, CDEPGE, and CDEFNT printer file parameters may be used to select a font resource to be downloaded to the printer. This applies to all IPDS printers except the 4224, 4230, 4234, 4247, and 64xx. If these parameters are not used, then the value specified in the FONT printer file parameter is used.

When FONT(*DEVD) is specified on the CRTPRTF, CHGPRTF, and OVRPRTF commands, the following limitations are imposed:

- Bar codes are positioned on the page assuming a 10-pitch font is specified in the device description.
- When using a combination of bar code (BARCODE), page rotate from a printer file or DDS (PAGRTT), and character size (CHRSIZ) parameters in an externally described printer file, unpredictable results may occur. This is because the printer device FONT parameter value is not known when the spooled file is created.
- The data stream created may be longer than if a specific font were selected. This means the spooled file may take more storage in the output queue. Fields are positioned with spaces (hex 40) between them instead of using commands 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 may 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 *DEVD 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 recommended 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.

Alternative character sets and code pages for printer output considerations

Character sets are used with code pages to determine how each character will appear 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 may 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 section, these characters are called **extended alphabetics**.

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 could have been entered on a display device that can handle the character or could have been 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 could 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:

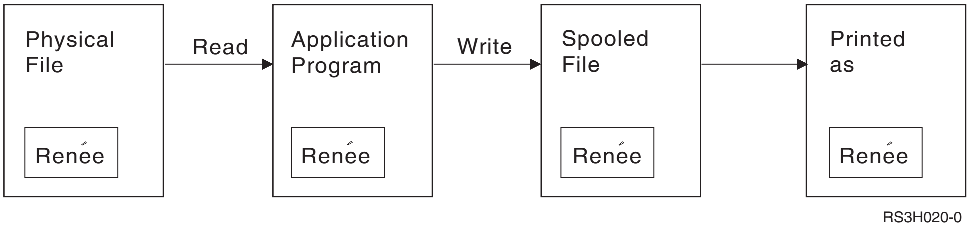
- With an explicit value specified for the CHRID parameter, the printer interprets the data as if the data were in the character set and code page specified.
- With CHRID(*SYSVAL) specified, the printer file takes the value specified in the Graphic character set/Code page (QCHRID) system value when the 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. Go to the Printer Device Programming manual 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 alphabetics 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 containing 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 alphabetics. 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 had been selected, hex C0 (é) would have been 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

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:

- 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 may 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 DDS Reference in the Programming topic.

User-defined edit codes

You can also define five edit codes to provide more editing function than is available with the OS/400 edit codes, and to handle common editing functions that would otherwise require the use of an edit word. These are called user-defined edit codes. For example, you may 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 DDS or a high-level language program by number (5, 6, 7, 8, or 9).

These edit codes are created by using 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 DDS Reference in the Programming topic for more information about using these edit descriptions.

Before using any of the user-defined edit codes, you should check its contents on your system, since it may 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 using 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).

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 the 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 is specified (LVLCHK parameter), 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 may not have to recompile your program. If you do not have to recompile your program, you should specify LVLCHK(*NO) for the file (CHGPRTF or 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 will require 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 occurs:

- If the printer file specifies a characters-per-inch value 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 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 change if any of the spooled file attributes are not supported by the printer device. For example, 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 describes the actions taken when a spooled file is redirected to an SCS printer and cannot be printed without change (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 any of the following 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 may be unpredictable because the file is printed using 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:

  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 not supported by the printer, an inquiry message is sent to the message queue of the writer with the option to:

  - 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 option to:
  - 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 may contain print controls that are not supported by the 5219 or 3812 Printers. These controls may 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 may 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 describes the actions that 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 may 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:

  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: 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 may 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 is 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 may 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:
  - 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 iSeries server 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 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.

Once 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 will not print.

The iSeries server 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 will print. The iSeries server adjusts the starting 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 may have 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 iSeries server does not adjust the position of printed text for this printer. When directing output from printing on a 3835 Model 1 to a Model 2, the difference in how the no-print border is used must be considered.

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 will not print.

If the 3912, 3916, or 4028 is configured AFP(*NO), you may have 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 AFP(*YES), the iSeries server 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 will print. The iSeries server adjusts the starting 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 may have 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 AFPDS

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 signalled 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 signalled 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 AFPDS 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(*DEVD) or an externally described printer file is used, the Graphic character set/Code page (QCHRID) system value is substituted.
- If FONT(*DEVD) is used, font 11 is substituted.
- If FORMFEED(*DEVD) is used, drawer 1 is substituted.
- If PAGRTT(*DEVD), PAGRTT(*AUTO), or PAGRTT(*COR) is used, page rotation of 0 is substituted.

Notes:

1. On iSeries servers, offset stacking of printed output is used. As 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 iSeries server 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 may be issued.
2. Use of the DDS DRAWER and PAGRTT keywords cause the OS/400 to generate an AFPDS datastream 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 DDS considerations for AFPDS

To use Advanced Function Presentation (AFP) support on the iSeries server to print on IPDS printers, you must install Print Services Facility for OS/400 (PSF/400).

See the iSeries Guide to Output for information on when PSF/400 is required. If you have additional questions about PSF/400, contact your IBM representative.

The following is a list of DDS keywords that 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. For more detailed information about DDS keywords, see DDS Reference: Printer Files in the Programming topic.

- 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
- UNISCRIPT
- ZFOLD

Performance considerations

The following 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, by putting several lines of text within a record instead of each line as a separate record, system overhead involved with the processing of each record is reduced.
- When coding the DDS 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 may be noticeable.
- For externally described printer files, specify a specific font or FONT(*CPI) on the CRTPRTF, CHGPRTF, or OVRPRTF command instead of FONT(*DEVD). 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 printing 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 may occur before the syntax of the entire spooled file is checked. A data stream error may 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 print while convert (PRTCVT) parameter of the printer device description to *NO.

Plan for printing

The iSeries servers offer a wide range of printing solutions, ranging from high-function IPDS printers supporting bar-codes and complex document formatting on high performance printers, to simple document printing using a directly attached printer. Clearly understanding your organizations requirements, the type of printed output you need, and knowing the capabilities of the hardware you currently have or intend to purchase, are all critical components of configuring an iSeries printing solution. After you have determined your requirements, you should decide whether you will use IPDS or Host Print Transform to convert and send print data. Generally, IPDS (usually used with PSF/400) offers higher performance and additional functions, but at a higher price. Refer to the following resources for more information about determining what type of printing solution will best meet your business needs.

“Print protocols” on page 81
This topic describes some of the common options for iSeries printing, and compares their features and requirements.

Printing Solutions for iSeries
(www.printers.ibm.com/R5PSC.NSF/Web/as400overview)
Use this IBM Web site to help identify which printing solution best meets your needs.

IBM printer selector
(www.printers.ibm.com/R5PSC.NSF/Web/pselect)
This Web-based tool from ibm.com® can help you select the best printing hardware for your requirements.

Support for Various ASCII Printers
One of the most common causes of problems in a printing environment is hardware compatibility. Refer to this IBM Knowledge Base document (document number 17690939 at www.ibm.com/eserver/iseries/support/s_dir/slkbase.NSF) to find out which printers will support the print protocols you require.

Print protocols

The iSeries server offers several different print protocols to meet a variety of configuration and performance requirements. The following information introduces the capabilities and requirements of some of the common printing options.

SNMP

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. This 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 iSeries to efficiently share a printer with other users. Additionally, since SNMP supports both Post Script and PJL 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. OS/400 V4R5 or higher supports the SNMP print driver.

PJL

Printer Job Language (PJL) printing solutions also provide problem handling and status information during the print process by using bi-directional communication between the printer and print server over a single TCP/IP port. PJL printing will allow you to share a printer between the iSeries and other network users, but since the iSeries keeps communicating with the printer until the iSeries output queue is empty, resource sharing is more limited than with other either SNMP or LPR/LPD.

PJL printing through an iSeries server requires that the printer and printer adapter or network adapter support Printer Control Language level 5e. Also, the cabling, printer and network adapter or print server must be capable and configured for bidirectional communication. OS/400 version V4R1 (V3R7 with PTFs) or higher support PJL printing.

IPP

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 should be supported by a variety of printers and network adapters. This protocol provides the advantages of printing with 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 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.

LPR/LPD

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 that 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.

IPDS (PSF/400)

Intelligent Printer Data Stream IPDS with Print Services Facility for OS/400 (PSF/400) offers industry-leading printing performance and functions, and can take advantage of the iSeries Advanced Function Presentation print resources. While PSF/400 solutions offer excellent performance, they are also more expensive than other iSeries printing options.

Scenarios: Select a print protocol

There are many options available when selecting a print protocol. The one you should choose will depend on your business requirements, the current hardware and skills available, and the resources your organization is willing to commit. Also, since different protocols support different functions, and require different hardware and network architecture, it is import to carefully consider what your current hardware will support, and what specific functions you require in a printing solution.

The following scenarios provide a brief business case illustrating the advantages of four common printing methods, state the requirements for using each protocol, and then link to configuration examples and procedures.

"Scenario: SNMP printer configuration" on page 83
Share printing resources over a LAN with iSeries applications and user workstations. SNMP provides good support for printing functions like page ranges and allows for basic problem handling and resource sharing functions.

"Scenario: PJL printer configuration" on page 84
Share printing resources over a LAN with iSeries applications and user workstations. PJL uses Printer Control Language commands to format printed output, which supports a broad range of printing functions. PJL print protocols also support some problem handling and resource sharing features.

"Scenario: IPP printing configuration" on page 85
Print over the Internet using the iSeries as an IPP server for remote printing, and to share printing resources on your corporate intranet.

"Scenario: IPDS with PSF/400 printer configuration" on page 86
Provide detailed error recovery and resource sharing along with industry-leading support for print functions using IPDS printers with Print Services Facility for OS/400 (PSF/400) software from IBM.

“Scenario: LPR/LPD printing configuration” on page 86
Either start line printer daemon (LPD) on the iSeries server to handle print requests from attached clients, or use line printer requester (LPR or remote writer) to send print jobs from a remote output queue to another printer or print server on your network. This protocol has extensive hardware support, but provides only minimal support for error handling and resource sharing features.

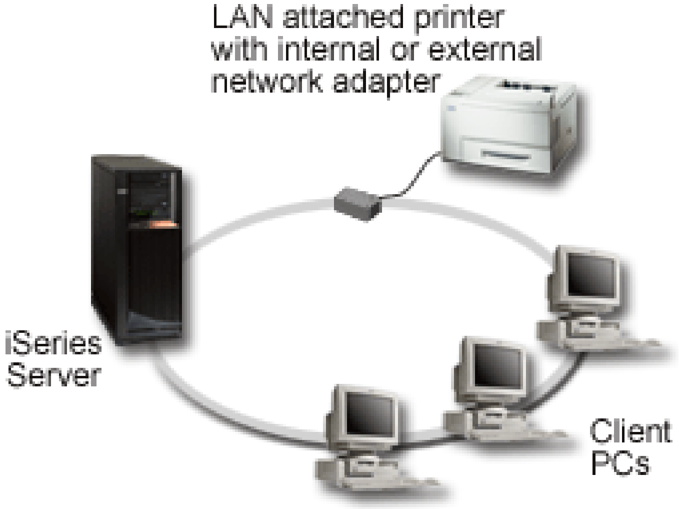
Scenario: SNMP printer configuration

Situation

As an administrator for your company’s network, you need to provide print services to several dozen client PCs by allowing them to share a common printer attached to the office LAN. Since this printer will have many users, you require a solution with reliable sharing capabilities and error handling. With these needs in mind, you have purchased printing hardware that supports SNMP printing.

Details

To provide printing resources to your client PCs over a LAN you will use your iSeries as a print server that manages print jobs and sends them to an SNMP-compatible printer attached to the LAN through a network adapter. Communication between the iSeries and the printer will be managed with the SNMP protocol. SNMP printing uses two distinct TCP/IP ports for communication, one port for sending the print job, and another for tracking status and errors.



- SNMP requires that the print server (or network adapter) and printer each support the Host Resource Management Information Base (Host Resource MIB), and recommends support for Printer Management Information Base (Printer MIB). This SNMP requirement is documented in RFC 1514 (Host Resource MIB) and RFC 1759 (Printer MIB).
  - OS/400 V4R5 and later support SNMP printing.
  - To find out if your printer and network adapter support the Host Resource and Printer MIBs, refer the manufacturer’s documentation. The IBM Knowledge Base document Support for “Various ASCII Printers”, 17690939 also maintains a list of protocols supported by common printers.
- The printer is attached to the first parallel port of the multi-port network adapter, and is the only SNMP-enabled device attached.
- SNMP allows for simple security by defining communities of either public or private access to a given resource. The public community name must have read authority.

Configuration steps

1. Define the printer's device description in OS/400 using the CRTDEVPRT command. For detailed instructions on how to complete this description, including recommended values for several printer types and an example device description, refer to the IBM Knowledge Base document "Configuring a *LAN 3812 SNMP Device Description", 19932815.
2. Configure the printer and print server (or network adapter). Refer to the manufacturer's documentation for more information. Also, the IBM Knowledge Base document "Configuration Settings and Error Messages for *LAN 3812 SNMP Device Descriptions", 19935220 lists suggested settings.
3. If necessary, configure your libraries, queues and spooling in OS/400. For more information refer to the Work Management topic.

