IBM System/3
Model 15
User's Guide to Spooling

GC21-7632-5
File No. S3-30

IBM System/3
Model 15
User's Guide to Spooling

GC21-7632-5
File No. S3-30

Preface

This publication is designed as a tutorial aid for IBM System/3 Model 15 system operators; it can also be used as a reference manual for spooling support. In general, the programmer or analyst does not need to be concerned with spooling functions and operations; however, with an understanding of spooling, he or she can alter the application to achieve optimal results.

These topics are described in this publication:

- Spooling concepts
- Processing with spooling
- Processing without spooling
- The relationship of spooling to other system facilities
- OCC (operator control commands) used with spooling
- Spooling implications for OCL (operation control language)
- Spooling performance
- Spooling problem situations

In order to understand the text, the reader must be familiar with the operation of the System/3 Model 15 or with a System/3 Model 10 Disk System. *IBM System/3 Model 15 Introduction,* GC21-5094, describes Model 15 concepts. Other related publications are:

- *IBM System/3 Model 15 Operator's Guide*, GC21-5075
- *IBM System/3 Model 15 System Control Programming Reference Manual*, GC21-5077 (for 5704-SC1)
- *IBM System/3 Model 15 System Control Programming Concepts and Reference Manual*, GC21-5162 (for 5704-SC2)
- *IBM System/3 Model 15 System Generation Reference Manual*, GC21-7616
- *IBM System/3 Model 15 System Messages*, GC21-5076
- *IBM System/3 Model 15 3340 Direct Access Storage Facility Reference Manual*, GC21-5111
- *IBM System/3 3741 Reference Manual*, GC21-5113
- *IBM System/3 Model 15 Master Index*, GC21-5202

Sixth Edition (September 1979)

This is a major revision of, and obsoletes, GC21-7632-4 and technical newsletter GN21-5633. Changes or additions to the text and illustrations are indicated by a vertical line to the left of the change or addition.

This edition applies to version 07, modification 00 of the IBM System/3 Model 15 System Control Program 5704-SC1; to version 04, modification 00 of the IBM System/3 Model 15 System Control Program 5704-SC2; and to all subsequent versions and modifications until otherwise indicated in new editions or technical newsletters.

Changes are periodically made to the information herein; changes will be reported in technical newsletters or in new editions of this publication.

Use this publication only for the purposes stated in the *Preface*.

Publications are not stocked at the address below. Requests for copies of IBM publications and for technical information about the system should be made to your IBM representative or to the branch office serving your locality.

This publication could contain technical inaccuracies or typographical errors. Address your comments about this publication to IBM Corporation, Publications, Department 245, Rochester, Minnesota 55901. IBM may use and distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

Contents

Chapter 1. Introduction

Spooling is an optional facility of the system control programs (Program Numbers 5704-SC1 and 5704-SC2) for System/3 Model 15.

In addition to the minimum system configuration for System Control Program 5704-SC1, spool requires 5445 Disk Storage Model 1, 2, or 3 when using a System/3 Model 15A. For information concerning the minimum system configurations for the system control programs (5704-SC1 and 5704-SC2), see one of the following publications:

- *IBM System/3 Model 15 System Generation Reference Manual*, GC21-7616

- *IBM System/3 Model 15 Introduction*, GC21-5094

Although partition 3 (P3) and 3344 Direct Access Storage are discussed throughout this manual, they are supported only by System Control Program 5704-SC2.

A comparison of possible disks for the spool file is given in Figure 1-1.

	Spool File Disk Types			
	5445	3340 Main Data Area		3344 Main Data Area
System Control Program Number	5704-SC1	5704-SC1	5704-SC2	5704-SC2
Possible Unit Codes	D1, D2, D3, D4	D1, D2, D3, D4	D1, D2, D3, D31, D4, D41	D3, D4, D31, D32, D33, D34, D41, D42, D43, D44
Maximum Size in Cylinders	199	166	166	186

Figure 1-1. Comparison of Possible Disks for the Spool File

With 5704-SC2, the spool file copy program ($QCOPY) can be used to copy entries to and from the spool queues. Refer to the *IBM System/3 Model 15 System Control Programming Concepts and Reference Manual*, GC21-5162, for information about $QCOPY.

When directly attached to the Model 15, the 3741 Data Station Model 1 or 2 or 3741 Programmable Work Station Model 3 or 4 can be used as the spooled input device. When this option is selected, 1- through 128-byte records can be read. Spooling will truncate (on the right) or pad with blanks (on the right) the 3741 records to fit the user program's requested length. In this publication, references to card input also apply to input from the directly attached 3741.

It is possible to spool unit-record devices used in a program partition that is performing teleprocessing operations. It is not possible to spool readers, punches, or printers attached to terminals that are connected to the system using the binary synchronous communications adapter, multiple line terminal adapter, or local communications adapter.

The IBM System/3 Model 15 is a two- or three-partition multiprogramming system. (SCP 5704-SC1 supports two partitions; SCP 5704-SC2 supports three partitions.) Spooling, an optional facility of the system control program, can improve the throughput of an application. System/3 Model 15 offers a variety of operating environments:

- One-, two-, or three-batch partitions, each with or without spooling

- Combinations of batch and communications, each with or without spooling

Spooling support for the Model 15 facilitates handling unit-record input and output at disk I/O speed, thereby increasing total system throughput.

With spooling, a job is stored on disk. When the processing unit becomes available, the job is transferred from disk to a program partition. As the job executes, its printed and/or punched output is stored on disk. Upon operator request, the output may be printed and/or punched while other jobs are executing; thus, the job executes without having to wait for the slow unit-record devices.

Whenever a spooled reader, card punch, or printer becomes inoperative, the system continues processing jobs already on disk. When the I/O unit becomes operative again, reading, punching, or printing continues without a loss of output or processing time.

