IBM

System i

Files and file systems
Spooled files

Version 6 Release 1

System i

Files and file systems
Spooled files

Version 6 Release 1

Note

Before using this information and the product it supports, read the information in "Notices," on page 33.

This edition applies to version 6, release 1, modification 0 of IBM i5/OS (product number 5761–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

Spooled files

Spooling is a system function that saves data in a spooled file for later processing or printing. Spooled files work in a way similar to tape files or other device files. Spooled files can help you manage your data targeted for externally attached devices, such as a printer.

A spooled file holds output data until it can be printed. The spooled file collects data from a device until a program or device is able to process the data. A program uses a spooled file as if it were reading from or writing to an actual device.

Spooling functions can help system users to manage input and output operations more efficiently.

At the end of a job, the job log can be written to spooled file QPJOBLOG to be printed. After the job log is written to the spooled file, the job log is deleted.

Related concepts

Spooled files and output queues

PDF file for Spooled files

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

To view or download the PDF version of this document, select Spooled files (about 433 KB).

You can view or download these related topics:

- Control language provides a description of the control language (CL) and its commands. Each command is defined including its syntax diagram, parameters, default values, and keywords.
- Basic printing provides information about how to understand and control printing: printing elements and concepts, printer file support, print spooling support, printer connectivity, advanced function printing, and printing with personal computers

Saving PDF files

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

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

Downloading Adobe Reader

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

Spooling concepts

The system supports output spooling and input spooling. Output spooling can be used for printer devices. Input spooling applies to database file input.

Output spooling

Output spooling can be used for both printer and diskette devices. Output spooling sends job output to disk storage instead of sending it directly to a printer or diskette output device. Output spooling allows the job that produces the output to continue processing without consideration for the speed or availability of output devices.

Additionally, output spooling allows the server to produce output on multiple output devices, such as printer and diskette devices, in an efficient manner. It does this by sending the output of a job destined for a printer to disk storage. This process breaks a potential job limitation imposed by the availability or speed of the output devices.

The main elements of output spooling are:

- **Device description:** A description of the printer device.
- **Spooled file:** A file containing spooled output records that are to be processed on an output device.
- **Output queue:** An ordered list of spooled files.
- **Writer:** A program that sends files from an output queue to a device.
- **Application program:** A high-level language program that creates a spooled file using a device file with the spooling attribute specified as SPOOL(*YES).
- **Device file:** A description of the format of the output and a list of attributes that describe how the server should process the spooled file.

Output spooling functions are performed by the server without requiring any special operations by the program that produces the output. When a device file is opened by a program, the operating system determines whether the output is to be spooled. When a printer file that specifies spooling is opened, the spooled file that contains the output of the program is placed on the appropriate output queue in the server.

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. A printer writer is started in the spooling subsystem to send the records to the printer. The spooled file is selected from an output queue.

Spooling device descriptions

Device descriptions must be created for each printer and diskette device in order to define that device to the server. Printer device descriptions are created using the Create Device Description for Printer (CRTDEVPRT) command; diskette device descriptions are created using the Create Device Description for Diskette (CRTDEVDKT) command.

File redirection of spooled files

File redirection occurs when a spooled file is sent to an output device other than the one for which it was originally intended. File redirection can involve devices that process different media (such as printer output sent to a diskette device) or devices that process the same type of media but are of different device types (such as 5219 Printer output sent to a 4224 Printer).

Depending on the new output device for the spooled file, the file can be processed just as it would have been on the originally specified device. However, differences in devices often cause the output to be formatted differently. In these cases, the server sends an inquiry message to the writer’s message queue to inform you of the situation and allow you to specify whether you want printing to continue.

Output queues and spooled files

Batch and interactive job processing can result in spooled output records that are to be processed on an output device, such as a printer or diskette drive. These output records are stored in spooled files until they can be processed. A single job can have many spooled files.

When a spooled file is created, the file is placed on an output queue. Each output queue contains an ordered list of spooled files. A job can have spooled files on one or more output queues. All spooled files on a particular output queue should have a common set of output attributes, such as device, form type, and lines per inch. Using common attributes on an output queue reduces the amount of intervention required and increases the device throughput.

The following lists some of the parameters on the Create Output Queue (CRTOUTQ) command and what they specify:

- MAXPAGES: Specifies the maximum spooled file size in pages that is allowed to be printed between a starting and ending time of day.
- AUTOSTRWTR: Specifies the number of writers that are started automatically to this output queue.
- DSPDTA: Whether users without any special authority but who do have *USE authority to the output queue can display, copy, or send the contents of spooled files other than their own. By specifying *OWNER for DSPDTA, only the owner of the file or a user with *SPLCTL special authority can display, copy, or send a file.
- JOBSEP: The number of job separator pages, if any, that are to be printed between the output of each job when the output is printed.
- DTAQ: The data queue associated with this output queue. If specified, an entry is sent to the data queue whenever a spooled file goes to ready status on the queue.
- OPRCTL: Whether a user who has job control authority can control the output queue (for example, if the user can hold the output queue).
- SEQ: Controls the order in which spooled files are sorted on the output queue.
- AUTCHK: Specifies what type of authority to the output queue that enables a user to control the spooled files on the output queue (for example, enables a user to hold the spooled files on the output queue).
- AUT: Public authority. Specifies what control users have over the output queue itself.
- TEXT: Text description. Up to 50 characters of text that describes the output queue.