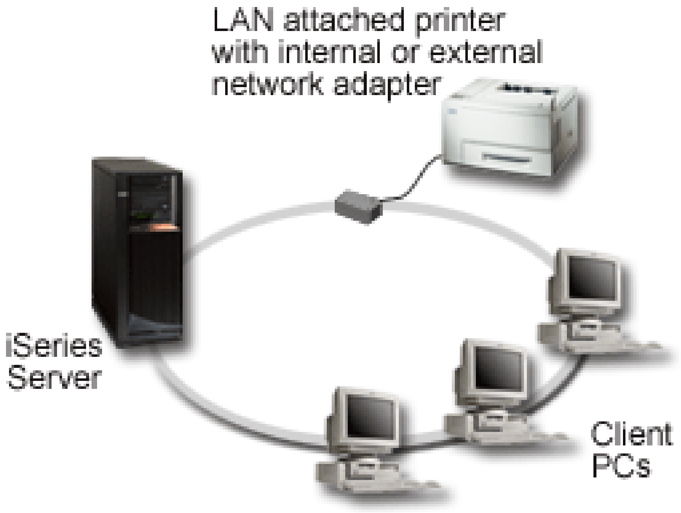
Scenario: PJL printer configuration

Situation

As an administrator for you company's network, you need to provide print services to several dozen client PCs by allowing them to share a common printer attached to the office LAN. You need a solution that provides some job accounting information to allow for error handling, and would like to use your current hardware that supports Printer Job Language (PJL).

Details

To provide printing resources to your client PCs over a LAN you will use a network adapter and a LAN attached printer. Communication between the iSeries and the printer will be managed using PJL. This provides applications with a significant level of control for how print output should be formatted, even allowing for distinctions in paper handling like sorting and stapling. Communication between the print server and the printer is bi-directional, meaning that information about errors that occur during printing will be available.



- PJL requires that the print server (or network adapter), and printer each support PCL level 5e.
  - OS/400 V4R1 and above support PCL level 5e. Release V3R7 requires PTFs for this support.
  - To find out if your printer and network adapter support PCL level 5e, refer to the manufacturer's documentation. The IBM Knowledge Base document "Support for Various ASCII Printers", 17690939 also maintains a list of protocols supported by common printers.
- The network adapter must provide bidirectional communication with the printer.
- If the network adapter is external, its connection with the printer must use IEEE 1284 bidirectional printer cable.

Configuration steps

1. Define the printer's device description in OS/400 using the CRTDEVPRT command. For detailed instructions on how to complete this description, including recommended values for several printer types and an example device description, refer to the IBM Knowledge Base document "Configuring a *LAN 3812 PJL Device Description",8695149.
2. Configure the printer and print server (or network adapter). Refer to the manufacturer's documentation for more information. Also, the IBM Knowledge Base document "Configuration Settings and Error Messages for *LAN 3812 PJL Device Descriptions", 14461435 lists suggested settings.
3. If necessary, configure your libraries, queues, and spooling in OS/400. For more information, refer to the Work Management topic.

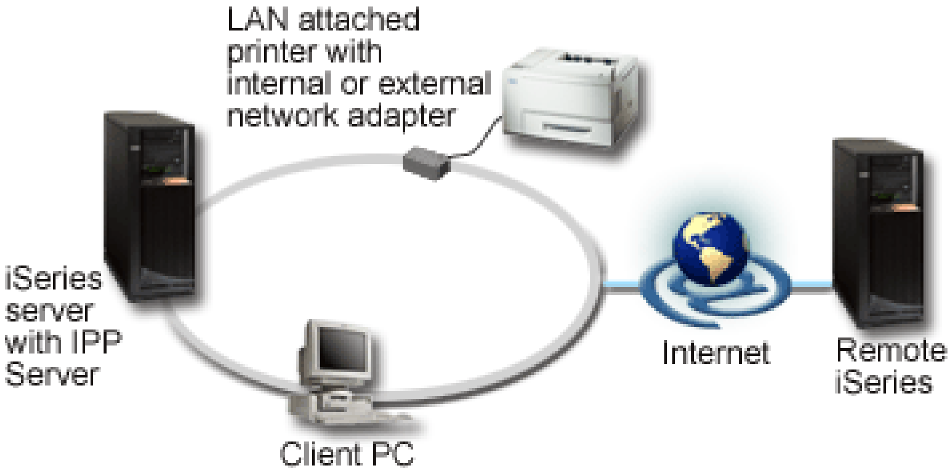
Scenario: IPP printing configuration

Situation

You need to provide printing services across several existing LANs, which consist of both PCs and several servers, and must include your sales staff at several remote offices across the country. Because you are printing solution ties in existing hardware from a several LANs, you require a printing solution that will be compatible with a wide variety of printers and network adapters. Moreover, you will have to configure printers and work with print jobs and errors at several remote sites. Also, since some of the communication involves confidential information, you require a printing solution that provides secure communication. With these needs in mind, you decide that printing over your corporate intranet will be the most versatile option, so you need to configure your network to support IPP printing.

Details

IPP printing requires an IPP server handling requests from a variety of IPP client devices. Since this communication uses HTTP, print requests can travel over your company's intranet or over the Internet. IPP uses SSL to encrypt communication and provide secure connections. The IPP driver requires OS/400 V5R2 or later. Also, IPP requires an HTTP server, Java, and DCM (for SSL encryption).


LAN attached printer with internal or external network adapter
iSeries server with IPP Server
Internet
Remote iSeries
Client PC

Configuration steps

1. Set up the IPP printer device description on the iSeries. Refer to "Recommended settings for IPP printer device description" on page 88 for recommended settings.
2. Configure the IPP printers and network adapters.
3. Configure the IPP server from The iSeries Tasks Page by entering the printer URL in a Web browser. For example: `http://mysystem:2001` (or `http://mysystem:2010` for an SSL connection).

Scenario: IPDS with PSF/400 printer configuration

Situation

Your company requires a print solution that handles complex forms that include barcodes and other graphics for printing shipping and receiving requests in your warehouse. Since the success and normal operation of your business depends on the ability to print reliably and to generate forms rapidly, you decide to invest in a high-end printing solution.

Details

Print Services Facility for OS/400 (PSF/400) with IPDS printers provides industry leading support and performance for complex and demanding printing environments. PSF/400 is included in OS/400, but must be activated before it is used. Fees are based on the performance requirements for your printing solution.

Configuration steps

For information about purchasing and configuring PSF/400, refer to PSF/400 home page .

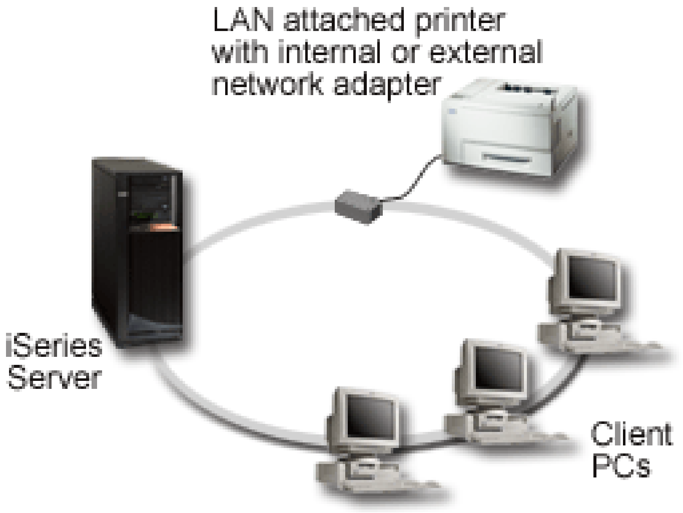
Scenario: LPR/LPD printing configuration

Situation

You need a simple way to print iSeries reports and status information, and since you will only be printing occasionally, you would like to use an older printer that is already in use and attached to the corporate LAN, and use a protocol that supports some resource sharing. You do not need the function supported with Host print transform or IPDS, and since your printing needs are minimal, you do not need the job accounting and tracking information provided by other protocols. With these needs in mind, you decide that LPR/LPD printing will meet your needs, without requiring you to purchase new hardware or implement a new printing solution.

Details

You can set up a remote output queue to automate line printer requester (LPR or remote writer) on the iSeries server. This sends printer output to another printer or print server that is using a line printer daemon (LPD).



Although the hardware support for printing through a remote output queue using LPR/LPD is extensive, not all printers will support it. You should verify that the printer you would like to use supports this printing method. The IBM Knowledge Base document 17690939 ″Support for Various ASCII Printers″ maintains a list of protocols supported by common printers.

Configuration steps

1. Start LPD on the print server, PC, or iSeries that you are printing to. If you are printing to another iSeries, you can start LPD with iSeries Navigator:
   a. Expand **Network –> Servers –> TCP/IP**
   b. Right-click on **LPD** from the list of servers, and select **Start**.
2. Set up a remote output queue for the iSeries server. Refer to the IBM knowledge base document "Configuring a Remote Output Queue (RMTOUTQ)", 8983237 for instructions and additional information.
3. Vary on the printer.
   a. Enter WRKCFGSTS *DEV command. The Work with Configuration Status screen will display a list of devices.
   b. Enter a 1 next to the printer printer device description to vary on the printer.
4. Enter STRRMTWTR to start the remote writer.

Configure printing

Configuring iSeries 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 Knowledge Base document "Support for Various ASCII Printers", 17690939 maintains a list of known protocols supported by some common printers.

See the following for more information on how to configure printing:

"Configure line printer requester/line printer daemon (LPR/LPD) printing" on page 88
Provides information about configuring line printer requester/line printer daemon (LPR/LPD) printing.

"Configure PJL, SNMP, or IPP printers" on page 88
Provides information about configuring printer job language (PJL), simple network management protocol (SNMP), or Internet printing protocol (IPP) printers.

"Configure remote system printing" on page 89
Provides information about configuring printing on remote servers.

PSF/400 home page
The Print Services Facility for OS/400 (PSF/400) home page provides information about purchasing and configuring IDPS (PSF/400) printing.

iSeries NetServer print shares
This topic provides information about sharing output queues with PC clients in the network.

iSeries Access for Windows® User's Guide
Provides information about configuring a printer that is locally attached to an iSeries Access client. This information is installed as part of the online help system on the iSeries Access client.

For additional information about how to configure printers and related devices, refer to the Printer Device Programming .

Configure line printer requester/line printer daemon (LPR/LPD) printing

1. Start LPD on the print server, PC, or iSeries that you are printing to. If you are printing to another iSeries, you can start LPD with iSeries Navigator:
   a. Expand **Network –> Servers –> TCP/IP**.
   b. Right click on **LPD** from the list of servers, and select **Start**.
2. Setup a remote output queue for the iSeries server. Refer to the IBM knowledge base document "Configuring a Remote Output Queue (RMTOUTQ)", 8983237 for instructions and additional information.
3. Vary on the printer.
   a. Enter WRKCFGSTS *DEV command. The Work with Configuration Status display shows a list of devices.
   b. Enter a 1 next to the printer device description to vary on the printer.
4. Enter STRREMWTR to start the remote writer.

Configure PJL, SNMP, or IPP printers

1. Set up and configure the printer and the network adapter or print server. Refer to the following IBM Knowledge Base documents for recommended hardware settings:
   - Configuration Settings and Error Messages for *LAN 3812 SNMP Device Descriptions, 19935220
   - Configuration Settings and Error Messages for *LAN 3812 PJL Device Descriptions, 14461435
   - For IPP printer settings, refer to the printer documentation.
2. Create a device description in OS/400 using the CRTDEVPRT command. Refer to the following IBM Knowledge Base documents for recommended settings and examples:
   - Configuring a *LAN 3812 SNMP Device Description, 19932815
   - Configuring a *LAN 3812 PJL Device Description, 8695149
   - "Recommended settings for IPP printer device description"
3. Vary the printer on.
   a. Enter WRKCFGSTS *DEV command. The Work with Configuration Status display shows a list of devices.
   b. Enter a 1 next to the printer device description to vary on the printer.
4. Enter STRPRTWTR to start the Print Writer.

Recommended settings for IPP printer device description

For an IPP printer, enter the following information for the device description.

Field	**Suggested value**
Device description	PRT01
Device class	*LAN
Device Type	3812
Device model	1
LAN attachment	*IP
Port number	631

Field	Suggested value
Online at IPL	*YES
Font Identifier	11
Printer error message	*INFO, or *INQ — Selecting *INQ will force a response to error messages.
Host print transform	*YES
Remote location: Name or address:	Enter the IP address, system name, or URL of the printer. For example: http://prt01
Validation list	Optional. If specified, users must be validated before the printer will accept communication.
System driver program	*IBMIPPDRV

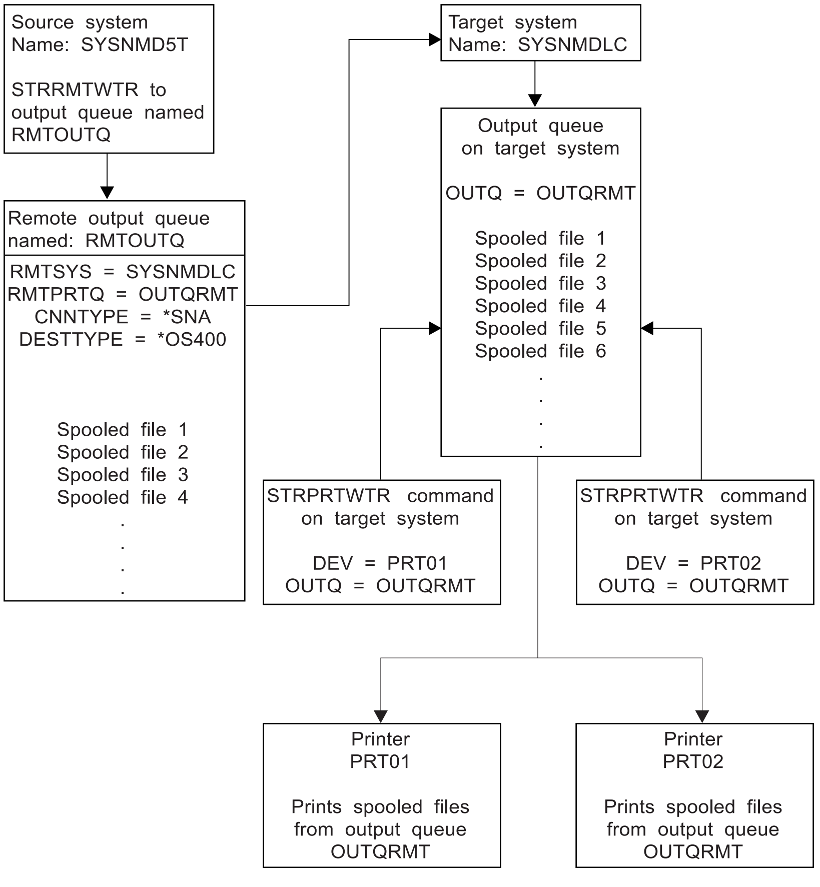
Configure remote system printing

For information about remote system printing, refer to the following:

- "OS/400 to OS/400 for remote system printing"
- "OS/400 to VM/MVS for remote system printing" on page 95
- "OS/400 to NetWare for remote system printing" on page 99

OS/400 to OS/400 for remote system printing

The diagram below illustrates OS/400-to-OS/400 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. Table 3 provides a list of things that must be present or created before remote system printing can be used.