With spooling, programs can be run in different partitions without the need for separate unit-record devices for each partition. The printer, for example, can be used by multiple partitions if it is being serviced by the spooling function.

Performance of a spooling system varies widely. In some cases, such as applications with short programs or with small amounts of unit-record I/O, spooling may offer little or no advantage. In other cases, such as those in which unit-record device contention is a problem, spooling can provide substantial benefits in overall application throughput.

LEVELS OF SUPPORT

A user need not spool all unit-record I/O activity; during system generation, one of the following four levels of I/O support can be selected:

- Print spooling
- Print and punch spooling
- Input and print spooling
- Input, print, and punch spooling

The following unit-record devices can be spooled:

- 5424 MFCU
- 2560 MFCM
- 1442 Card Read Punch
- Directly attached 3741 Data Station/Programmable Work Station (input only)
- 2501 Card Reader
- 1403 Printer

The combination of devices that can be used with the different levels of spooling support is shown in Figure 1-2. The user also chooses the program partitions to be supported by spooling: partition 1 only, partition 2 only, partition 3 only, or any combination of partitions.

This publication assumes that you have selected input, print, and punch spooling.

		LEVEL OF SUPPORT																			
		Print	Print/Punch			Input/Print					Input/Print/Punch										
Printer	1403	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Punch Device	MFCU		X								X	X	X	X							
	MFCM			X											X	X	X				
	1442				X													X	X	X	X
Input Device	MFCU					X					X							X			
	MFCM						X								X						
	1442							X				X							X		
	2501								X				X			X				X	
	3741									X				X			X				X

Figure 1-2. Combination of Devices Available for Different Levels of Spooling Support

CONCEPTS OF SPOOL OPERATION

To handle a system's unit-record I/O without depending on I/O device speeds, spooling support utilizes disk space on 5445 Disk Storage or on the main data area of a 3340 Direct Access Storage Facility or 3344 Direct Access Storage as intermediate storage for areas known as queues (see Figure 1-3). As job streams (including operation control language statements, programs, and data) for the individual partitions are read, a spooling routine stores each job in the reader queue; this routine is called the reader.

After a job has been completely read and stored in the reader queue, it is ready to be transferred to the appropriate partition for execution. All records appear to the partition as though they were just read from the physical input device. When all job steps in the job have been executed, the spooling support removes the job from the reader queue.

Printed and punched output is also stored temporarily on disk in queues. Spooling routines called the print writer and punch writer enable printing and/or punching of the data in the print queue and punch queue upon operator request (see Figures 1-4 and 1-5).

ADVANTAGES OF SPOOLING

The preceding concepts of spooling operation point out these advantages of spool support:

- Asynchronous operation
- Improved system performance
- Reduced device contention

Asynchronous operation. Reading the input records from the spooled reader to disk, printing the spooled print records on the printer, and punching the spooled punch records on the card punch are functions of spooling that occur independently of programs executing in the partitions.

Improved system performance. The execution time of any one job is likely to be decreased because unit-record operations are performed at disk I/O speeds. Spooling writes blocked input and output records on disk, further increasing system throughput because the number of disk I/O operations is less than the equivalent number of unit-record operations.

Reduced device contention. Because more than one partition can reference the same device at the same time, one partition does not have to wait for the other partitions to free the device.

USER CONTROL

Execution of the spooling function is controlled mainly by the system operator through OCC (operator control commands) but also by the system programmer through OCL (operation control language).

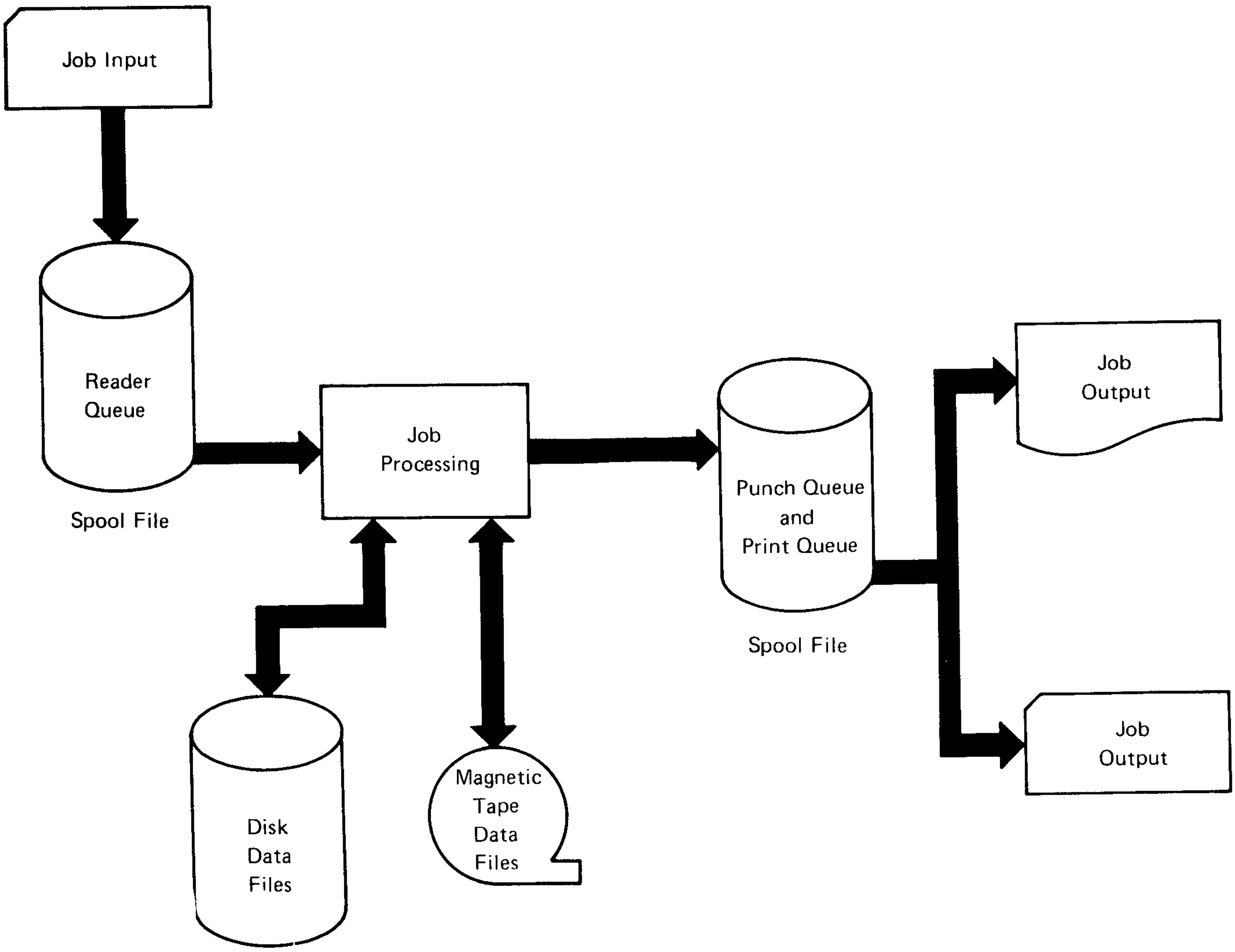
The operator controls system spooling activity by entering, through the CRT/keyboard, OCC to monitor, control, and communicate with the spooling routines. Through operator control, the spooled reader and output writers can start and stop independent of program execution. For example, the operator can hold job input in the reader queue for execution at a later time, allowing jobs that need more execution time to be held until the system is less occupied. Or, the operator can hold job output in the punch and print queues until it is convenient to punch or print it.