Default system output queues

Defaults on CL commands 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 QPRTDEV server value.

When a spooled file is created by opening a device 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 provided:

- **QDKT**: Default diskette output queue
- **QPRINT**: Default printer output queue
- **QPRINTS**: Printer output queue for special forms
- **QPRINT2**: Printer output queue for 2-part paper

Spooling writers

A writer is an i5/OS® program that takes spooled files from an output queue and produces them on an output device. The spooled files that have been placed on a particular output queue remain stored in the system until a writer is started to the output queue.

The writer takes spooled files one at a time from the output queue, based on their priority. The writer processes 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.

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

The AUTOEND parameter on the start writer commands determines whether the writer continues to wait for new spooled files to become available to be written, end after processing one file, or end after all spooled files that have a ready status have been taken from the output queue.

Spooling writer commands

Here are the commands that you can use to control spooling writers.

- Start Diskette Writer (STRDKTWTR): Starts a spooling writer to a specified diskette device to process spooled files on that device.
- Start Printer Writer (STRPRTWTR): Starts a spooling writer to a specified printer device to process spooled files on that device.
- Start Remote Writer (STRRMTWTR): Starts a spooling writer that sends spooled files from an output queue to a remote system.
- Change Writer (CHGWTR): Changes some writer attributes, such as form type, number of file separator pages, or output queue attributes.
- Hold Writer (HLDWTR): Stops a writer at the end of a record, at the end of a spooled file, or at the end of a page.
- Release Writer (RLSWTR): Releases a previously held writer for additional processing.
- End Writer (ENDWTR): Ends a spooling writer and makes the associated output device available to the server.

Note: You can define some functions to provide additional spooling support. Example source and documentation for the commands, files, and programs for these functions are part of library QUSRTOOL, which is an optionally installed part of i5/OS.

Input spooling

Input spooling takes the information from the input device, prepares the job for scheduling, and places an entry in a job queue. Using input spooling, you can typically shorten job run time, increase the number of jobs that can be run sequentially, and improve device throughput.

The main elements of input spooling follow:

- **Job queue**: An ordered list of batch jobs submitted to the system for running and from which batch jobs are selected to run.
- **Reader**: A function that takes jobs from an input device or database file and places them on a job queue.

When a batch job is read from an input source by a reader, the commands in the input stream are stored in the system as requests for the job, the inline data is spooled as inline data files, and an entry for the job is placed on a job queue. The job information remains stored in the system where it was placed by the reader until the job entry is selected from the job queue for processing by a subsystem.

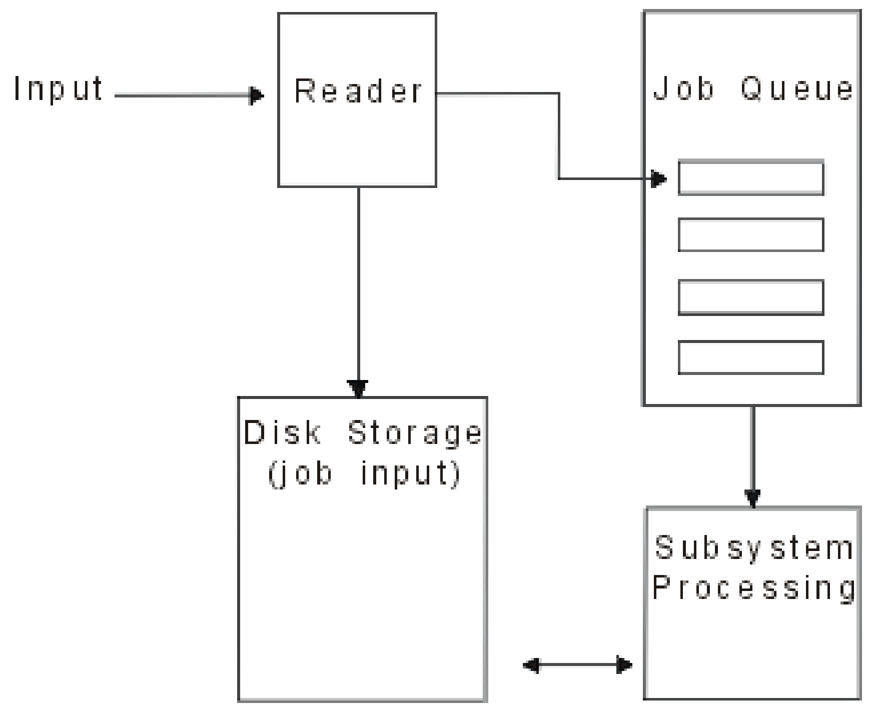

Figure 1. Relationship of input spooling elements

You can use the reader functions to read an input stream from diskette or database files.