Table 3. Source system and target system preparation for remote system printing

Source system	Target system
Line, controller, and device descriptions created Additional information for these configuration descriptions can be obtained in the Communications Configuration book on the V5R1 Supplemental Manuals Web site.	Line, controller, and device descriptions created Additional information for these configuration descriptions can be obtained in the Communications Configuration book .
Identify the user profiles to be used for remote system printing. 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.	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.

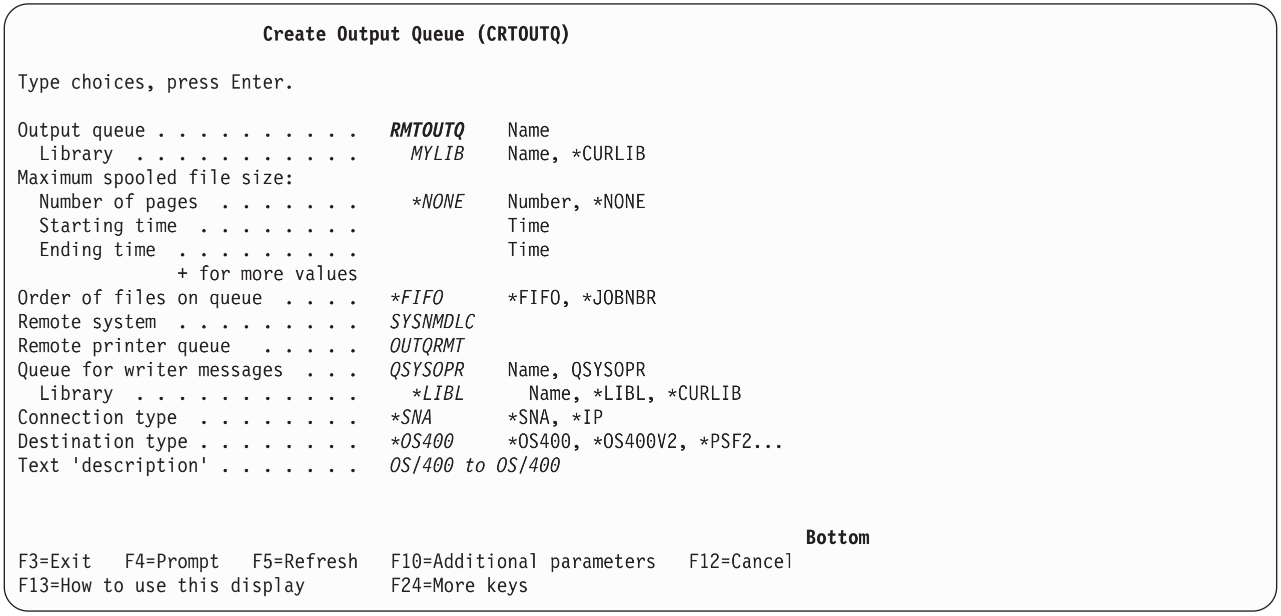
Table 3. Source system and target system preparation for remote system printing (continued)

Source system	Target system
Set up the SNADS network. • Create distribution queues and routing entries using the Configure Distribution Services (CFGDSTSRV) command. – Create a distribution queue. – Create a routing entry and associate the target system name (SYSNMDLC) with the distribution queue. • Add users to the system directory. Use the Add Directory Entry (ADDDIRE) command. – Add a directory entry for the QNETSPLF user profile. (This entry is used to send spooled files to the remote system.) **User ID** This must be QNETSPLF. **Address** This is the value specified on the RMTSYS parameter (SYSNMDLC). **User profile** This must be *NONE. **System name** This is the value specified on the RMTSYS parameter (SYSNMDLC). **System group** Not used in this example. – Add a directory entry for the QNETSPLF user profile. (This entry is the user ID used by SNADS on the source system.) **User ID** This must be QNETSPLF. **Address** This is the name of the source system (SYSNMD5T). **User profile** This must be QNETSPLF. **System name** This is the name of the source system (*LCL). **System group** Not used in this example.	Set up the SNADS network. • Create distribution queues and routing entries using the Configure Distribution Services (CFGDSTSRV) command. – Create a distribution queue. – Create a routing entry; associate the name of the distribution queue and the network ID of the target system (SYSNMD5T). • Add users to the system directory. Use the Add Directory Entry (ADDDIRE) command. – Add a directory entry for QNETSPLF. (This entry is used to receive spooled files from the source system.) **User ID** This value must be QNETSPLF. **Address** This is the value specified on the RMTSYS parameter (SYSNMDLC). **User profile** This must be QNETSPLF. **System name** This must be *LCL. **System group** Not used in this example. – Add another directory entry for user profile QNETSPLF. (This entry is used to send messages back to the source system). **User ID** This must be QNETSPLF. **Address** This is the name of the source system (SYSNMD5T) **User profile** This value must be *NONE. **System name** This is the name of the source system (SYSNMD5T) **System group** Not used in this example.
Create a remote output queue for remote system printing. Use the Create Output Queue (CRTOUTQ) command.	Create an output queue to receive the spooled files using the value on the RMTPRTQ parameter (OUTQRMT).

Source system activity - creating the remote output queue

The result of using this example would be spooled output files sent to output queue OUTQRMT on the target system (SYSNMDLC).

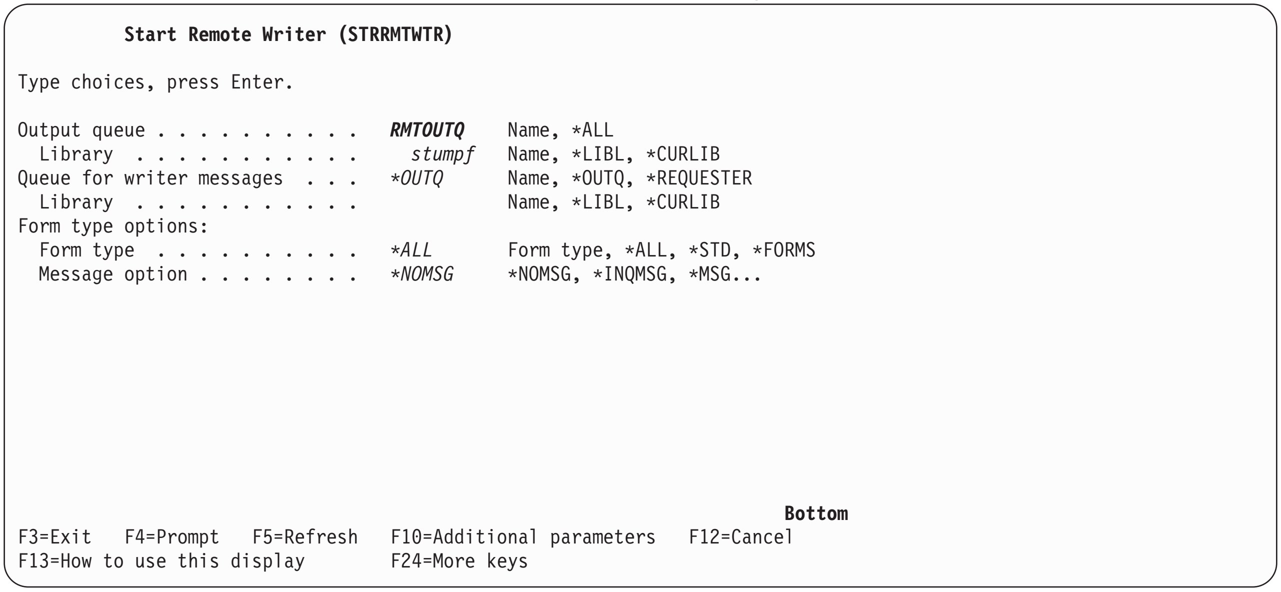
Type CRTOUTQ and press F4 (Prompt). The following display appears. Type the values shown and press the Enter key. The remote output queue RMTOUTQ is created. The other parameter values are needed to ensure connection and delivery to the designated system.



```
                       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' . . . . . . .   OS/400 to OS/400


                                                                          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).

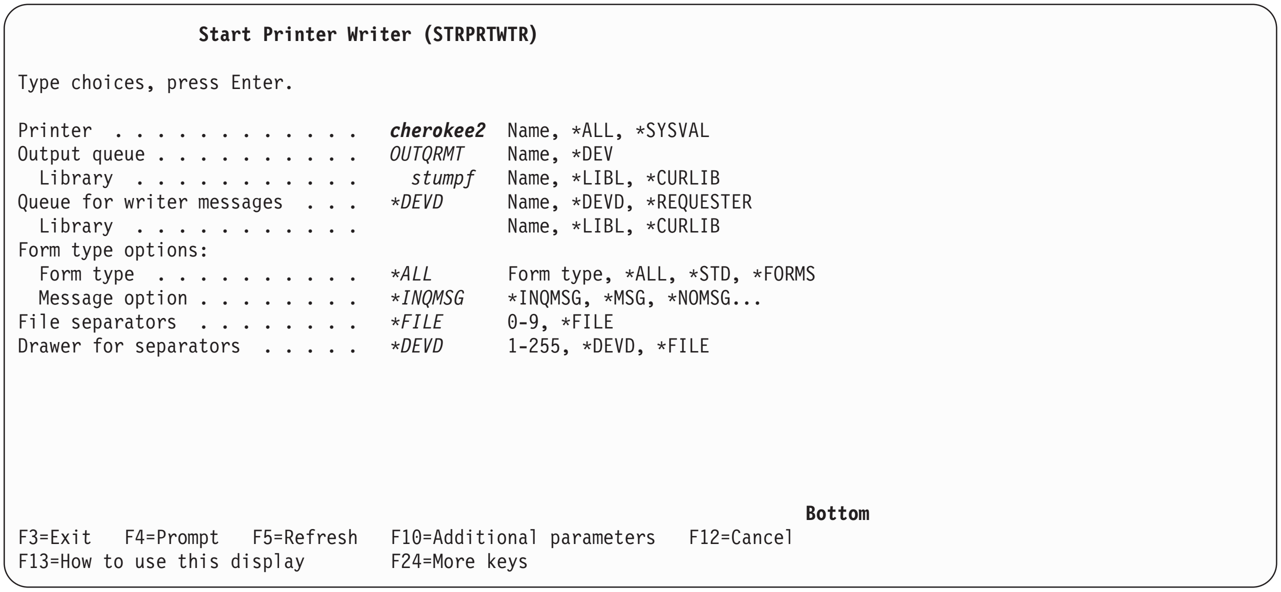


```
          Start Remote Writer (STRRMTWTR)

Type choices, press Enter.

Output queue . . . . . . . . . .   RMTOUTQ    Name, *ALL
  Library  . . . . . . . . . . .     stumpf   Name, *LIBL, *CURLIB
Queue for writer messages  . . .   *OUTQ      Name, *OUTQ, *REQUESTER
  Library  . . . . . . . . . . .              Name, *LIBL, *CURLIB
Form type options:
  Form type  . . . . . . . . . .   *ALL       Form type, *ALL, *STD, *FORMS
  Message option . . . . . . . .   *NOMSG     *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).

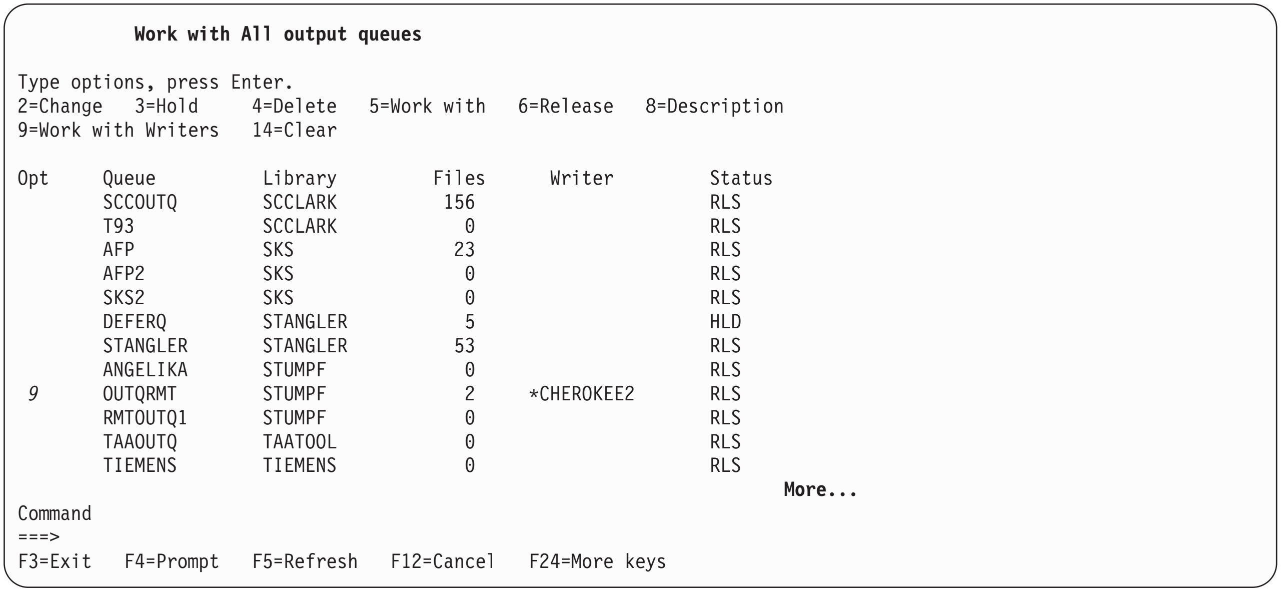
Type the name of the printer (CHEROKEE2) you want to use and the name of the output queue with the received spooled output files (OUTQRMT). Press the Enter key and spooled output files will begin to print on the printer named CHEROKEE2.