The system programmer can make efficient use of the system in defining the jobs and supplying OCL for each job. For example, if one job (D) spools a large volume of print and/or punch output, it is advantageous to complete that job before executing a series of processing-oriented jobs (A, B, C):

Job	**Characteristics**
A	Processing-oriented
B	Processing-oriented
C	Processing-oriented
D	I/O oriented

If the output writers are started while the processing-oriented jobs are executing, the processing unit and the I/O devices can be used simultaneously.

An example of OCL spooling control is the DEFER-NO parameter on the PRINTER and PUNCH statements. When the system programmer includes this statement, the operator can start the print or punch writer while the job step producing the output is still executing.

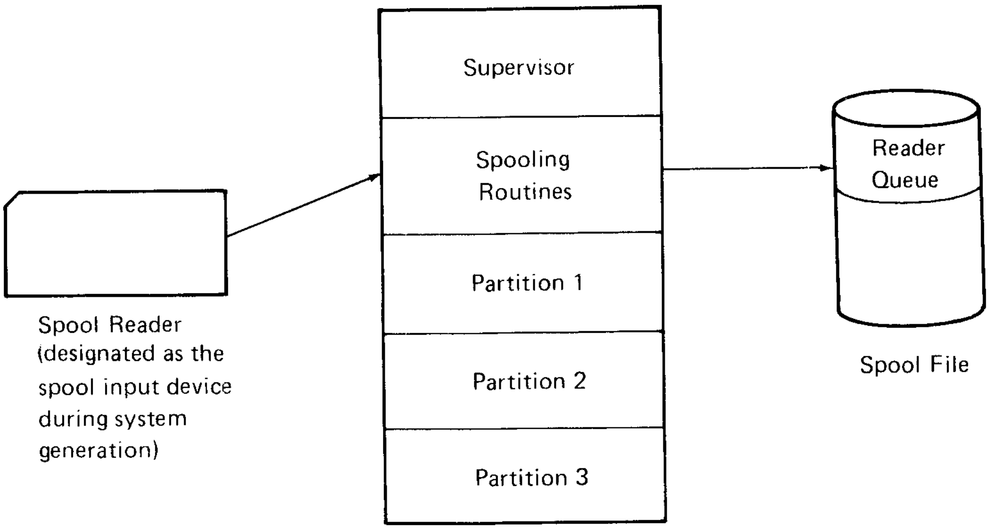


Note: User files on disk and magnetic tapes are not handled by the spooling function.

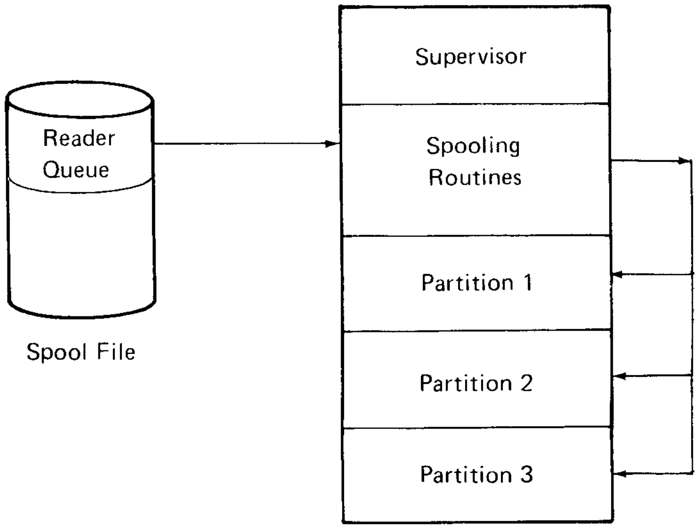
Figure 1-3. Processing with Spooling

The JOB statement is read from the spooled reader and queued on the disk before the start of a job. Similarly, the job output is stored on the disk in punch and/or print queues; punching and printing occurs later.