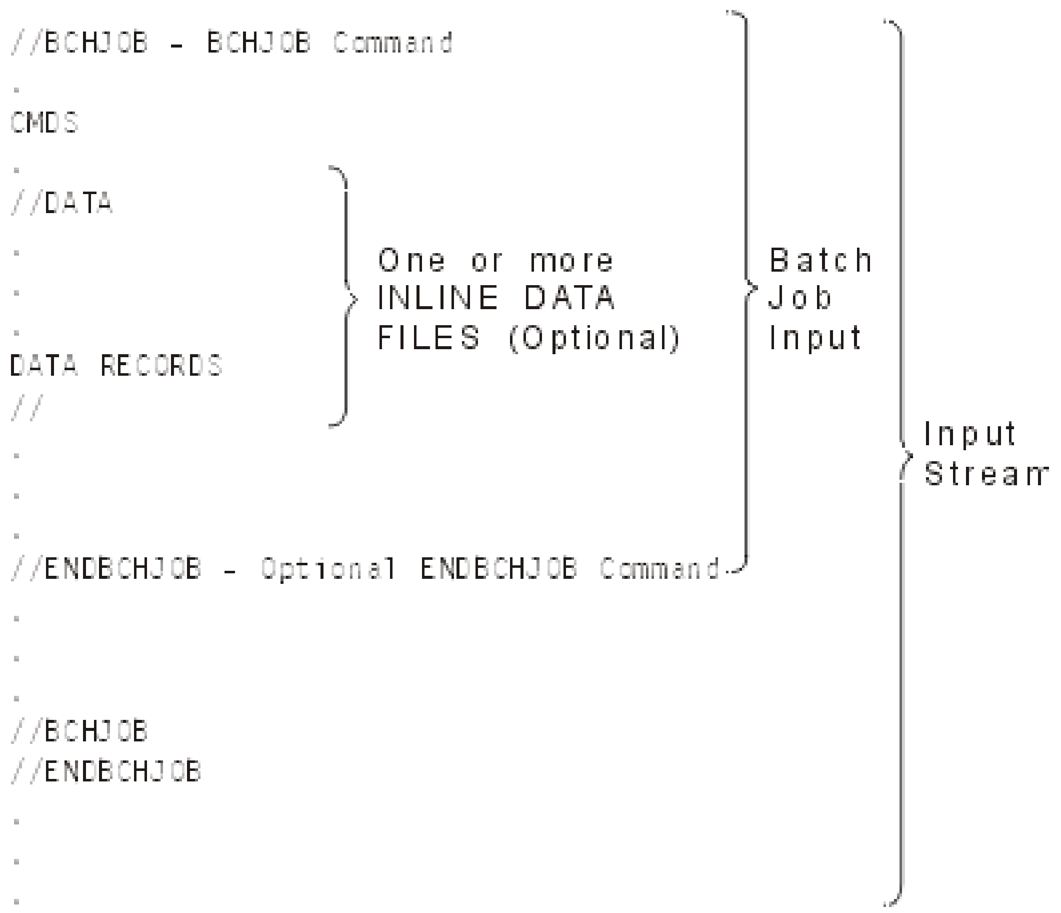
//BCHJOB - BCHJOB Command
.
CMDS
.
//DATA
.
.
.
DATA RECORDS
//
.
.
.
//ENDBCHJOB - Optional ENDBCHJOB Command
.
.
.
//BCHJOB
//ENDBCHJOB
.
.
.
One or more INLINE DATA FILES (Optional)
Batch Job Input
Input Stream

Figure 2. Typical organization of an input stream

The job queue on which the job is placed is specified on the JOBQ parameter of the Batch Job BCHJOB or the Start Database Reader STRDBRDR command, or in the job description. The values of the JOBQ parameter for the BCHJOB command follow:

- *RDR: The job queue is selected from the JOBQ parameter on the STRDBRDR command.
- *JOBD: The job queue is selected from the JOBQ parameter in the job description.
- A specific job queue: The specified queue is used.

For jobs with small input streams, you might improve system performance by not using input spooling. The Submit Job (SBMJOB) command reads the input stream and places the job on the job queue in the appropriate subsystem, bypassing the spooling subsystem and reader operations.

If your job requires a large input stream to be read, you should use input spooling (Start Diskette Reader STRDKTRDR or STRDBRDR command) so that the job can be imported independent of when the job is actually processed.

Job input commands

You can use these commands to submit jobs to the system. The start reader commands can be used for spooling job input; the submit job commands do not use spooling.

- Batch Job (BCHJOB): Marks the start of a job in a batch input stream and defines the operating characteristics of the job.
- Data (DATA): Marks the start of an inline data file.
- End Batch Job (ENDBCHJOB): Marks the end of a job in a batch input stream.
- End Input (ENDINP): Marks the end of the batch input stream.