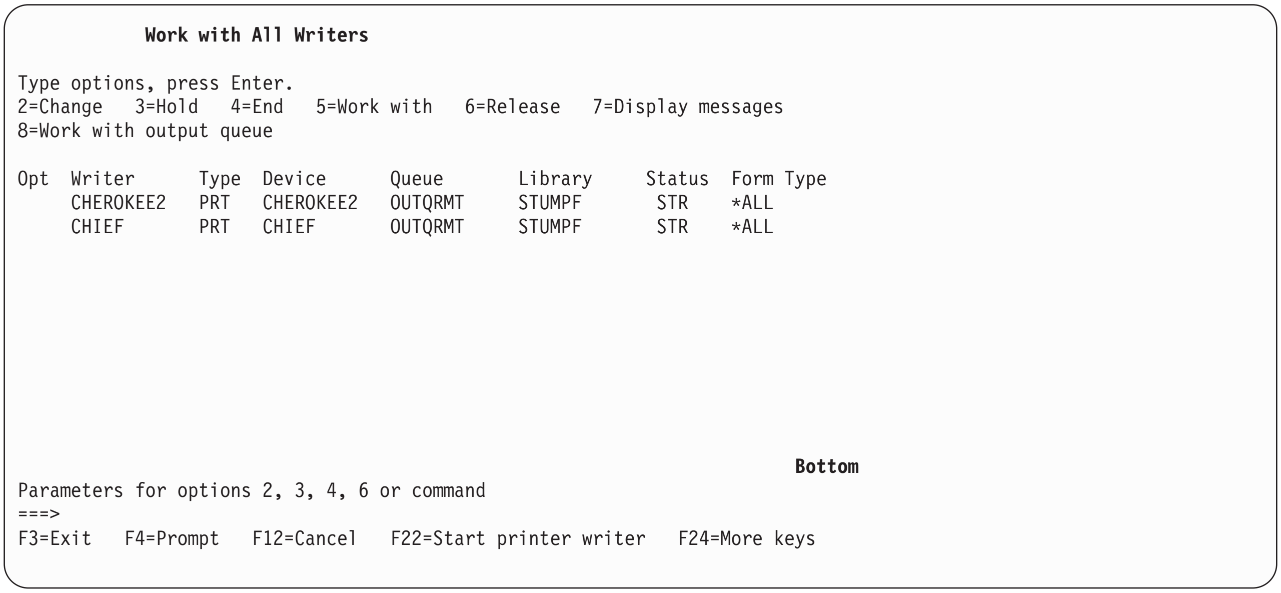
```
                 Start Printer Writer (STRPRTWTR)

Type choices, press Enter.

Printer  . . . . . . . . . . . .   cherokee2  Name, *ALL, *SYSVAL
Output queue . . . . . . . . . .   OUTQRMT    Name, *DEV
  Library  . . . . . . . . . . .     stumpf   Name, *LIBL, *CURLIB
Queue for writer messages  . . .   *DEVD      Name, *DEVD, *REQUESTER
  Library  . . . . . . . . . . .              Name, *LIBL, *CURLIB
Form type options:
  Form type  . . . . . . . . . .   *ALL       Form type, *ALL, *STD, *FORMS
  Message option . . . . . . . .   *INQMSG    *INQMSG, *MSG, *NOMSG...
File separators  . . . . . . . .   *FILE      0-9, *FILE
Drawer for separators  . . . . .   *DEVD      1-255, *DEVD, *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).

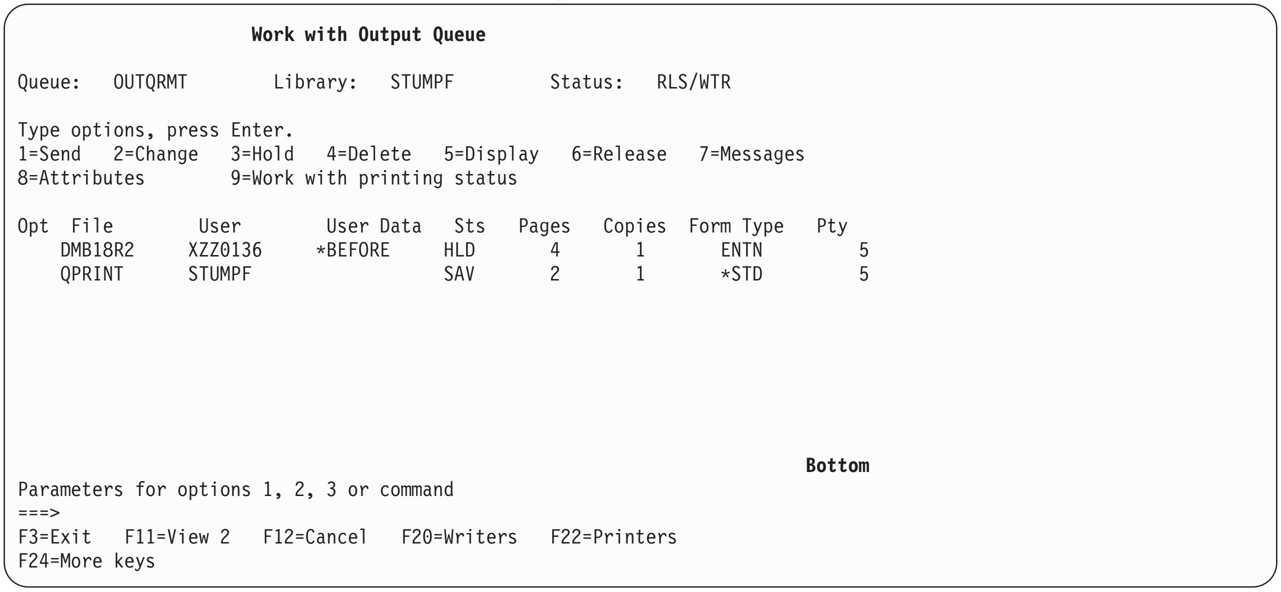


```
           Work with All output queues

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
```

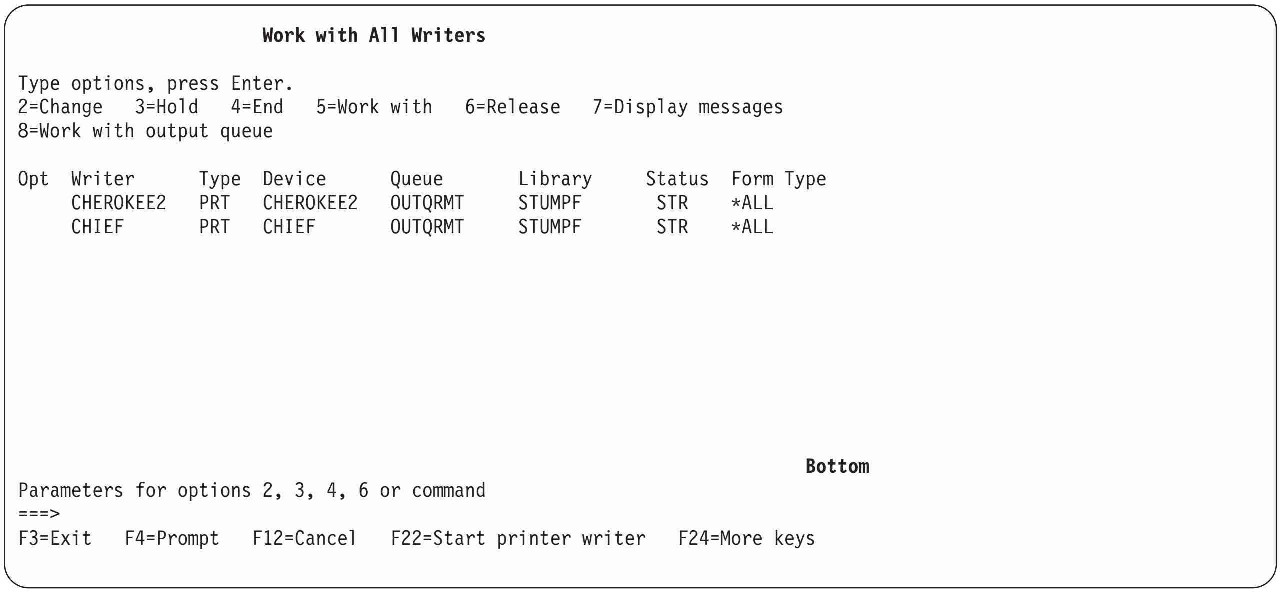


```
             Work with All Writers

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.



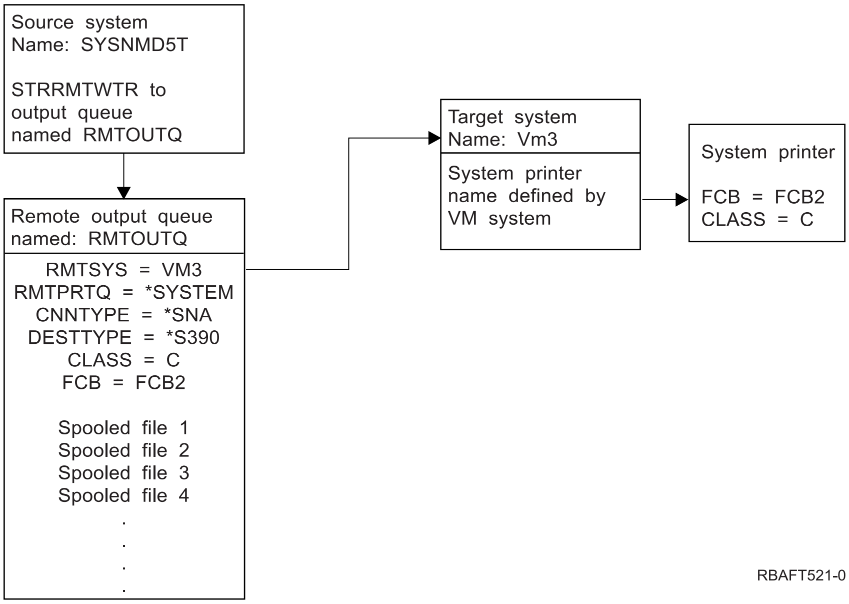
```
                      Work with Output Queue

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
```

OS/400 to VM/MVS for remote system printing

The diagram below illustrates OS/400-to-VM/MVS 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 remote system printing to work. Table 4 on page 96 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 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 the SNA Distribution Services book on the on the V5R1 Supplemental Manuals Web site.	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 OS/400 will require.
Line, controller, and device descriptions created Additional information for these configuration descriptions can be obtained in the Communications Configuration book .	Communication descriptions must exist or be created to connect the VM/MVS system to OS/400. 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 OS/400.
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 OS/400.

Table 4. Source system and target system preparation for remote system printing (continued)

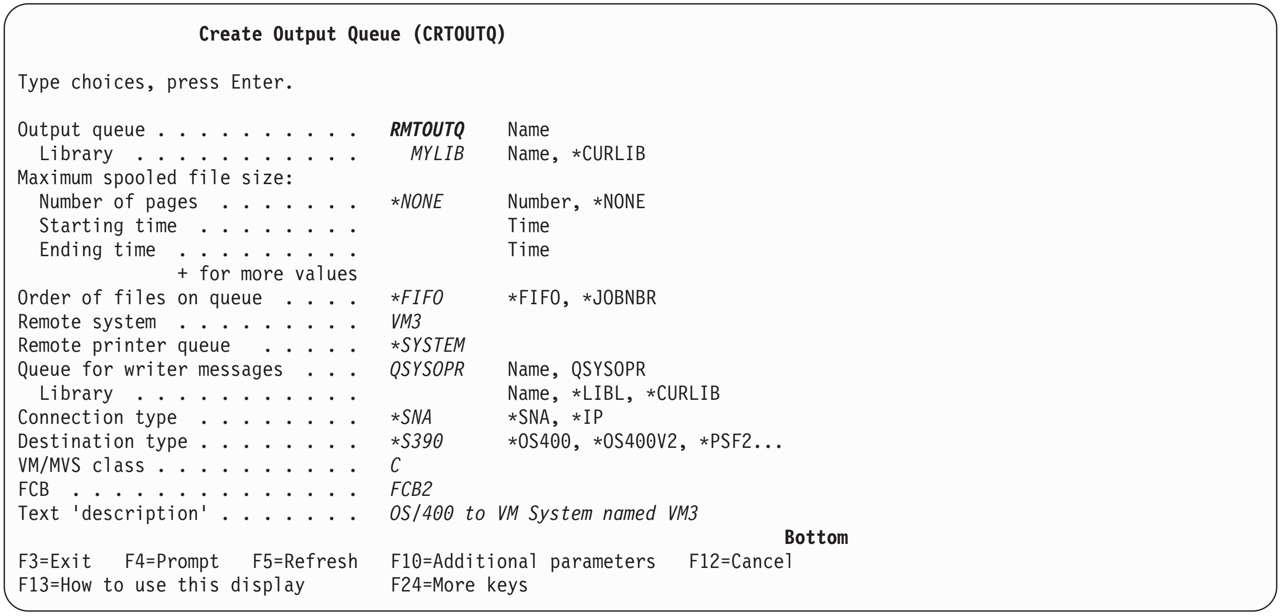
Source system at V3R1 or later	VM/MVS target system
Set up the SNADS network. • Create distribution queues and routing entries using the Configure Distribution Services (CFGDSTSRV) command. – Create a distribution queue. – Create a routing entry and associate the target system name (VM3) with the distribution queue. • Add users to the system directory. Use the Add Directory Entry (ADDDIRE) command. – Add a directory entry for the QNETSPLF user profile. (This entry is the user ID used by SNADS on the source system.) **User ID** This must be QNETSPLF. **Address** This is the name of the source system (SYSNMD5T). **User profile** This must be QNETSPLF. **System name** This is the name of the source system (*LCL). **System group** Not used in this example. – Add a directory entry for QNETSPLF. (This entry is the user ID used by SNADS to route to the target system.) **User ID** This must be QNETSPLF. **Address** This is the name of the target system (VM3). **User profile** This must be *NONE. **System name** This is the name of the target system (VM3). **System group** Not used in this example.	For MVS, identification of the iSeries server must be entered in the JCL or JES and associated with a printer defined to the MVS system. For VM, identification of the iSeries server must be entered in RSCS and associated with a printer defined to the VM system.
Create a remote output queue for remote system printing. Use the Create Output Queue (CRTOUTQ) command.	
When sending spooled output files to VM/MVS, the default forms type must be STANDARD (not to be confused with *STD for OS/400). 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.	