Note: A single disk contains the reader, print, and punch queues.

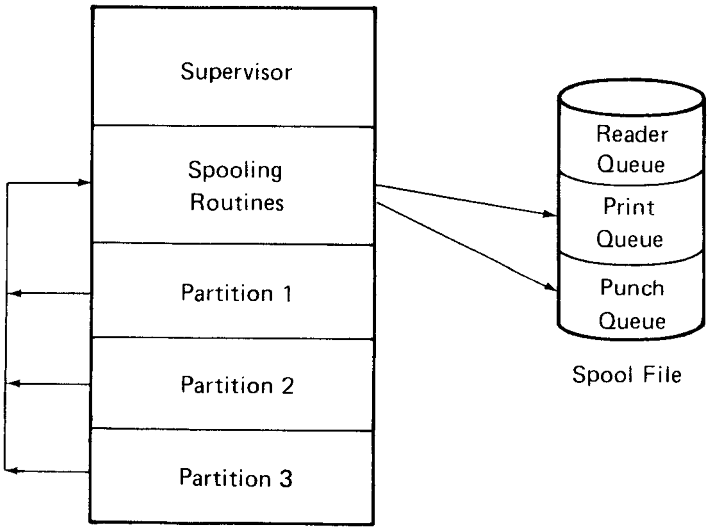


Jobs are read by the spool reader and stored on disk.

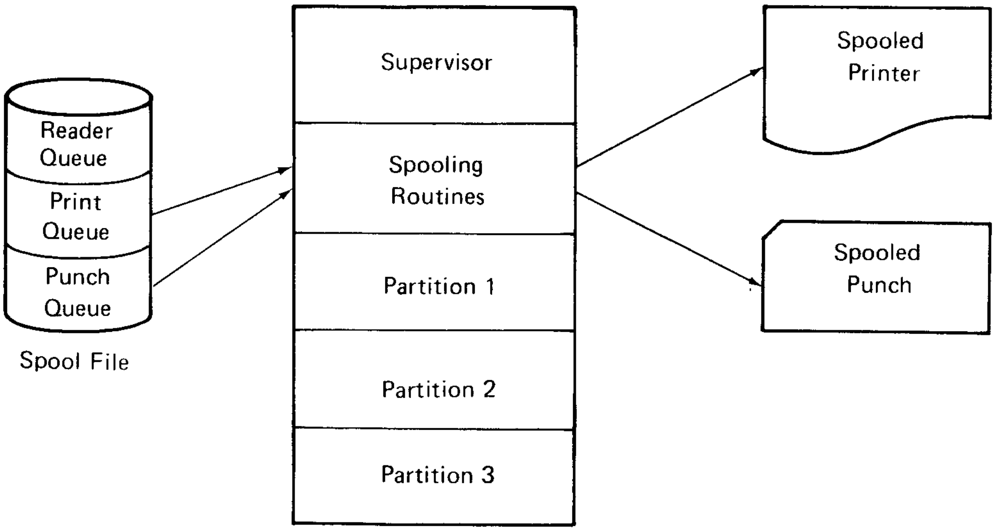


Jobs are transferred from the reader queue to one of the partitions, as required.

Figure 1-4. Spooling Operation



As the jobs are executed in the partitions, their output is stored in the spool file.



Job output is punched or printed at a later time, independent of jobs currently executing in the partitions.

Figure 1-5. Spooling Operation

This chapter covers the following spooling topics:

- Generating a spooling system
- IPL (initial program load)
- Reading (input spooling)
- Job execution
- Printing
- Punching

GENERATING A SPOOLING SYSTEM

During system generation, the user chooses the spooling support needed according to these considerations:

Amount of main storage available
Disk space available
Job stream characteristics
Desired throughput

One of the following four levels of spooling support can be selected during system generation:

Print spooling
Print and punch spooling
Input and print spooling
Input, print, and punch spooling

The level of spool support determines which unit-record devices can be spooled. The selected level can apply to partition 1, partition 2, partition 3, or any combination of partitions. If input spooling is chosen during system generation, one of the following input devices must be selected as the device from which spooled input will be read:

MFCU1 (MFCU primary hopper)
MFCU2 (MFCU secondary hopper)
1442
MFCM1 (MFCM primary hopper)
MFCM2 (MFCM secondary hopper)
2501
3741 (directly attached)

If punch spooling is chosen during system generation, one of the following punch devices must be selected as the device on which spooled output will be punched:

MFCU1 (MFCU primary hopper)
MFCU2 (MFCU secondary hopper)
1442
MFCM1 (MFCM primary hopper)
MFCM2 (MFCM secondary hopper)

The 1403 printer is always supported whenever spooling is specified, and it is the only printer that can be spooled. For 5704-SC2 only, an additional 1403 printer can be selected during system generation and used by a second spool print writer. See Chapter 9 for information on the second printer.

Figure 1-2 shows the combinations of unit-record I/O devices that can be used with the different levels of spooling support.

In addition to selection of the level of spooling support and the spool I/O devices, the following spool information must be provided during system generation:

Spool disk unit
Spool disk space
Track group size
Partitions with spool support
Forms type and card type
Autostart option
Autowrite option
Extended restart option (5704-SC2 only)
Time recording option
System input device
Main storage size

Spool Disk Unit

Only one disk unit (see Figure 1-1) can be specified to contain all spooled data; the spool area cannot be extended to another disk. The unit assignment can be changed during IPL using the START SPOOL command (see Chapter 5).