- Submit Database Jobs (SBMDBJOB): Reads an input stream from a database file and places the jobs in the input stream on the appropriate job queues.
- Submit Diskette Jobs (SBMDKTJOB): Reads an input stream from diskette and places the jobs in the input stream on the appropriate job queues.
- Start Database Reader (STRDBRDR): Starts a reader to read an input stream from a database file and places the job in the input stream on the appropriate job queue.
- Start Diskette Reader (STRDKTRDR): Starts a reader to read an input stream from diskette and places the job in the input stream on the appropriate job queue.

Inline data files

An inline data file is a data file that is included as part of a batch job when the job is read by a reader or a submit jobs command. You use SBMDBJOB or STRDBRDR to queue a CL batch stream (stream of CL commands to be run). That CL batch stream can include data to be placed into inline data files (temporary files). When the job ends, the inline data files are deleted.

An inline data file is delimited in the job by a //DATA command at the start of the file and by an end-of-data delimiter at the end of the file.

The end-of-data delimiter can be a user-defined character string or the default of //. The // must appear in positions 1 and 2. If your data contains // in positions 1 and 2, you should use a unique set of characters, such as // *** END OF DATA. To specify this as a unique end-of-data delimiter, the ENDCHAR parameter on the //DATA command should be coded as:

```
ENDCHAR('// *** END OF DATA')
```

Note: Inline data files can be accessed only during the first routing step of a batch job. If a batch job contains a Transfer Job (TFRJOB), a Reroute Job (RRTJOB), or a Transfer Batch Job (TFRBCHJOB) command, the inline data files cannot be accessed in the new routing step.

An inline data file can be either named or unnamed. For an unnamed inline data file, either QINLINE is specified as the file name in the //DATA command or no name is specified. For a named inline data file, a file name is specified.

A *named inline data file* has the following characteristics:
- It has a unique name in a job. No other inline data file can have the same name.
- It can be used more than once in a job.
- Each time it is opened, it is positioned to the first record.

To use a named inline data file, you must either specify the file name in the program or use an override command to change the file name specified in the program to the name of the inline data file. The file must be opened for input only.

An *unnamed inline data file* has the following characteristics:
- Its name is QINLINE. (In a batch job, all unnamed inline data files are given the same name.)
- It can only be used once in a job.
- When more than one unnamed inline data file is included in a job, the files must be in the input stream in the same order as when the files are opened.

To use an unnamed inline data file, do one of the following:
- Specify QINLINE in the program.
- Use an override file command to change the file name that is specified in the program to QINLINE.

If your high-level language requires unique file names within one program, you can use QINLINE as a file name only once. If you need to use more than one unnamed inline data file, you can use an override file command in the program to specify QINLINE for additional unnamed inline data files.

Note: If you run commands conditionally and process more than one unnamed inline data file, the results cannot be predicted if the wrong unnamed inline data file is used.

Considerations for opening inline data files

You need to consider these elements when you open inline date files.

- The record length specifies the length of the input records. (The record length is optional.) When the record length exceeds the length of the data, a message is sent to your program. The data is padded with blanks. When the record length is less than the data length, the records are truncated.
- When a file is specified in a program, the system searches for the file as a named inline data file before it searches for the file in a library. Therefore, if a named inline data file has the same name as a file that is not an inline data file, the inline data file is always used, even if the file name is qualified by a library name.
- Named inline data files can be shared between programs in the same job by specifying SHARE(*YES) on a create file or override file command. For example, if an override file command specifying a file named INPUT and SHARE(*YES) is in a batch job with an inline data file named INPUT, any programs running in the job that specify the file name INPUT shares the same named inline data file. Unnamed inline data files cannot be shared between programs in the same job.
- When you use inline data files, make sure the correct file type is specified on the //DATA command. For example, if the file is to be used as a source file, the file type on the //DATA command must be source.
- Inline data files must be opened for input only.

Spooled files and output queues

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

Spooling overview

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

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

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

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

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

Spooled file

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

Almost all application programs that generate printed output make use of the spooling support provided by the i5/OS operating system. The values SPOOL(*YES) and SPOOL(*NO) on the SPOOL parameter of a printer file determine whether spooling support is requested.

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

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

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

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 can be set from 08:00:00 to 17:30:00 hours. During this time, only spooled files of 100 or fewer pages are printed. After 5:30 p.m. any spooled file prints. Spooled files that are too large are placed in deferred status (*DFR) until they can be printed. See Controlling printing by spooled file size for more information about how to configure deferred printing.

Multiple output queues

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

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

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

  For example, each programmer can be given a separate output queue.
- Output of special system files

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

Output queue recovery

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

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

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

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

Recovery of user-created output queues

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

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

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

If the output queue resides in an independent ASP, then the spooled files are moved to output queue QSPRCLOUTQ in library QRCL*xxxxx*, where *xxxxx* is the independent ASP number of the primary independent ASP (for example, QRCL00033 if the primary independent ASP number is 33). The move operation is performed by the QSPRC*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 QSPRC00001 system job. The system job issues a completion message to the QSYSOPR message queue when all spooled files have been moved.