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.

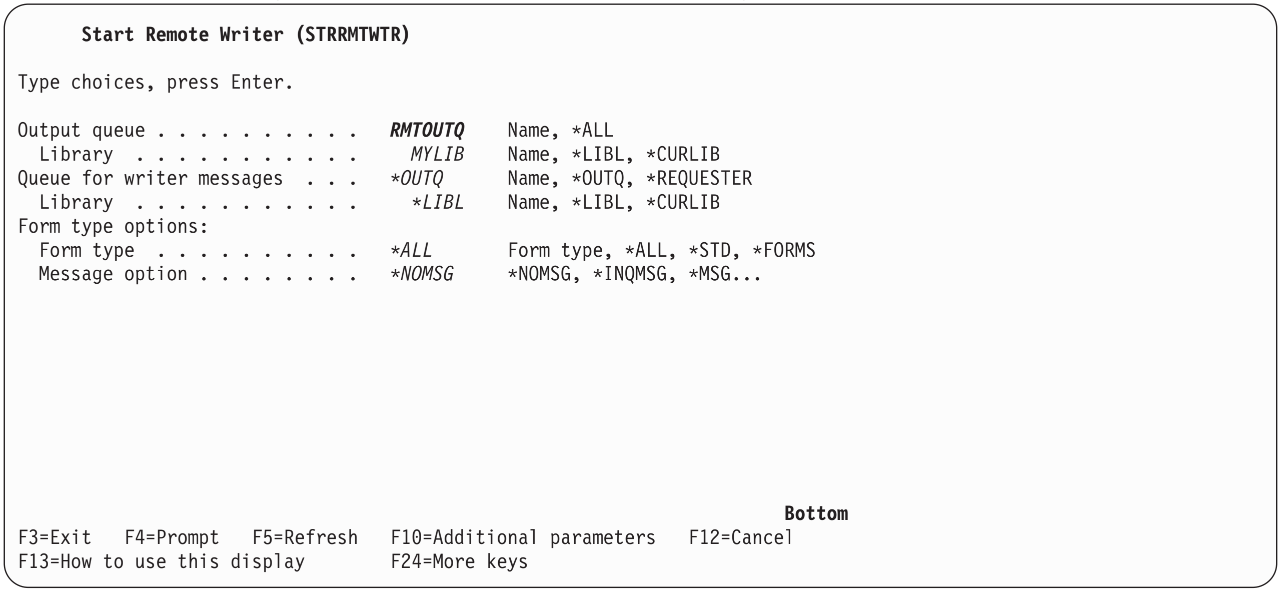
Type CRTOUTQ and press F4 (Prompt). The following display appears. Type the values shown and press the Enter key. The remote output queue TEST1 is created. The other parameter values are needed to ensure connection and delivery to the designated system.



```
                 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' . . . . . . .   OS/400 to VM System named VM3
                                                                        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 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...









                                                                        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 files were sent to the system printer on the target system (VM3).

OS/400 to NetWare for remote system printing

See the Printing from iSeries to NetWare printers topic for instructions on how you can print from iSeries to a NetWare printer that uses the standard NetWare print support and an iSeries remote output queue and remote writer.

Manage 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:

Spooled file related tasks

- “Display a list of spooled files” on page 100
- “Display the contents of a spooled file” on page 100
- “Display messages associated with a spooled file” on page 100
- “Hold a spooled file” on page 100
- “Release a spooled file” on page 101
- “Move a spooled file to another queue” on page 101
- “Delete a spooled file” on page 101
- “Convert a spooled file to PDF” on page 102
- “Copy a spooled file to a physical file” on page 102
- “Send a spooled file” on page 102
- “Change a spooled file’s attributes” on page 102
- “Restart the printing of a spooled file” on page 102
- “Suspend one spooled file and print another” on page 103
- “Enable spooled file notification message” on page 103
- “Control the number of spooled files” on page 103
- “Reclaim spooled file storage” on page 103
- “Save and restore spooled files” on page 105
- “Control printing by spooled file size” on page 105

Miscellaneous tasks

- “Check the status of a printer” on page 105
- “Check the status of a printer writer” on page 105
- “Allow direct printing” on page 105
- “Specify separator pages” on page 106
- “Specify print text” on page 106
- “Replace unprintable characters” on page 107
- “Print a graphic with other output” on page 107
- “Display graphic symbol sets” on page 108
- “Change printer files” on page 108
- “Override printer files” on page 109
- “Delete printer file overrides” on page 111
- “Display printer file overrides” on page 112

Display a list of spooled files

Use the following method to display a list of spooled files (printer output):

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**.

Display the contents of a spooled file

Use either of the following methods to display the contents of a spooled file (printer output):

iSeries Navigatior

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 iSeries Navigator interface has the additional capability of being able to display ASCII spooled files.
2. The character-based interface has the additional capability of being able to display *LINE and *IPDS spooled files.

Display messages associated with a spooled file

Use the following method to display messages associated with a spooled file (printer output):

1. Expand **Basic Operations**.
2. Click **Printer Output**.
3. Right-click the printer output file that has a message.
4. Click **Reply**.

Hold a spooled file

Use either of the following methods to temporarily prevent the spooled file (printer output) that you selected from printing:

iSeries 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.

Release a spooled file

Use either of the following methods to release a spooled file (printer output) that is being held:

iSeries 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.

Move a spooled file to another queue

Use either of the following methods to the spooled file (printer output) from one output queue to another output queue:

iSeries 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.

For more information about moving a spooled file, see “Redirecting output considerations” on page 73.

Delete a spooled file

Use either of the following methods to delete a spooled file (printer output):

iSeries Navigatior

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.

Convert a spooled file to PDF

Do the following to convert a spooled file to a PDF file:

1. Right-click the printer output file that you want to convert.
2. Click **Convert to PDF**.
3. Specify the Convert Printer Output to PDF options and click **OK**.

Copy a spooled file to a physical file

Use the Copy Spooled Files (CPYSPLF) command to copy a spooled file to a physical file.

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.

Send a spooled file

Use the following method to send a spooled file (printer output) to a remote system that is running TCP/IP or to another user on the SNADS network:

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**.

Change a spooled file’s attributes

Use either of the following methods to change a spooled file’s (printer output’s) attributes:

iSeries 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.

Restart the printing of a spooled file

Use the following method to restart printing a spooled file on a particular page:

1. Expand **Basic Operations**.
2. Click **Printer Output**.

3. Right-click the printer output file that you want to restart.
4. Click **Properties**.
5. Click **Pages**.
6. Specify the page information and click **OK**.

Suspend one spooled file and print another

You can temporarily halt the printing of one spooled file (printer output) and start the immediate printing of a second spooled file (printer output). Use the following method:

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.

Enable spooled file notification message

You can be notified when a spooled file (printer output) completes printing or is held by the print writer program. Use the following method to enable notification:

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**.

Control the number of spooled files

The number of spooled files in your server 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 server and the number of spooled files known to the server increase the amount of time needed to perform IPL and internal searches, and increases the amount of temporary storage required.

Periodically identify spooled files that are no longer needed and delete them. For more information on how to display a list of spooled files, see “Display a list of spooled files” on page 100.

You can control the number of job logs that are generated by using the LOG parameter of the Create Job Description (CRTJOBD) command.

Reclaim spooled file storage

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” on page 30.

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 cleanup unused printer output storage on system 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, do the following:

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

Another choice is to use the Reclaim Spool Storage (RCLSPLSTG) command with the DAYS parameter set to *NONE to immediately reclaim all empty spooled file members. This command can be used to cleanup unused printer output storage on system ASPs and basic user ASPs. This command cannot be used to cleanup unused printer output storage on independent ASPs.

Notes:

1. The database member is immediately deleted after the deletion of a spooled file. 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 do the following:

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 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 desired user ASP or independent ASP.

All spooled files you place in this output queue will have the spooled file data stored in the user ASP or independent ASP in a library QSPL*xxxx* 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.

Save and restore spooled files

The Save Spooled File (ZSAVSPLF) and Restore Spooled File sample commands can be used to save or restore a spooled file or a group of spooled files into a library. The library can optionally be saved to a device. The ZRSTSPLF command works only with spooled files saved using the ZSAVSPLF command.

The ZSAVSPLF and ZRSTSPLF commands and documentation are included in the QUSRTOOL library. See member TSRINFO in file QATTINFO in library QUSRTOOL.

Check the status of a printer

Use the following method to check the status of a printer:

1. Expand **Basic Operations**.
2. Click **Printers**. The printers and their status are displayed.
3. Right-click on the printer for a menu of actions.

Check the status of a printer writer

Use the Work with Writer (WRKWTR) command to check the status of a printer writer. For example, enter the following to check the status of PRT01:

```
WRKWRT *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 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.

Control printing by spooled file size

You can use the MAXPAGES parameter on the Create Output Queue (CRTOUTQ) or Change Output Queue (CHGOUTQ) commands 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))
```

Allow 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 ALWDRTPRT parameter of the Start (PRTWTR) 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 cancelled. 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 ALWDRTPRT 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).

For more information about direct printing, see “Direct print considerations” on page 65.

Specify 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 will 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 will 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. See member TBSINFO in file QATTINFO in library QUSRTOOL for information about this sample exit program.

Specify print text

A line of text that is to be printed at the bottom of every page is called the print text. Print text is set by using the PRTTXT parameter on the CRTPRTF, CHGPRTF, or 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.

Replace 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 CRTPRTF, CHGPRTF, or 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. Once 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, may 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 may 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 may 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.

Print 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:

1. There must be only one space between the command word and the parameters.

2. 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 would look like:

   ```
   #$@INCLGRPH filename,,,9.5,
   ```

3. The #$@INCLGRPH control record should be in a print record by itself, because any other data along with it may be considered as parameters.
4. Parameters should immediately follow one another, separated by commas, using no blanks.
5. The characters INCLGRPH must be all uppercase.
6. The #$@INCLGRPH control record must begin in the first column.
7. 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 may 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 below. Parameters `x`, `y`, `w`, and `l` define the area on the page where the graphics file will be 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.

l
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.

Display 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)
```

Change 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 would look like:

```
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 will go to output queue LABELS and two copies of the mailing labels will be printed.

Override printer files

You can temporarily specify a different printer file or printer file attribute by using the Override Printer File (OVRPRTF) CL command. For more information about overriding printer files, see "Printer file overrides" on page 16.

See the following for different methods of overriding printer files:

- "Override file attributes"
- "Override file names or types" on page 110
- "Override file names or types and file attributes of the new file" on page 110
- "Generic override for printer files" on page 111

Override 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)
```

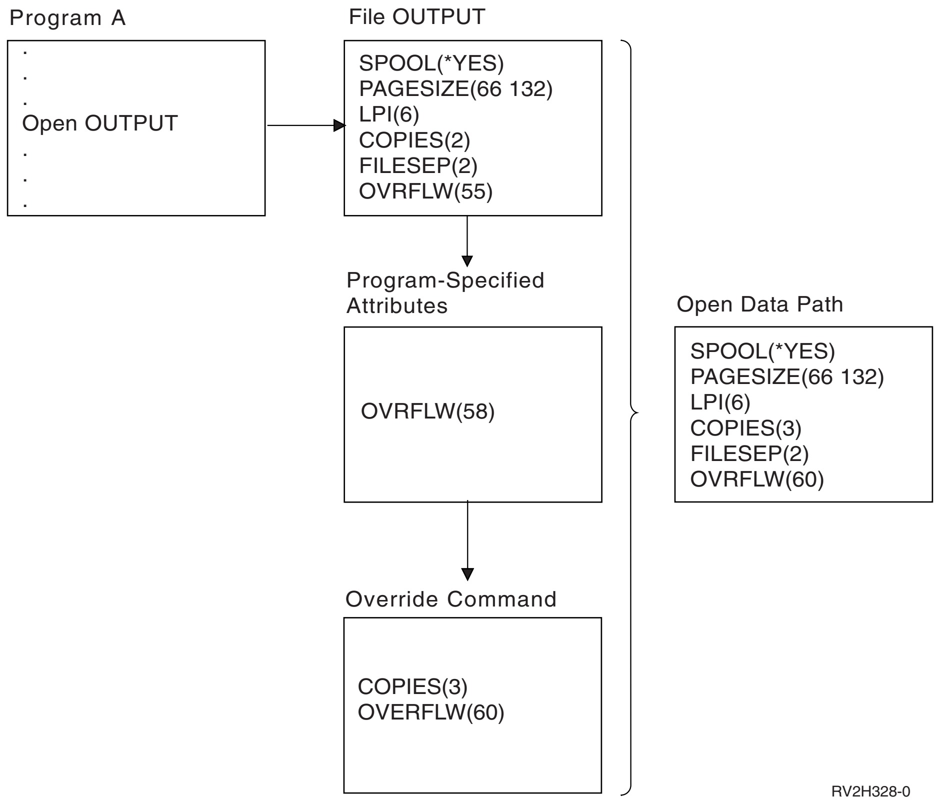
The printer file named OUTPUT is specified in your application program with an overflow line number of 58. However, before you run the application program, you want to change the number of copies of printed output to 3 and the overflow line to 60. The override command looks like this:

```
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 will be 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:



Override file names or types

Another simple form of overriding a file is to change which file is used by the program. This may 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:

```
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.