Spool Disk Space

The amount of space (in number of cylinders) allowed for spooled records must be specified with a maximum of 199 cylinders on a 5445 disk, 166 cylinders on a 3340 main data area, or 186 cylinders on a 3344 main data area indicating full pack capacity. The space is obtained on the specified disk unit during IPL and reserved for use by spooling under a filename ($SPOOL) in the VTOC. [The $SPOOL file is a special type of file that cannot be accessed by data management; it is listed when you run the File and Volume Label Display Program ($LABEL).] Once the spool area is full, space must be freed in order to continue. The amount of disk space can be changed during IPL by using the START SPOOL command (Chapter 5). The following items should be considered when specifying the size of the spooling area (Appendix C contains formulas for estimating the size of the spool area):

Level of spooling function
Mode of operation
Type of printing and punching
Volume

Level of Spooling Function: A system with only print spooling requires only disk space to contain the print records produced, whereas a system with input/print/punch spooling support requires disk space to contain the input records read and the print and punch records created.

Mode of Operation: When the spooled output of a job step is completely punched or printed, the space in the spool area is available for other output records. Thus, if the purpose of spooling is to save output until a later time (such as second shift), a larger spool area is required than if the records are punched/printed as available.

Type of Printing and Punching: Output records containing few characters require less disk space than records containing many characters. This is because spooling support truncates rightmost blanks when storing output records on disk. For example, Job A punches columns 1-20, and Job B punches columns 11-20 and 30-70. Each record written by spool to disk for Job A contains the 20 characters to be punched, while each record for Job B contains 70 characters.

Volume: Jobs that have few input records or produce only a few output records require less disk space than jobs that have a large volume of input data or create a large volume of output records.

Track Group Size

The disk space for the spool file is segmented into equal size groups of tracks. Each track group represents an extent obtained by spooling each time space is needed. The allowed track group sizes are 1, 2, 4, 5, and 10 tracks. The smaller sizes minimize unused areas within spooling disk space; the larger sizes improve performance for jobs with large volumes of output by minimizing the number of times spooling is required to obtain additional disk space (see Appendix D).

Partitions with Spool Support

If single partition support is desired, the partition (1, 2, or 3) to be spooled must be specified. When multiple-partition spool support is chosen, the support is the same in each partition.

Forms Type and Card Type

The forms type and card type options supply the default forms type and card type that will be used for spooled printed and punched output. An alternate forms type or card type may be supplied on the PRINTER or PUNCH OCL statements at execution time or by using a CHANGE command after execution has completed.

Autostart Option

The autostart option automatically starts the reader, print writer, and/or punch writer when spooling is activated at IPL. This option relieves the operator of having to enter the START RDR, START PRT, and START PCH commands at IPL; however, once the reader has been terminated, a START RDR command is required to activate the reader again. Likewise, once the writer has been terminated, a START or RESTART command is required to activate it again. Because the punch writer and reader cannot be operating simultaneously, both cannot be automatically started when input and punch spooling are supported.

Note: If the autostart option was selected, the print writer or punch writer begins automatically at IPL if there are records already in the queues. If there are no records in the queues, the output writers are started by the operator response to a message that the queues are empty or held.

Autowrite Option

The autowrite option reduces the interaction between spooling and the operator. When the print writer or punch writer starts, it automatically prints or punches whenever output is available on the queue. When an output queue is held, all steps on the queue are held, or when the queue is empty, no message is written to the CRT informing the operator of the situation. No operator action is necessary to cause the writer to begin execution again. The writer merely waits until output is available and begins automatically.

A job step specifying that the output may be printed or punched during execution causes the writer to begin printing or punching if there is no output on the queue ahead of the executing job step.

The writers can still be stopped and started by using OCC. When stopped, the writer does not automatically begin when output is available; the operator must start the writer before output can begin.

Extended Restart Option (5704-SC2 only)

The extended restart option automatically restarts some system messages; two such spool messages are NO INPUT JOB AVAILABLE and RDRQ IS EMPTY/HELD.

Whenever a job is placed on the reader queue, all NO INPUT JOB AVAILABLE and RDRQ IS EMPTY/HELD messages will be restarted if the new job can be run in a partition that is halted by one of these messages. The messages will not be restarted if:
- The queue is held,
- The job is for another partition,
- The JOB statement has PRIORITY-0 specified, or
- The CORE parameter value specified on the JOB statement is too large.

For information on other system messages that can be automatically restarted, see the *IBM System/3 Model 15 System Control Programming Concepts and Reference Manual*, GC21-5162.

Time Recording Option

The time recording option causes a message to be written in the system history area each time the print or punch writer completes a job step. The message contains the date and time that the output for the job step was started and the time that the output was completed.

This message may be used as an aid for performance planning on a spooling system. For additional information about performance, see Chapter 7. The time recording message may also be used for job accounting purposes.

Note: If consistent elapsed times are expected, the output writers should have a system priority higher than that of the program partitions. For information about the PTY command, see Chapter 5.

System Input Device

A system input device is assigned to each partition; for example, the 2501 can be assigned for partition 1 and the CRT/Keyboard for partition 2. When the operator initiates a job for a partition, the partition reads OCL statements from its input device. Also, during system generation, if input spooling is to be used, an input device must be assigned as the spooled reader. The device assigned as the spooled reader can be the same device specified as the system input device.

If input spooling for the 2501 were chosen for partition 1, the 2501 should also be specified as the system input device. Jobs are initiated from the reader queue when the system input device is also the spool input device.

A partition's system input device can be changed by a READER OCL statement or by a READER command. Consider a situation where (1) the system input device for partition 1 is the 2501, (2) the spool input reader is the MFCM1, and (3) a job to be executed in partition 1 consists of RPG II programs that use the MFCM1 for card input data. The job is placed in MFCM1, the MFCM is readied, and the operator issues the START RDR command.

After the job has been read, the operator should not initiate job execution in partition 1 until the system input device for partition 1 is changed. This can be done with a RDR P1,MFCM1 command or by putting // READER MFCM1 in the 2501. Now partition 1 can be started. If the operator initiated partition 1 before changing the reader, the system would have attempted to read from the 2501, which could be empty, contain blank cards, or contain some other job.