Spooled file cleanup

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

Spooled file recovery also starts immediately following a vary-on operation of an independent ASP group. Spooled file recovery is done under the system job QSPRC*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

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

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

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

For first-in-first-out (*FIFO) output queues, the date and time change to the current system date and time when:
- A spooled file is created by opening a device file.
- The output priority of the job that created the spooled file is changed.
- The status of the spooled file changes from non-RDY to RDY.

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

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

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

Data queue support

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

- **Data queue support on output queues**

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

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

  A spooled file can reach RDY status in these instances:

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

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

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

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

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

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

- **Environment variable data queue support**

  Using the Add Environment Variable (ADDENVVAR) or Change Environment Variable (CHGENVVAR) command, you can associate a data queue with a job or the system. As spooled files are created, the entries are logged in the data queue. Using the Receive Data Queue (QRCVDTAQ) API to receive information from the data queue, a user program can determine when a spooled file has been created by the job or by the system. Use the environment variable data queue support if you need to determine the identity of a spooled file that was stored under a QPRTJOB and was created by a job such as a remote command system job.

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

The command use is as follows:

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

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

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

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

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

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

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

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

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

Error conditions

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

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

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

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

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

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

Additional considerations

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

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

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

Record type 01 data queue entry format

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

Table 1. Record type 01 data queue entry format

Decimal Offset	**Hex Offset**	**Type**	**Description**
0	0	CHAR(10)	Function Identifies the function that created the data queue entry. The value for a spooled file is *SPOOL.
10	A	CHAR(2)	Record type Identifies the record type within the function. Valid values are: **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.

Table 1. Record type 01 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(8)	Job system name Identifies the name of the system on which the spooled file was generated.
80	50	CHAR(7)	Spooled file creation date Identifies the date on which the spooled file was created in CYYMMDD format (local system time).
87	57	CHAR(1)	Reserved
88	58	CHAR(6)	Spooled file creation time Identifies the time that the spooled file was created in HHMMSS format (local system time).
94	5E	CHAR(7)	Spooled file creation date in UTC Identifies the date on which the spooled file was created in CYYMMDD format.
101	65	CHAR(1)	Reserved
102	66	CHAR(6)	Spooled file creation time in UTC Identifies the time that the spooled file was created in HHMMSS format.
108	6C	CHAR(20)	Reserved

Record type 02 data queue entry format

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

Table 2. Record type 02 data queue entry format

Decimal Offset	Hex Offset	Type	Description
0	0	CHAR(10)	Function Identifies the function that created the data queue entry. The value for a spooled file is *SPOOL.
10	A	CHAR(2)	Record type Identifies the record type within the function. Valid values are: **02** A spooled file has been created and placed on the output queue.

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

Decimal Offset	Hex Offset	Type	Description
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.
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)	Auxiliary Storage Pool Identifies the Auxiliary Storage Pool where the spooled file was created.
112	70	CHAR(8)	Thread ID Identifies the thread of the job that created the spooled file.

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

Decimal Offset	Hex Offset	Type	Description
120	78	CHAR(10)	System name Identifies the name of the system on which the spooled file was generated.
130	82	CHAR(7)	Creation date Identifies the date on which the spooled file was created in CYYMMDD format (local system time).
137	89	CHAR(6)	Creation time Identifies the time that the spooled file was created in HHMMSS format (local system time).
143	8F	CHAR(1)	Reserved

Record type 03 data queue entry format

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

Table 3. Record type 03 data queue entry format

Decimal Offset	Hex Offset	Type	Description
0	0	CHAR(10)	Function Identifies the function that created the data queue entry. The value for a spooled file is *SPOOL.
10	A	CHAR(2)	Record type Identifies the record type within the function. The valid value is: **03** 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 3. Record type 03 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)	Auxiliary storage pool Identifies the auxiliary storage pool where the spooled file was created.
112	70	CHAR(8)	Thread ID Identifies the thread of the job that created the spooled file.
120	78	CHAR(10)	System name Identifies the name of the system on which the spooled file was generated.
130	82	CHAR(7)	Creation date in UTC Identifies the date in UTC on which the spooled file was created in CYYMMDD format.
137	89	CHAR(6)	Creation time in UTC Identifies the time that the spooled file was created in HHMMSS format.
143	8F	CHAR(57)	Reserved

Spooled file names

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

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

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

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

  causes the name of the spooled file to be REPORT1 instead of QSYSPRT.
- The OVRPRTF command was used and a different printer file is specified in the TOFILE parameter. For example, typing the following command:

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

  causes the spooled file to be called PRTF2 (the name of the printer file specified in the TOFILE parameter of the OVRPRTF command).
- Some IBM® applications might create spooled files that have names different from the printer files used to create them. Users have no control over spooled file names in this situation.

Spooled file security

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

In general, there are 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 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) can display, copy, or send a file on that queue. Even users with SPCAUT(*JOBCTL) on an operator-controlled (OPRCTL(*YES)) output queue cannot display, copy, move, or send spooled files they do not own.