Override 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 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, type 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 will cause the output to be printed using the REPORTS printer file, but the second override will be ignored.

Generic override for printer files

The OVRPRTF command allows you to have one override for all the printer files in your job with the same set of values. Without the generic override, you would have to 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

Delete 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.

Following is an example that 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 will be 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.

Following is an example that 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 the overrides to PRTC and PRT3 are deleted.

```
OVRPRTF FILE(PRTC)  COPIES(2)
OVRPRTF FILE(PRT3)  COPIES(4)
DLTOVR FILE(*ALL)
```

Display 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.

Printing reference

The following are additional reference materials related to printing:

“Printer file parameter considerations”
Provides more detailed information on certain printer file parameters.

“Printing related CL commands” on page 135
Provides a list of CL commands that are related to printing.

DDS Reference: Printer files
Provides information you need to know for coding the data description specifications (DDS) for printer files.

Printer file parameter considerations

The following printer file parameters are discussed in more detail to help you better understand how to use them:

- “Use the device type (DEVTYPE) parameter” on page 113
- “Use the spooled output schedule (SCHEDULE) parameter” on page 113
- “Use the output priority (OUTPTY) parameter” on page 114
- “Use the align (ALIGN) parameter” on page 115
- “Use the page rotation (PAGRTT) parameter” on page 115
- “Use the MULTIUP(1, 2, 3, or 4) and REDUCE(*TEXT) parameters” on page 118
- “Use the MULTIUP(1, 2, 3, or 4) and REDUCE(*NONE) parameters” on page 122
- “Use the fidelity (FIDELITY) parameter” on page 124
- “Use the overlay (FRONTOVL and BACKOVL) parameters” on page 125

- “Use the margin (FRONTMGN and BACKMGN) parameters” on page 129
- “Use the corner staple (CORNERSTPL) parameter” on page 132
- “Use the edge stitch (EDGESTITCH) parameter” on page 132
- “Use the saddle stitch (SADLSTITCH) parameter” on page 133
- “Use the DBCS coded font (IGCCDEFNT) parameter” on page 134
- “Use the font character set (FNTCHRSET) parameter” on page 135
- “Use the coded font (CDEFNT) parameter” on page 135

Use the 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 (*AFPDSLINE).

An AFP data stream can be created from several sources:

- zSeries
- PrintManager™
- AFP Utilities for OS/400
- Using the virtual print function with the iSeries Access for Windows licensed program

However, to use the functions provided by many of the printer file parameters that are supported for AFPDS, 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 iSeries server does not actually generate the AFP data stream (the list above), either a printer file is not used or the DEVTYPE parameter for the printer file is ignored.

Throughout this topic, AFPDS output that is created on the iSeries server with a printer file specifying DEVTYPE(*AFPDS) is referred to as *AFPDS created on the iSeries server.

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 iSeries server will 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 an iSeries server. It is recommended 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. Instead, put only ASCII commands the target printer can understand. The iSeries server inserts the hex 03 commands with the correct lengths; thus, no EBCDIC to ASCII translation occurs.

Use the 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 may 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 may 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.

Use the 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 may 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.

Use the align (ALIGN) parameter

The align parameter on the Start Printer Writer (STRPRTWTR) and Create Printer File (CRTPRTF) commands influences how the iSeries server 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 iSeries server does not send any messages to check alignment.

Use the 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: 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 would be a candidate for computer output reduction (COR). This applies for printers other than the 3831, 3835, 3900, InfoPrint 3000, and InfoPrint 4000.

If PAGRTT = *AUTO:

- 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 an iSeries server. It is not supported for files with a device type of *USERASCII, *LINE, *AFPDSLINE, or *AFPDS that is not created on an iSeries server.

If PAGRTT = *DEVD:

- 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 may 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.

If PAGRTT = 0, 90, 180, or 270 degrees: 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).

If PAGRTT = *COR:

- 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 iSeries server, 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 a device type of:

– *USERASCII

– *LINE (if the spooled file is sent to a printer configured with AFP(*YES))

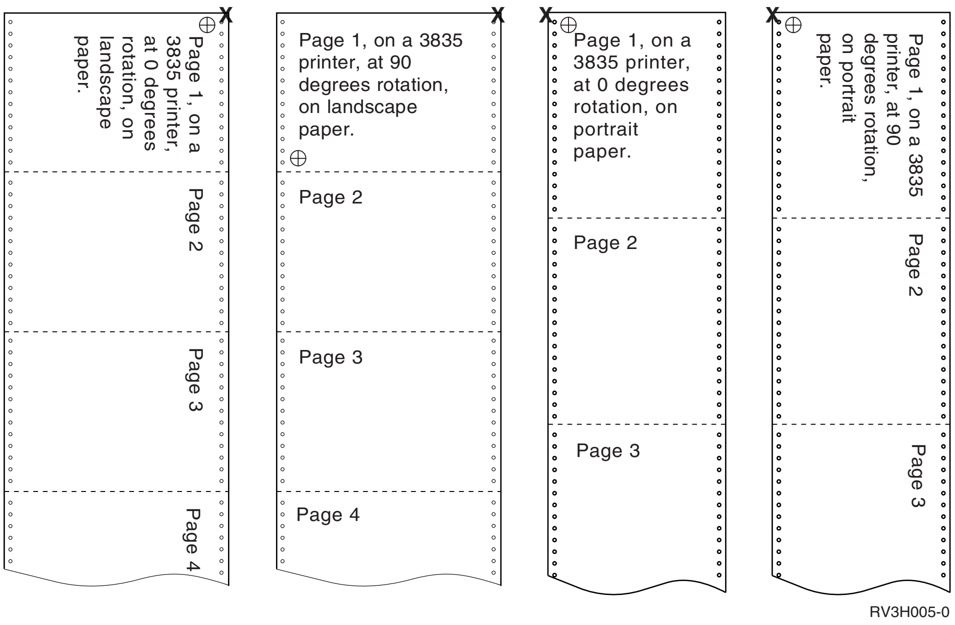
– *AFPDSLINE

– *AFPDS (if the spooled file was not created on the iSeries server)

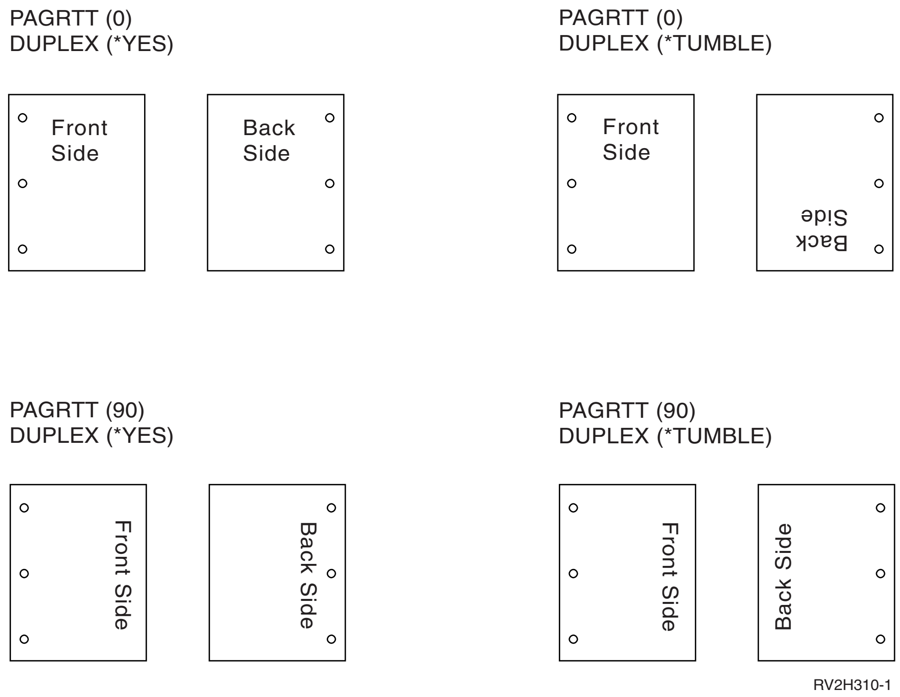
Page rotation on the 3831, 3835, 3900, InfoPrint 3000, and InfoPrint 4000 printers: The 3831, 3835, 3900, InfoPrint 3000, and InfoPrint 4000 printers are continuous-forms printers that can accept page rotation commands. Paper that is wider than it is long has output that is printed at 90-degree rotation. The paper rotation is in a counter-clockwise direction. Normally output rotation is in a clockwise direction. See the diagram below 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.


X
⊕
Page 1, on a 3835 printer, at 0 degrees rotation, on landscape paper.
Page 2
Page 3
Page 4
X
Page 1, on a 3835 printer, at 90 degrees rotation, on landscape paper.
⊕
Page 2
Page 3
Page 4
X
⊕
Page 1, on a 3835 printer, at 0 degrees rotation, on portrait paper.
Page 2
Page 3
X
⊕
Page 1, on a 3835 printer, at 90 degrees rotation, on portrait paper.
Page 2
Page 3
RV3H005-0

Page rotation and duplex printing: The examples below show 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.



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 iSeries can be used to create overlays and page segments.

Use the 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 could 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 on MULTIUP when the REDUCE parameter value is *NONE, see “Use the MULTIUP(1, 2, 3, or 4) and REDUCE(*NONE) parameters” on page 122.

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(*DEVD), 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 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. Below is an example of what the physical output page would look 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
```

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)

```
Page 1

Page 1
Page 1
Page 1
Page 1
Page 1
Page 1

Page 2
Page 2
Page 2
Page 2
Page 2
Page 2
Page 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 5. 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[1]	15
1 through 123 characters	Greater than 5 inches	254	17
124 through 139 characters	All	254	17
140 through 163 characters	All	281[2]	20[2]
Greater than or equal to 164 characters	All	290[3]	27[3]

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 6. 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[1]	15
1 through 73 characters	Greater than 8 inches	254	17
74 through 83 characters	All	254	17
84 through 97 characters	All	281[2]	20[2]
Greater than or equal to 98 characters	All	290[3]	27[3]

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 7. 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[2]	20[2]
Greater than or equal to 72 characters	All	290[3]	27[3]

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 8. 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[1]	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[2]	20[2]
Greater than or equal to 98 characters	All	290[3]	27[3]

Restrictions for MULTIUP with REDUCE(*TEXT): The following are 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 may become too narrow to print.
- MULTIUP with REDUCE(*TEXT) is not supported for spooled files created with device type (DEVTYPE) of *LINE, *AFPDSLINE, or *AFPDS unless the AFPDS 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)

Use the MULTIUP(1, 2, 3, or 4) and REDUCE(*NONE) parameters

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.

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. 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 *AFPDSLINE data streams can be multiupped as can data streams which contain advanced functions and host resident fonts.

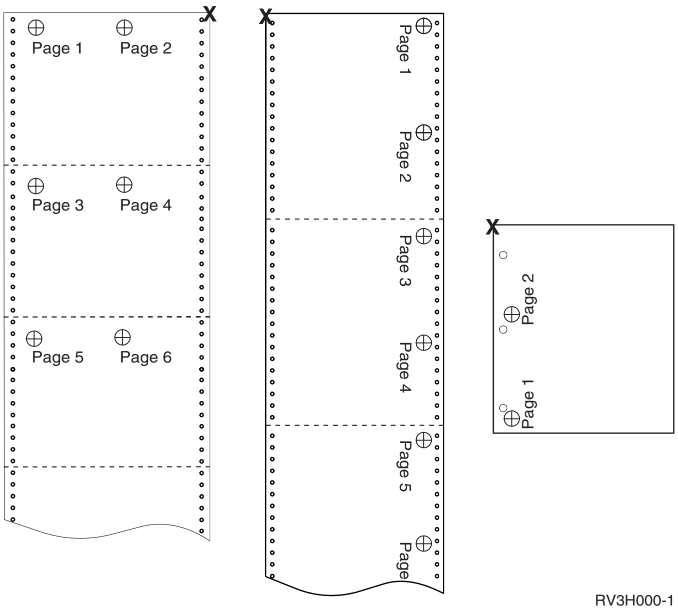
When REDUCE(*NONE) is specified, overlays, page segments, and resources stored in the integrated file system apply to each partition of the multiupped page. Thus, if front overlay (FRONTOVL) and back overlay (BACKOVL) are used for a MULTIUP(2) REDUCE(*NONE) job, the FRONTOVL is printed in the first partition 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.