Main Storage Size

The amount of main storage available for spooling should be taken into consideration based on the amount required to support the desired functions. The main storage requirements for the spooling functions selected are stated in Appendix B.

IPL

Spooling, selected as a feature during system generation, is loaded at IPL with the supervisor. Upon completion of the IPL process, the system date can be entered before processing continues. A display on the CRT contains EJ messages for the partitions. At this time, before initiating any jobs, the operator decides whether to activate or cancel the spool support. The CANCEL SPOOL command (see Chapter 5) causes the spooling support to be removed from main storage, making additional main storage available to any partition. If spooling support is desired later, the operator must perform an IPL again.

If the operator chooses to operate with spool, spooling can begin either by a command prior to job execution or by operator response to an IPL EJ message for a partition.

To start without any changes to the spool disk space, the operator begins job execution by replying to one of the IPL EJ messages or entering a START SPOOL command (see Chapter 5). Either action causes the disk space used by spooling to be obtained according to the disk unit and size specified during system generation. Any existing spooled input and/or output remains in the spool area and is not lost.

To override specifications made during system generation, the operator enters a START SPOOL command with the parameters to be changed. The operator may (1) alter the spool disk unit used, (2) change the size of the spooling area, (3) change the size of the track groups within the spooling area, or (4) reformat the spool area (causing existing spooled I/O data to be lost). Once spooling is initiated (at IPL), the disk pack used for the spooling area should not be removed until another IPL is performed.

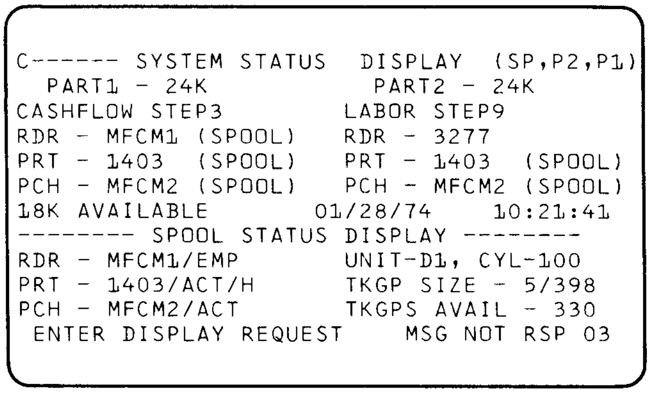
Note: If another IPL is performed while spooling is being activated, there is a chance that the spool disk space was not completely allocated for spooling. That is, the space may have been obtained but an entry was not yet made in the VTOC to account for the space. To determine whether this situation exists, either attempt to use the disk pack again for spooling or display the VTOC using $LABEL. If this situation occurs, the FORMAT function of $DELET will free the inaccessible space.

Activating spooling does not cause the reader, print writer, or punch writer to start unless autostart has been generated. If the autostart function was specified during system generation, those devices so specified are automatically started when spooling is activated. If autostart was not generated, the operator must enter START commands to initiate the desired spooling function.

Entering the START RDR command causes the spooled input device to start reading and placing the jobs in the reader queue. Entering the START PRT or START PCH commands activates the respective output writer so the printing or punching of spooled records from the print or punch queue can begin. If job steps have not put any records into the print or punch queues, a message (PRTQ IS EMPTY/HELD or PCHQ IS EMPTY/HELD) is shown on the display screen.

If the operator attempts to start either the reader or the punch writer after the other has started, a message appears on the CRT indicating READER/WRITER ACTIVE since both the reader and the punch writer cannot be operating at the same time. As soon as the initiated function terminates, the operator can start the other function.

At this point, it may be helpful for the operator to display the system status to determine that the system is set-up as desired. The DISPLAY command (see Chapter 5) with the STATUS operand shows the system status; for example, for 5704-SC1:



```
C------ SYSTEM STATUS DISPLAY (SP,P2,P1)
  PART1 - 24K          PART2 - 24K
CASHFLOW STEP3       LABOR STEP9
RDR - MFCM1 (SPOOL)  RDR - 3277
PRT - 1403  (SPOOL)  PRT - 1403  (SPOOL)
PCH - MFCM2 (SPOOL)  PCH - MFCM2 (SPOOL)
18K AVAILABLE      01/28/74    10:21:41
-------- SPOOL STATUS DISPLAY --------
RDR - MFCM1/EMP      UNIT-D1, CYL-100
PRT - 1403/ACT/H     TKGP SIZE - 5/398
PCH - MFCM2/ACT      TKGPS AVAIL - 330
 ENTER DISPLAY REQUEST   MSG NOT RSP 03
```

READING (INPUT SPOOLING)

Job Stream

Prior to executing any programs on the Model 15 using spooling support, the system programmer should consider the organization of the job stream. Although no changes in the programs are required; the following should be observed:

- Several additional OCL statements can be added.
- The order of program execution can be changed for better performance and device utilization.

The input stream consists of two different units of work: jobs and job steps. Each job is a group of related job steps to be executed in a sequential manner. A job includes all necessary programs, file definitions, and OCL statements needed to accomplish a specific function. When input spooling is used, the steps must consist of OCL or OCL plus data; you cannot spool only the data.

Defining a series of related steps as a single job may be required either because of the sequence of functions or because it may provide specific advantages. For instance, multiple steps within a single job are preferable when the execution of a later step is directly dependent upon the successful completion of an earlier step. If a step terminates abnormally, the remaining steps in the job are bypassed.

When using input spooling, it is also advantageous to consider the placement of jobs within the job stream. A job with a small amount of input should be placed on the queue before a job with a large amount of input, allowing the first job to begin execution while the second one is being read into the queue.