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

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

- Spool control authority (SPCAUT(*SPLCTL)) in the user profile. The user must also have 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 about 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 system, the file is spooled for job 999999/TINA/QPRTJOB. Spooling the file for this user's job instead of the system job makes sure that user TINA owns the spooled file. Then, when she runs the Work with Spooled Files (WRKSPLF) command, the spooled file sent to her is shown.

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

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

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

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

- Using the Set Profile (QWTSETP) Security API to set the user profile to a different user and then create a new spooled file.

  Other applications that are running can use the QSPCRTSP and QWTSETP APIs resulting in additional QPRTJOB jobs on the system.
- Using the UNIX® SETGID API to create a spooled file for a different, current, or group user profile when SPLFOWN is set to *CURGRPPRF.
- Using the UNIX SETUID API to set the user profile to a different user and then create a new spooled file for that user.

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

Spooled file subsystem

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

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

Requests for writer jobs are placed on the QSPL job queue, and the next entry on the QSPL job queue is selected to run if:
- The number of active jobs is less than the QSPL subsystem attribute of MAXJOBS.
- The number of active jobs from the QSPL job queue is less than the MAXACT attribute for the job queue.

Spooled file library

The spooled file library (QSPL or 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 might take 24 KB of storage.

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

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

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

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

Spooled files in independent ASPs

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

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

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

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

The QSPMN*xxxxx* job has the following responsibilities:

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

The QSPRC*xxxxx* job has the following responsibility:

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

Note: There is one QSPMN*xxxxx* and one QSPRC*xxxxx* system job for each ASP group that is varied on.

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

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

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

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

The following CL commands have spooled file in independent ASP limitations related to S/36 support, operational assistant support, and library name space.
- Change Job (CHGJOB) command
- Change Writer (CHGWTR) command
- Copy Spooled File (CPYSPLF) command
- Hold Job (HLDJOB) command
- Reclaim Spool Storage (RCLSPLSTG) command
- Release Job (RLSJOB) command
- Work with Job (WRKJOB) command
- Work with Spooled Files (WRKSPLF) command

Managing spooled files

Spooled files management includes tasks such as holding a spooled file, releasing a spooled file, and moving a spooled file.

Displaying a list of spooled files

To display a list of spooled files (printer output), use either of these methods.

System i Navigator
1. Expand **Basic Operations**.
2. Click **Printer Output**.

   The default setting is to display all printer output associated with the current user. You can display other printer output by right-clicking **Printer Output** and then clicking **Customize this view** → **Include**.

Character-based interface

Use the Work with Spooled Files (WRKSPLF) command.

Displaying the contents of a spooled file

To display the contents of a spooled file (printer output), use either of these methods.

System i Navigator
1. Expand **Basic Operations**.
2. Click **Printer Output**.
3. Right-click the printer output file that you want to display.
4. Click **Open**.

Character-based interface

Use the Work with Spooled Files (WRKSPLF) command, and then use option 5 (Display).

Notes:

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

Displaying messages associated with a spooled file

To display messages associated with a spooled file (printer output), use either of these methods.

System i Navigator

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

Character-based interface

Use the Work with Spooled Files (WRKSPLF) command, and then use option 7 (Message).

Holding a spooled file

To temporarily prevent the spooled file (printer output) that you selected from printing, use either of these methods.

System i Navigator

1. Expand **Basic Operations**.
2. Click **Printer Output**.
3. Right-click the printer output file that you want to hold.
4. Click **Hold**.
5. Specify the hold options and click **OK**.

Character-based interface

Use the Work with Spooled Files (WRKSPLF) command, and then use option 3 (Hold).

Note: The character–based interface has the additional capability of being able to hold all spooled files with the same user, print device, form type, user data, or ASP with one action.

Releasing a spooled file

To release a spooled file (printer output) that is being held, use either of these methods.

System i Navigator

1. Expand **Basic Operations**.
2. Click **Printer Output**.
3. Right-click the printer output file that you want to hold.
4. Click **Release**.

Character-based interface

Use the Work with Spooled Files (WRKSPLF) command, and then use option 6 (Release).

Note: The character–based interface has the additional capability of being able to release all spooled files with the same user, print device, form type, user data, or ASP with one action.

Moving a spooled file

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

System i Navigator

1. Expand **Basic Operations**.
2. Click **Printer Output**.
3. Right-click the printer output file that you want to move.
4. Click **Move**.
5. Specify the name of the printer or output queue where you want to move the printer output and click **OK**.

Character-based interface

Use the Work with Spooled Files (WRKSPLF) command, and then use option 2 (Change).

Note: The character–based interface has the additional capability of being able to change an attribute for all spooled files with the same user, print device, form type, user data, or ASP with one action.

Deleting spooled files

To delete a spooled file (printer output), use either of these methods.

System i Navigator

1. Expand **Basic Operations**.
2. Click **Printer Output**.
3. Right-click the printer output file that you want to delete.
4. Click **Delete**.
5. Click **Delete** to confirm.

Character-based interface

Use the Work with Spooled Files (WRKSPLF) command, and then use option 4 (Delete).