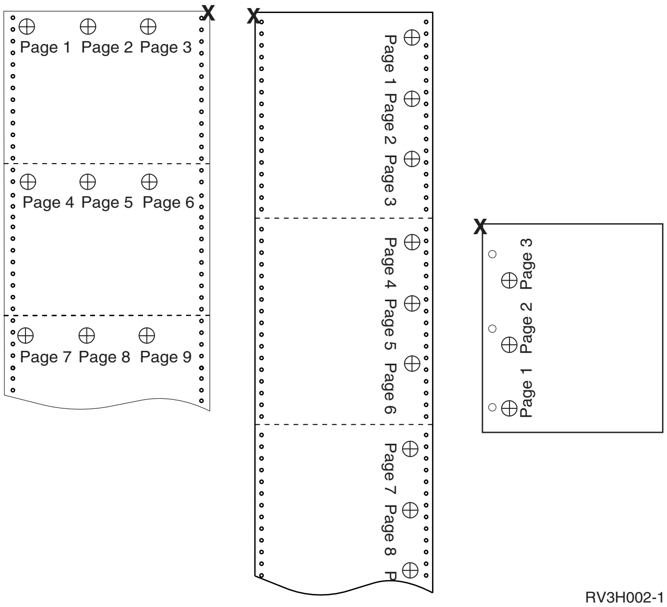
The following diagrams are examples of how output prints when REDUCE(*NONE) and MULTIUP(2, 3, or 4) are specified.

MULTIUP(2), REDUCE(*NONE), and PAGRTT(0)



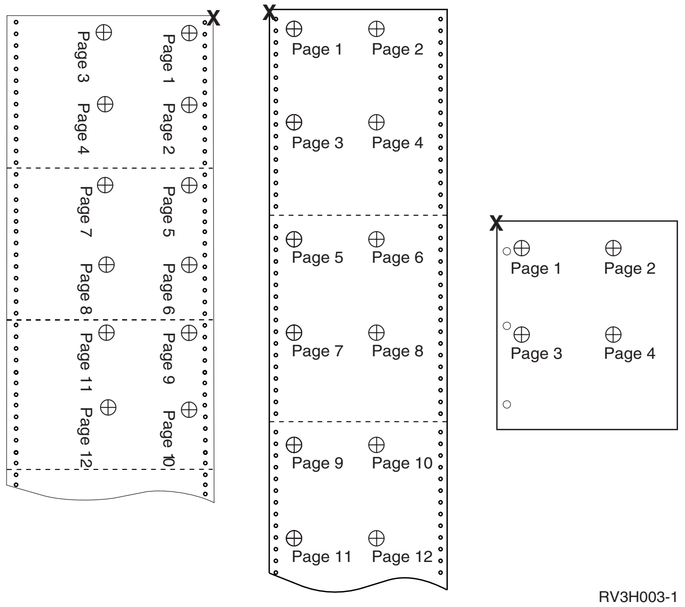
RV3H000-1

MULTIUP(3), REDUCE(*NONE), and PAGRTT(0)



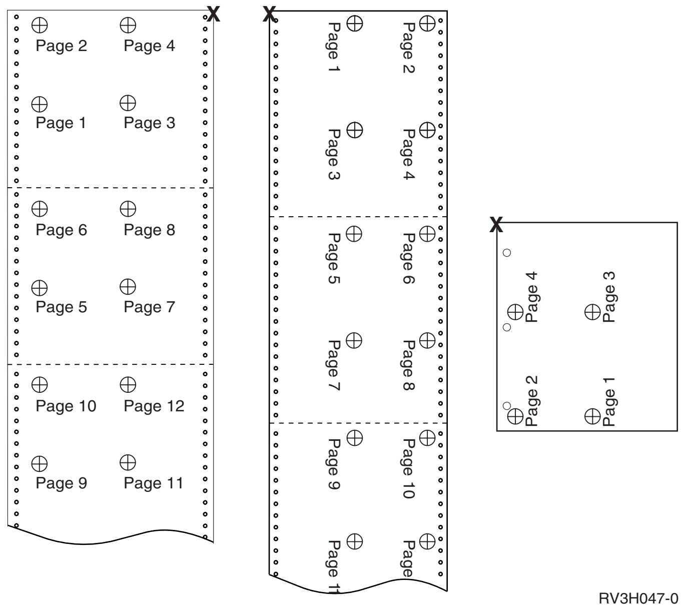
RV3H002-1

MULTIUP(4), REDUCE(*NONE), and PAGRTT(0)



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.



Use the 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.

Use the 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.

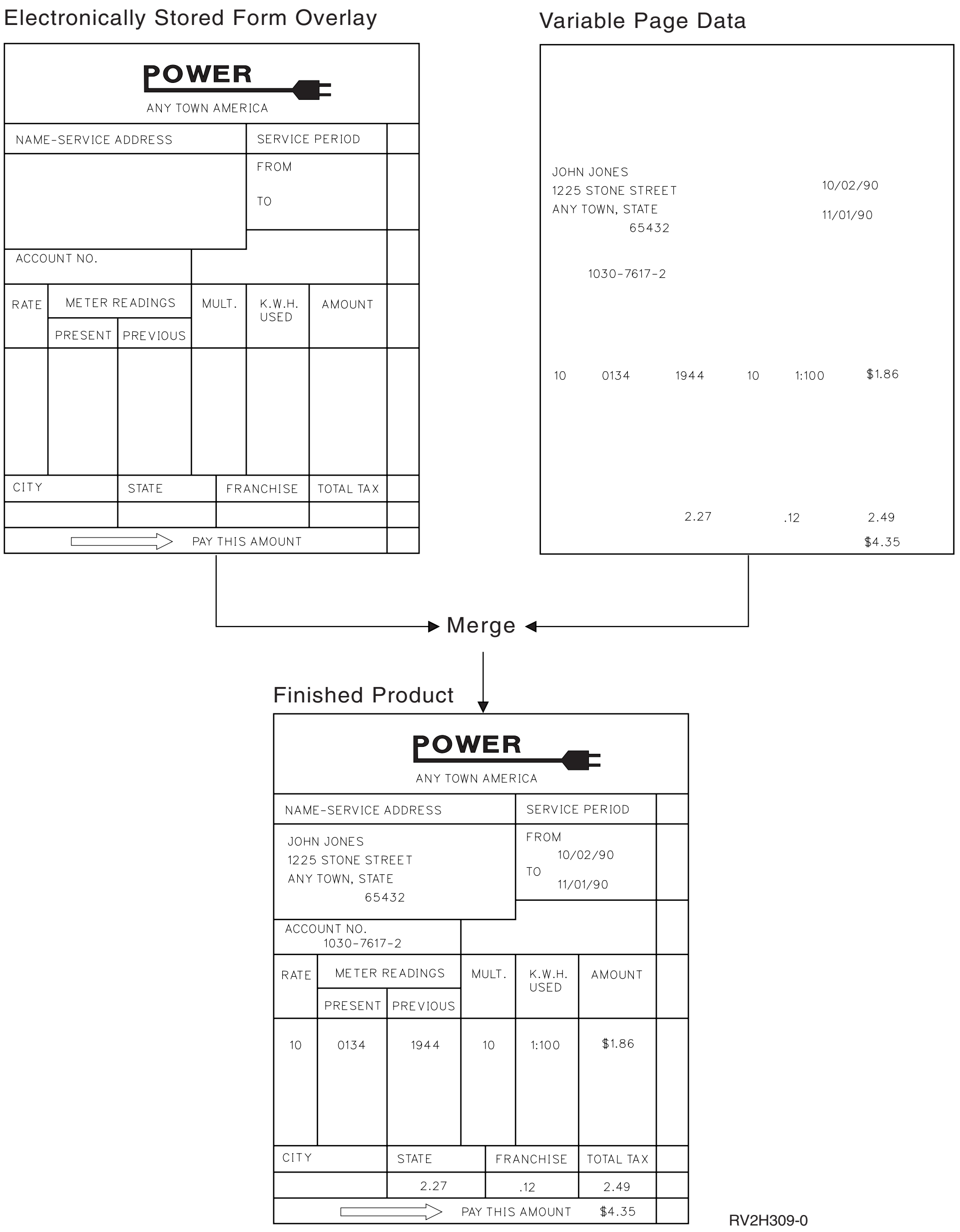
With your application program, you can use overlays that you created yourself using the Advanced Function Printing Utilities for iSeries licensed program, or overlays sent from zSeries.

For more information about the Advanced Function Printing Utilities for iSeries licensed program, see the AFP Utilities for iSeries User's Guide .

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 an iSeries server.



Using overlays and rotation: Rotation can occur for text and overlays. The iSeries server 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.

Notes:

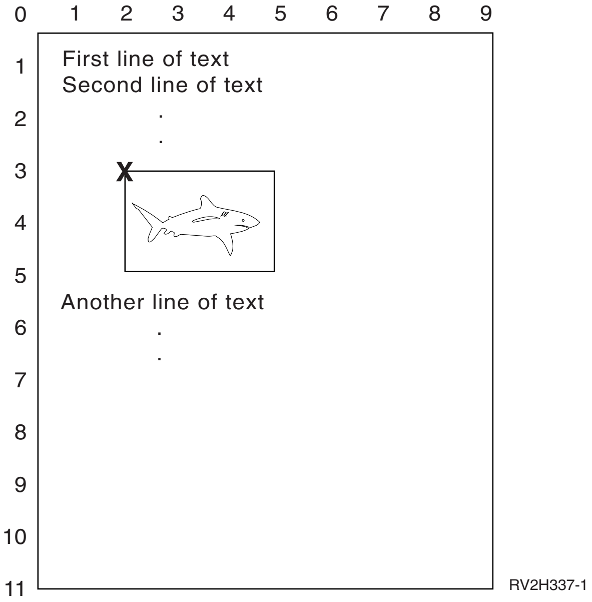
1. Determining the overlay parameter values is made easier by focusing on a particular corner of the overlay depending on the degree of rotation specified.

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 1: 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)



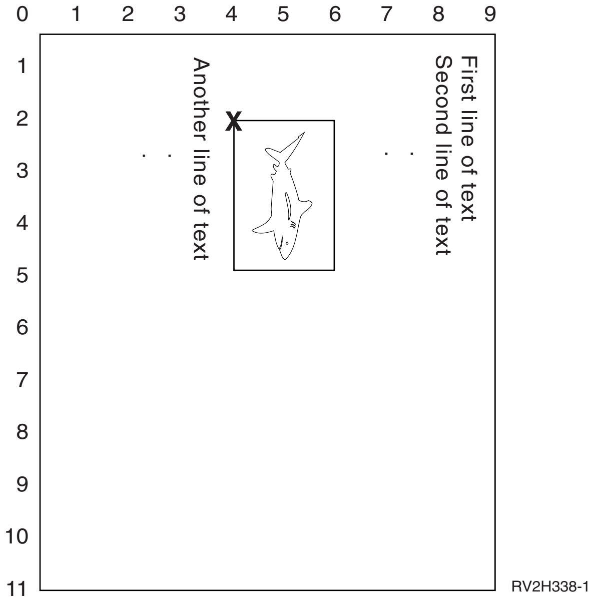
When PAGRTT is 0, the focus point for determining the overlay parameter values is the top left of the overlay (indicated by the **X** in the example).

Use the numbers beside the example page to determine the overlay down (3) and across (2) values.

Overlay parameter values are (3,2).

Example 2: 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)



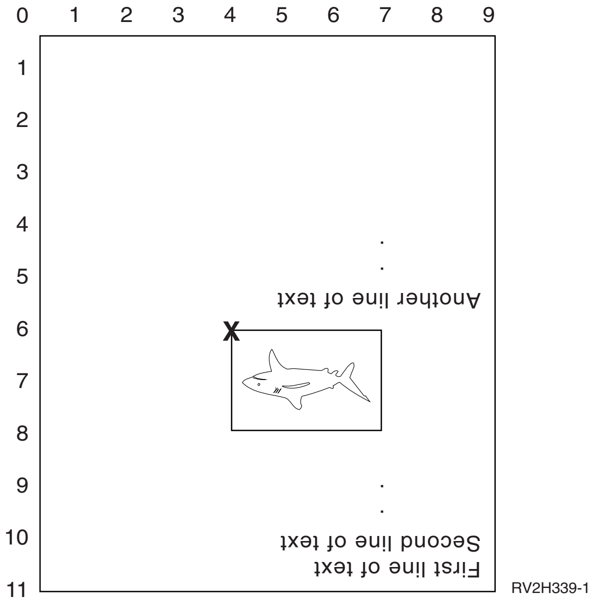
When PAGRTT is 90, the focus point for determining the overlay parameter values is the bottom left of the overlay (indicated by the **X** in the example).

Use the numbers beside the example page to determine the overlay down (2) and across (4) values.

Overlay parameter values are (2,4).

Example 3: 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)



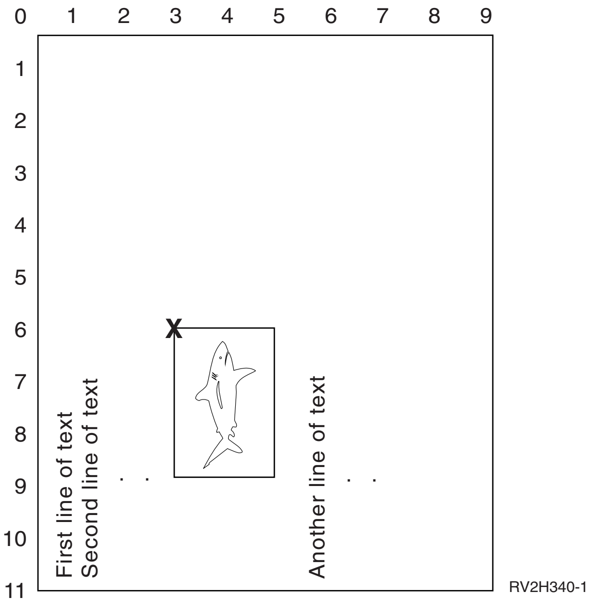
When PAGRTT is 180, the focus point for determining the overlay parameter values is the bottom right of the overlay (indicated by the **X** in the example).