OCL Considerations

The beginning of a job's OCL is indicated by the JOB OCL statement or the statement after a /. statement. All OCL statements and data that follow the JOB statement, up to the next JOB or /. statement, are associated with that job. Any number of LOAD/RUN and/or CALL/RUN combinations can be included. Spooling uses the information from the JOB statement to build the queues on disk. For instance, the PARTITION parameter enables jobs for all partitions to be read from one system input device by specifying the partition designated for job execution.

The PUNCH and PRINTER OCL statements are also used by spooling support and can be included in the job stream (see Chapter 6).

A /. statement should be used after each job as a delimiter. (Although, if a /. statement is not used, the JOB statement for the next job acts as a job delimiter.) When the job being put in the reader queue is not separated from the next job by a /. statement, spooling generates a /. statement as the last statement for the job in the queue. When the system history area is displayed or printed, or when the statement is logged on the system logging device, this statement may contain unpredictable characters following the /..

Two consecutive /. statements indicate the end of the entire job stream. When using input spooling, these two consecutive /. statements cause the reader to be terminated, releasing the input device.

Performance Considerations

To improve performance, the following items should be considered when planning a job stream:

1. When spooling input, make the first job one that contains little or no input data. The sooner a job is on the reader queue, the sooner a partition can be initiated to execute that job.

2. When spooling input, define the first job with a minimum number of job steps – preferably one. A partition cannot be initiated until the entire job is on the reader queue.

3. When executing in multiple partitions, avoid unnecessary disk contention caused, for example, by running two sort programs at the same time.

4. When not using punch spooling support, carefully plan the placement of job steps that require the punch device.

5. Avoid contention between the reader and the punch device and also between punch requests to the same device from multiple partitions at the same time.

Limitations of Input Spooling

The following limitations apply to any input being spooled from the job stream:

- Output operations (such as stacker selection or combined files) cannot be performed on spooled input files.

- The following record lengths for each device are used:
  a. From the MFCM, 2501, and 1442, only 80-byte records are written in the reader queue.
  b. From the MFCU, only 96-byte records are written in the reader queue.
  c. Other card sizes, such as 51-column cards, cannot be used.
  d. From the 3741, 1- to 128-byte records are written in the reader queue.

- The read card image (column binary) operation is not supported.

Reader Queue

Once the job stream has been constructed, the records have been placed into the reader, and IPL has been completed on a system with input spooling, the reader queue can be built on disk. If autostart was specified for input spooling during system generation, the reader starts reading records when spool is activated following an IPL. If autostart is not in effect, the operator must enter a START RDR command when jobs are to be read into the reader queue.

The reader queue is built on disk as the job stream is read (see Figure 2-1). To the spool reader, a JOB statement indicates the beginning of a new job; a /. or another JOB statement indicates the end of the job. One reader queue entry is created for each job read. All steps within a job are scheduled for sequential execution in the same partition. Each job is identified on the queue by the jobname. If any errors are found on the JOB statement, that job is assigned a priority of 5 on the readei queue. In addition, if the jobname is missing or invalid, the default name JOB is assigned to that job on the reader queue.