Note: The character–based interface has the additional capability of being able to delete all spooled files with the same user, print device, form type, user data, or ASP with one action.

Converting a spooled file to PDF

To convert a spooled file to a PDF file, follow these steps.

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**.

Copying a spooled file to a physical file

To copy a spooled file to a physical file, use the Copy Spooled Files (CPYSPLF) command.

The original spooled file is not affected by the copy operation and can still be printed by the printer writer program. When copying a spooled file to a physical file, many device attributes cannot be copied.

Sending a spooled file to another user or system

To send a spooled file (printer output) to a remote system that is running TCP/IP or to another user on the Systems Network Architecture distribution services (SNADS) network, use either of these methods.

System i Navigator

1. Expand **Basic Operations**.
2. Click **Printer Output**.
3. Right-click the printer output file that you want to send.
4. Click **Send**.
5. Click **Send via TCP/IP** to send the printer output to a remote system that is running TCP/IP or click**Send via SNA** to send the printer output to another user on the SNADS network.
6. Specify the send options and click **OK**.

Character-based interface

| Use the Work with Spooled Files (WRKSPLF) command, and then use option 1 (Send) to send the printer
| output to another user on the SNADS network. To send a spooled file to another user using TCP/IP, use
| the Send TCP/IP Spooled File (SNDTCPSPLF) command.

Changing attributes of a spooled file

To change a spooled file's (printer output's) attributes, use either of these methods.

System i Navigator

1. Expand **Basic Operations**.
2. Click **Printer Output**.
3. Right-click the printer output file that you want to change.
4. Click **Properties**.
5. Specify the attributes or properties that you want to change and click **OK**.

Character-based interface

Use the Work with Spooled Files (WRKSPLF) command, and then use option 2 (Change).

Note: The character–based interface has the additional capability of being able to change an attribute for all spooled files with the same user, print device, form type, user data, or ASP with one action.

Restarting the printing of a spooled file

To restart printing a spooled file on a particular page, follow these steps.

System i Navigator

1. Expand **Basic Operations** → **Printer Output**.
2. Right-click the printer output file for which you want to restart printing and select **Properties**.
3. On the **General** tab of the Properties window, click **Restart Printing**.
4. On the window that displays next, specify the page information and click **OK**.

Suspending one spooled file and printing another

To temporarily stop the printing of one spooled file (printer output) and start the immediate printing of another spooled file (printer output), follow these steps.

System i Navigator

1. Expand **Basic Operations**.
2. Click **Printer Output**.
3. Right-click the printer output file that you want to begin printing next.
4. Click **Print next**. This printer output is moved to the top of the output queue.
5. Right-click the printer output file that is currently printing.
6. Click **Hold**.
7. Specify to hold **At end of page** and click **OK**. This printer output stops printing at the end of the current page. The next printer output in the output queue starts printing.
8. Right-click the printer output file that is stopped.
9. Click **Print next**. This printer output is moved to the top of the output queue and will resume printing with the page following the last page printed.

Enabling spooled file notification message

To be notified when a spooled file (printer output) completes printing or is held by the printer writer program, you need to enable the spooled file notification function.

System i Navigator

1. Expand **Users and groups**.
2. Click **All users**.
3. Double-click the user name that you want to change.
4. Click **Jobs**.
5. Click **Display Session**.
6. Select **Send message to spooled file owner**.

Controlling the number of spooled files

The number of spooled files on your system should be limited. When a job is completed, spooled files and internal job control information are kept until the spooled files are printed or canceled. The number of jobs on the system and the number of spooled files known to the system increase the amount of time needed to perform IPL and internal searches, and increase the amount of temporary storage required.

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

| You can control the number of generated job logs with the LOG and LOGOUTPUT parameters on the
| Create Job Description (CRTJOBD) or Change Job (CHGJOB) commands, or with the QLOGOUTPUT
| system value. For more information, see Job log pending.

| You can specify the maximum number of spooled files generated by a job with the QMAXSPLF system
| value.

Deleting expired spooled files

You can use the EXPDATE or DAYS parameter on the Change Printer File (CHGPRTF), Create Printer File (CRTPRTF), Change Spooled File Attributes (CHGSPLFA), or Override with Printer File (OVRPRTF) command to make a spooled file eligible for deletion using the Delete Expired Spooled files (DLTEXPSPLF) command.

For example, the following command creates a job schedule entry which causes the DLTEXPSPLF command to delete all expired spooled files on your system every day:

```
ADDJOBSCDE JOB(DLTEXPSPLF) CMD(DLTEXPSPLF ASPDEV(*ALL)) FRQ(*WEEKLY) SCDDATE(*NONE) SCDDAY(*ALL)
SCDTIME(010000) JOBQ(QSYS/QSYSNOMAX) TEXT('DELETE EXPIRED SPOOLED FILES SCHEDULE ENTRY')
```

Reclaiming spooled file storage

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

For more information about spooled file storage, see Spooled file library.