Use the numbers beside the example page to determine the overlay down (6) and across (4) values.

Overlay parameter values are (6,4).

Example 4: 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).

Use the 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 iSeries server 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.

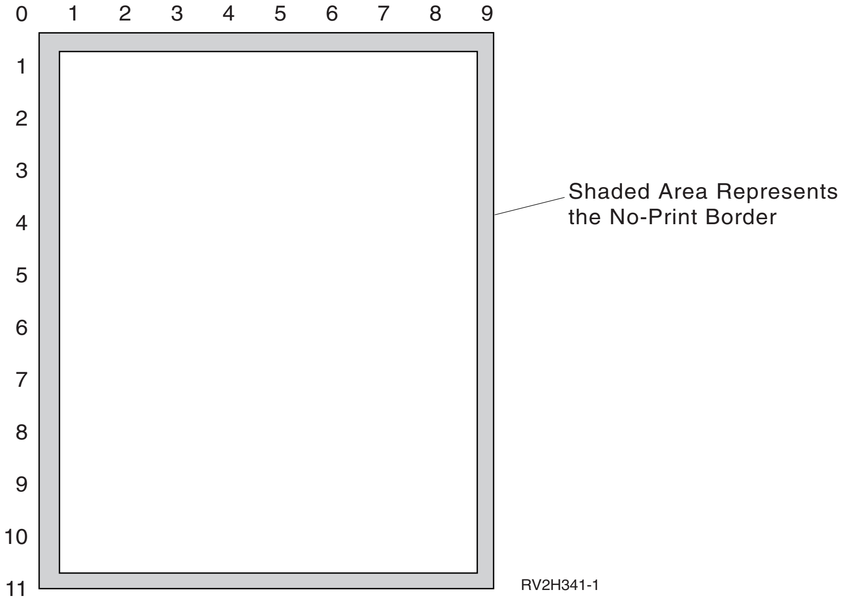
***DEVD**

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.



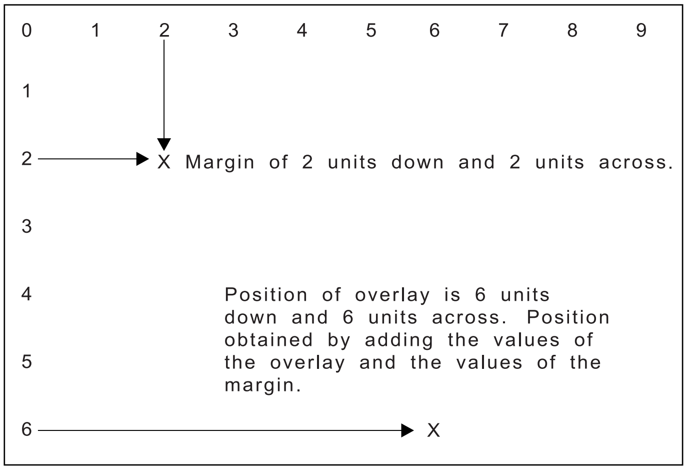
Restrictions with the margin parameters: The margin parameters (both front and back) are ignored for spooled files that have *COR specified as the value for the page rotation (PAGRTT) printer file parameter. This is because *COR assumes a half-inch margin. In addition, the margin parameters are ignored 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 would look 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:

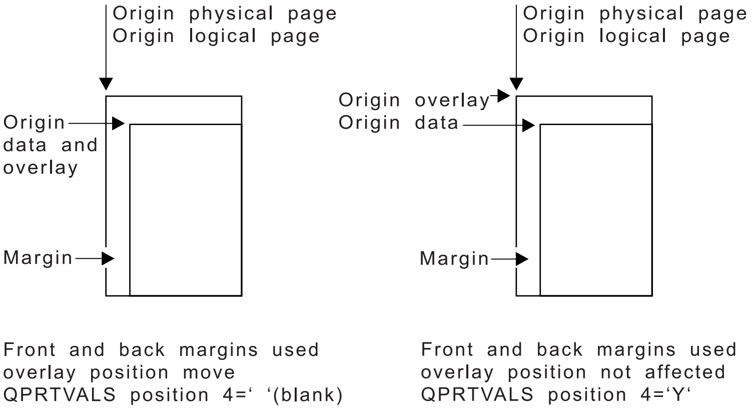

0
1
2
3
4
5
6
7
8
9
1
2
X Margin of 2 units down and 2 units across.
3
4
Position of overlay is 6 units down and 6 units across. Position obtained by adding the values of the overlay and the values of the margin.
5
6
X

RBAFT516-0

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:



RBAFT517-1

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:

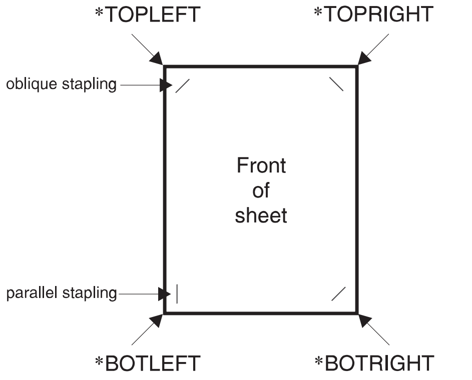
```
CHGDTAARA DTAARA (QUSRSYS/QPRTVALS (4 1)) VALUE ('Y')
```

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) CUROWNAUT(*SAME)`
3. `CRTOBJAUT OBJ(QUSRSYS/QPRTVALS) OBJTYPE(*DTAARA) USER(*PUBLIC) AUT(*ALL)`

Use the 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.

RV4W160-0

Use the 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 *DEVD 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:

***DEVD**
The default that is used by the device.

***BOTTOM**
The reference edge is the bottom edge.

***LEFT** The reference edge is the left-hand edge.

***RIGHT**
The reference edge is the right-hand 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:

***DEVD**
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**

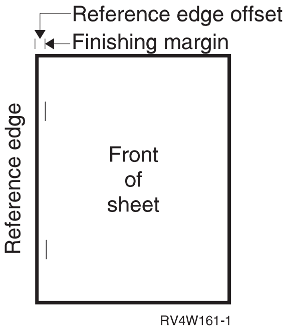
The default staple offset for the device. If you specify a value for the Number of staples, the printer will automatically calculate the position of each staple.

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 diagram below 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



Use the 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 *DEVD for the element.

Element 1: Reference Edge

Specifies which edge to use for the saddle stitch. Possible values are:

***DEVD**
The default that is used by the device.

***TOP** The reference edge is the top edge.

***LEFT** The reference edge is the left-hand edge.

Element 2: Number of Staples

Specifies the number of staples to use for saddle 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. When specifying *DEVD 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:

***DEVD**
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 *DEVD, the Number of staples must be *DEVD. This measurement is in centimeters (range 0 through 57.79) or inches (range 0 through 22.57)

Use the 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 may be specified for outline fonts. It will be ignored for raster fonts.

Transforming SCS DBCS data to AFPDS data: When transforming the SCS data stream to AFPDS, 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 AFPDS 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 AFPDS data to spool: The IGCCDEFNT parameter is used when generating AFPDS. 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.

Use the 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 may 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.

You can find out which font character sets and code pages come with the iSeries server by using the Work with Font Resources (WRKFNTRSC) command and specifying QFNTCPL for the library and *FNTCHRSET or *CDEPAG as the object attribute.

Font character sets and code pages are downloaded from the iSeries server 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 and the font and code page specified in the printer file parameter FNTCHRSET are used.

Use the 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 may be specified for outline fonts. It will be ignored for raster fonts.

You can find out which coded fonts come with the iSeries server by using the Work with Font Resources (WRKFNTRSC) command and specifying 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.

Printing related CL commands

The following groups of CL commands may be used to configure and manage printing.

- “Commands used with printer devices” on page 136
- “Commands used with printer files” on page 136
- “Commands used with printer writers” on page 136
- “Commands used with output queues” on page 137

- “Commands used with spooled files” on page 137
- “Commands used with jobs” on page 138
- “Commands used with user profiles” on page 138

Commands used with printer devices

The following commands may 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.
WRKGFGSTS	Display a list of devices and work with their status.

Commands used with printer files

The following commands may 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

Commands used with printer writers

The following commands may 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.
STRRMTWTR	Start a remote writer to a specified output queue. The spooled files in that queue are sent to a specified remote system.
WRKWTR	Display all the printers configured to the system by specifying WTR(*ALL) and the output queue (OUTQ) parameter.

Commands used with output queues

The following commands may 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.
RLSOUTQ	Release a previously held output queue for processing by the printer writer.
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.

Commands used with spooled files

The following commands may be used to work with spooled files (also known as printer output).

CL command	Description	iSeries Navigator instructions
CHGSPLFA	Change some attributes of a spooled file such as the output queue name or number of copies.	"Change a spooled file's attributes" on page 102
CPYSPLF	Copy a spooled file to a specified database file.	Not applicable.
DLTSPLF	Delete a spooled file from the output queue.	"Delete a spooled file" on page 101
DSPSPLF	Display data records of a spooled file.	"Display the contents of a spooled file" on page 100
HLDSPLF	Stop the processing of an output file by a printer writer.	"Hold a spooled file" on page 100
RCLSPLSTG	Reclaim empty spooled file members.	Not applicable.
RLSSPLF	Release a previously held spooled output file for processing by the printer writer.	"Release a spooled file" on page 101
SNDNETSPLF	Send a spooled file to another system using SNADS.	"Send a spooled file" on page 102
SNDTCPSPLF	Send a spooled file to another system using TCP/IP.	"Send a spooled file" on page 102
WRKSPLF	Display a list of spooled files and then be able to perform a variety of actions on those files.	"Display a list of spooled files" on page 100
WRKSPLFA	Show the current attributes of a spooled file.	Not applicable.

Commands used with jobs

The following commands may be used to work with jobs.

CL command	Description
CHGJOB	Change some of the attributes of a job such as the default printer device or output queue.
CHGJOBD	Change the values of a job description. If you have the correct level of authority, you can change other user's job description values.
CRTJOBD	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.
WRKJOBD	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.

Commands used with user profiles

The following commands may be used to work with user profiles.

CL command	Description
CHGUSRPRF	Change the values of a job description. If you have the correct level of authority, you can change other user's job description values.
CRTUSRPRF	Create job descriptions. When a user signs on, the values in the job description determine which output queue and printer device are used.
DSPUSRPRF	Create job descriptions. When a user signs on, the values in the job description determine which output queue and printer device are used.
DLTUSRPRF	Create job descriptions. When a user signs on, the values in the job description determine which output queue and printer device are used.
WRKUSRPRF	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.

Troubleshoot 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 on how to display a list of spooled files, including the status of the spooled files, see “Display a list of spooled files” on page 100.
- The printer may be varied off or may need you to reply to a message. For information on how to display a list of printers, including the status of the printers, see “Check the status of a printer” on page 105.
- The printer writer is not started, has been held, or has ended. For information on how to display the status of the printer writer, see “Check the status of a printer writer” on page 105.

For additional help troubleshooting printing problems, refer to IBM Printing Systems Support (www.printers.ibm.com/R5PSC.NSF/Web/support+overview) and the IBM Knowledge Base (www.ibm.com/eserver/iseries/support/s_dir/slkbase.nsf/slkbase).

Related information for printing

Listed below are the iSeries manuals and IBM Redbooks (in PDF format), Web sites, and Information Center topics that relate to the printing topic. You can view or print any of the PDFs.

Manuals

- Printer Device Programming
- Data Stream and Object Architectures: Graphics Object Content Architecture, SC31-6804
- For a list of Advanced Function Presentation (AFP) manuals, see Other printing.

IBM Redbooks (www.redbooks.ibm.com)

- IBM AS/400® Printing V, SG24-2160
- IBM AS/400 Printing VI, SG24-6250

Web sites

- Printing Solutions for iSeries (www.printers.ibm.com/R5PSC.NSF/Web/as400overview)
- PSF/400 home page (www.printers.ibm.com/R5PSC.NSF/Web/psfas400Home)
- IBM Printing Systems Support (www.printers.ibm.com/R5PSC.NSF/Web/support+overview)
- IBM Knowledge Base (www.ibm.com/eserver/iseries/support/s_dir/slkbase.NSF)

Other information

- DDS Reference: Printer Files
- iSeries Access for Web
- iSeries Access for Windows
- iSeries NetServer
- Work Management
- Working with printer output

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

1. Right-click the PDF in your browser (right-click the link above).
2. Click **Save Target As...** if you are using Internet Explorer. Click **Save Link As...** if you are using Netscape Communicator.
3. Navigate to the directory in which you would like to save the PDF.
4. Click **Save**.

You need Adobe Acrobat Reader to view or print these PDFs. You can download a copy from the Adobe Web site (www.adobe.com/products/acrobat/readstep.html) .

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 give 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:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

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 49XA
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 information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

The licensed program described in this information 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.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

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.

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. 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 INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO SOME OR ALL OF THE ABOVE LIMITATIONS OR EXCLUSIONS MAY NOT APPLY TO YOU.

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_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

400
Advanced Function Presentation
Advanced Function Printing
AFP
AS/400
GDDM
IBM
ibm.com
Infoprint
Intelligent Printer Data Stream
IPDS
iSeries
MVS
Operating System/400
OS/400
Print Services Facility
PrintManager
Redbooks
zSeries

Microsoft®, Windows, Windows NT®, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

Terms and conditions for downloading and printing information

Permissions for the use of the information you have selected for download are granted subject to the following terms and conditions and your indication of acceptance thereof.

Personal Use: You may reproduce this information for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative works of this information, or any portion thereof, without the express consent of IBM.

Commercial Use: You may reproduce, distribute and display this information solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of this information, or reproduce, distribute or display this information 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 information or any 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 information 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 THIS INFORMATION. THE INFORMATION IS 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.

By downloading or printing information from this site, you have indicated your agreement with these terms and conditions.

Printed in USA