Automatically clean up unused printer output storage (QRCLSPLSTG) system value

Use the Automatically clean up unused printer output storage (QRCLSPLSTG) system value to adjust the desired balance between spool performance and auxiliary storage. This system value can be used to clean up unused printer output storage on system auxiliary storage pools (ASPs), basic user ASPs and independent ASPs. For more information see, Storage system values: Automatically clean up unused printer output storage in the System values topic.

Note: System performance is degraded if Automatically clean up unused printer output storage (QRCLSPLSTG) is set to 0 days.

Assume that one of your application programs had an error and it produced thousands of spooled files that were of no value to you. When this happened those spooled files used lots of storage space on your system. To reclaim the spool storage, complete the following tasks:

1. Change the Automatically clean up unused printer output storage (QRCLSPLSTG) system value to 1.
2. Delete all the unwanted spooled files that the application program created. Note the time of day you deleted all the unwanted spooled files.
3. After 24 hours, provided the empty spooled file members are not reused, the system reclaims the auxiliary storage that was being used by the empty spooled files.
4. Change the Automatically clean up unused printer output storage (QRCLSPLSTG) system value back to its former value.

Reclaim Spool Storage (RCLSPLSTG) command

| Alternatively, you can immediately reclaim all empty spooled file members by using the Reclaim Spool
| Storage (RCLSPLSTG) command with the DAYS parameter set to *NONE. This command can be used to
| clean up unused printer output storage on system auxiliary storage pools (ASPs), basic user ASPs, and
| independent ASPs.

Notes:

1. Any unused database members are deleted immediately when running the RCPLSPLSTG command with the Days parameter set to *NONE. That means that there is no pool of unused members that can be used when creating spooled files.
2. Lock contention can occur on output queues or spool database files, resulting in bottlenecks and severe performance problems.

Assume that one of your application programs had an error and it produced thousands of spooled files that were of no value to you. When this happened those spooled files used lots of storage space on your system. To reclaim the spool storage, complete the following tasks:

1. Delete all the unwanted spooled files that the application program created.
2. Run the RCLSPLSTG command with the DAYS parameter set to *NONE. The system immediately reclaims all auxiliary spool storage that was being used by the unwanted spooled files.

System ASP storage

You can reduce the amount of storage taken up by spooled files by moving or creating spooled files directly into a user auxiliary storage pool (ASP) or independent ASP. You can accomplish this by specifying *OUTQASP on the SPLFASP parameter when creating an output queue in a library that is located in the user ASP or independent ASP.

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

Saving and restoring spooled files

You can use the SPLFDTA parameter on the Save Library (SAVLIB), Save Object (SAVOBJ), Restore Library (RSTLIB), and Restore Object (RSTOBJ) CL commands to save and restore spooled files without losing the print fidelity, attributes, or identity of the spooled files.

To maintain 100% print fidelity when you restore saved spooled files, you must:

- Save and restore all external resources within the libraries that they existed in when the spooled file was created.
- Make sure that the user profile (spooled file owner) exists and has the appropriate authorities to all the external resources required for the spooled file to print, including integrated file system directories, true type font integrated file system files, font resource objects, overlays, form definitions, page definitions, page segments, and embedded object integrated file system files.
- Make sure that the Resource Allocation Table (RAT) has the same fonts and linked fonts as when the spooled file was created. This is only necessary for those spooled files that use true type linked fonts.
- Make sure that all of the true type fonts that are used by the spooled files are in the appropriate directories on your system.
- Make sure that the environment variable QIBM_AFP_RESOURCES_PATH is set to the appropriate path for those spooled files that use embedded objects but were not directory qualified.
- Restore all embedded objects used by spooled files into the same directories that they were in when the files were created.

When a spooled file is in the process of being saved or restored, another restore or save operation will be blocked. The blocked operation might fail to save or restore that spooled file. A diagnostic message will be issued when this occurs.

When a spooled file is restored, it is reattached to the original job if the original job still exists on the system. If the original job does not exist, then the spooled file is restored in a detached state. If the spooled file is restored in a detached state, it is possible to have more then one spooled file with the same fully qualified job name, spooled file name, and spooled file number. Under these conditions, access to the restored spooled file requires the user or application to include the job system name and spooled file creation date. This enables the operating system to select the correct spooled file.

The order of spooled files that you restore is determined mainly by attributes such as the status of the file and is not necessarily the order in which you restore them. For more information, see Order of spooled files on an output queue.

For information about the procedures that were used to save and restore spooled files before V5R4 of i5/OS, see Save spooled files in the Systems management topic collection.

Controlling printing by spooled file size

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

For example, assume that you want to restrict spooled files with more than 40 pages from printing between 8 a.m. and 4 p.m. on output queue MYOUTQ. Between noon and 1 p.m. you want to allow spooled files with 10 pages or less to print. The following command implements these restrictions:

```
CHGOUTQ OUTQ(MYOUTQ) MAXPAGES((40 0800 1600) (10 1200 1300))
```

Appendix. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, 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

Programming interface information

This Spooled files publication documents intended Programming Interfaces that allow the customer to write programs to obtain the services of IBM i5/OS.

Trademarks

Terms and conditions

Printed in USA