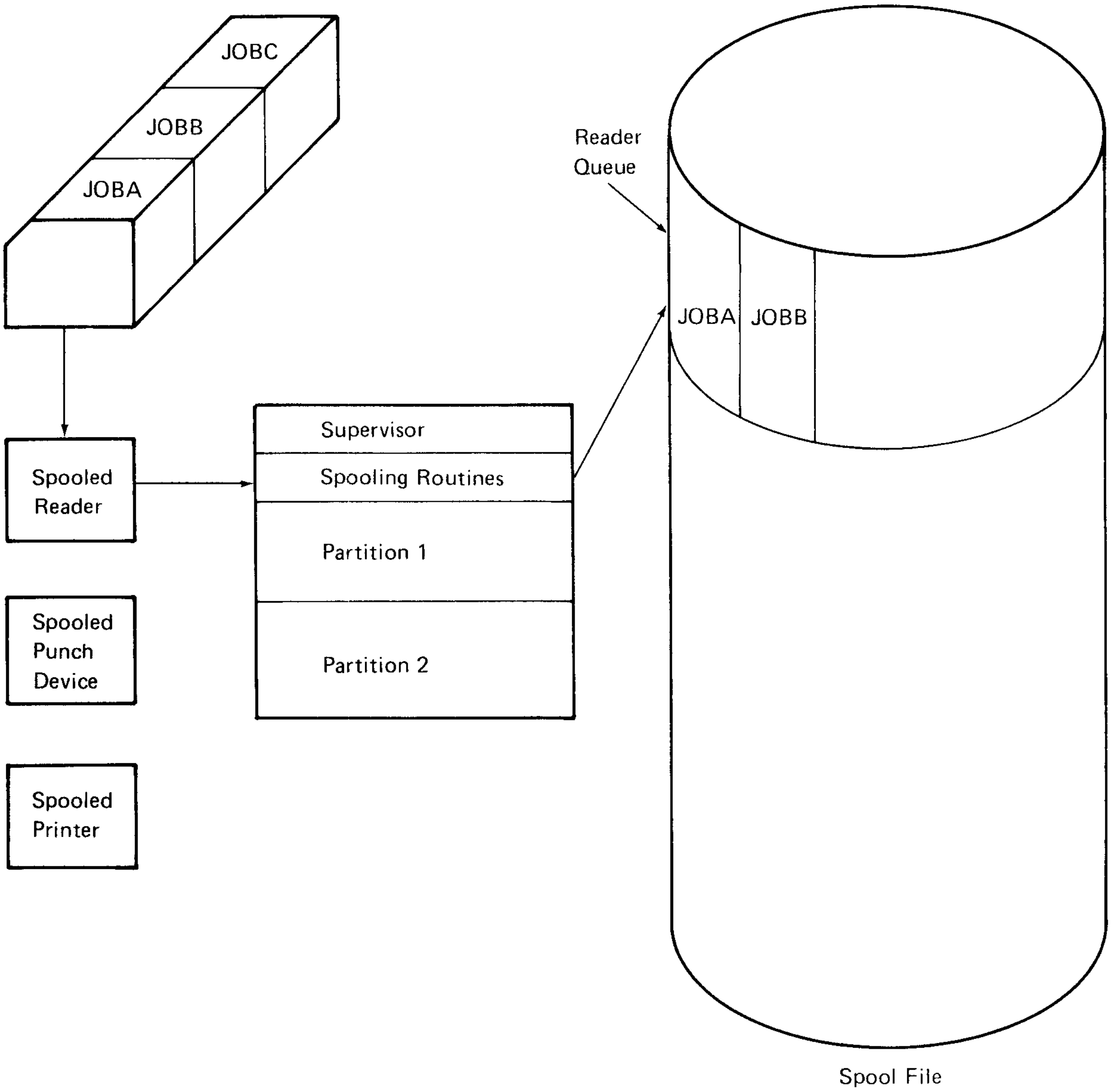


Figure 2-1. Building the Reader Queue

The spool reader continues reading jobs until (1) two /. statements are read consecutively in the job stream, (2) a STOP RDR command is entered to terminate the reader, or (3) a cancel option to a reader error message is taken. An informational message (SPOOL RDR TERMINATED) appears on the CRT to indicate that the spool reader has been terminated. Once the spool reader has terminated, a START RDR command is required to restart when additional jobs are to be added to the reader queue.

If jobs are intended to be placed on the reader queue from time to time without terminating the reader (for example, not providing two consecutive /. statements), the following operational considerations should be noted:

1. For the job to be placed on the reader queue, the /. statement must be read. Thus, the /. statement in the last job (or only job) to be read should be followed by a valid OCL statement.

2. When using the MFCM2, MFCU, 1442, 3741, or 2501, only one OCL statement is required. When using the MFCM1, two OCL statements are required.

3. If the next job is not yet available, any OCL statement that does not affect the job processing can be used (for example, a comment (*) statement).

The following example illustrates this concept:

```
//JOB1      JOB
      .
      .
      .
/.
//JOB2      JOB  This statement must be read before
      .          JOB1 is put on the reader queue.
      .
      .
//JOB3      JOB  This statement must be read before
      .          JOB2 is put on the reader queue.
      .
      .
/.
* END OF JOB3    If JOB3 is the last job to be put on
                 the reader queue and the reader is
                 not to be terminated, an OCL
                 statement must be provided here
                 to put JOB3 on the reader queue.
                 In this example, a comment state-
                 ment is used. (If the MFCM1 is
                 the input device, a second comment
                 statement should be used.)
```

If the spool reader is started, it is important to remember that the punch writer cannot be initiated until the reader is terminated.

Error Conditions

If hardware errors, such as feed checks, occur on the reader device, the operator should handle the error as though on a nonspooling system. When the situation is corrected, input spooling can be resumed. If the problem cannot be corrected and the operator cancels the operation, the spool reader is terminated and the job currently being read is not placed on the input queue. To execute the job, the spool reader must be started and the job must be reread entirely.

Operator Control

Operator control commands are used to manipulate the reader queue after it has been created (see Chapter 5). These commands are CANCEL, HOLD, RELEASE, CHANGE, and KEEP.

For 5704-SC2, the same commands can be used to manipulate multiple jobs by specifying, in place of the jobname, one to seven characters followed by two asterisks (**). Spooling will then perform the requested function on all jobs on the queue whose names begin with the specified characters. For example, if the HOLD RDRQ,ABC** command were entered, all jobs on the reader queue whose names begin with ABC would be held.

CANCEL: The entire reader queue may be canceled using CANCEL *RDRQ. This command is valuable when the operator realizes (after executing several job steps) that not all remaining jobs in the reader queue will execute correctly. All the jobs on the queue can be removed with a single command. If it is determined that only a particular job needs to be removed, the option RDRQ,jobname can be used. When it is necessary to cancel a job step on the reader queue, the entire job must be canceled.

A job currently executing in a partition cannot be canceled by simply entering CANCEL RDRQ,jobname. It must be canceled by specifying the partition; that is, CANCEL P1, CANCEL P2, or CANCEL P3.

HOLD: The HOLD command is used to prevent jobs from executing until a more convenient time. Spooling will not permit a job step to be selectively held on the reader queue; only an entire job can be held.

For example, if the next job on the queue requires being run on first shift and there isn't sufficient time to run it today, the operator may hold it for first shift tomorrow by specifying HOLD RDRQ,jobname. The HOLD command does not affect the priority of the job. If there is a need to hold the entire queue, the ooerator specifies HOLD RDRQ. This prevents all jobs on the reader queue from being scheduled for execution until the queue is released (using the RELEASE command).

RELEASE: The RELEASE command allows the operator to release:

- The reader queue
- A job on the queue that was placed in the hold state by a HOLD command or assigned a PRIORITY-0 on the JOB statement

If the entire reader queue was held, then the entire queue must be released; jobs cannot be selectively released.

In the previous example, on the second day, the operator releases the job for execution by specifying RELEASE RDRQ,jobname. The priority of the job is not affected when it is released.

CHANGE: The CHANGE command allows the operator to change the priority, partition assignment, or main storage requirement of a job on the reader queue.

The job priority can be increased or decreased, and the job in the reader queue is processed after all other jobs with the same priority have been processed. For example, the command CHANGE PTY,5,RDRQ,JOBA changes the current priority of JOBA to a priority of 5, and JOBA is scheduled for execution after all preceding priority 5 jobs on the reader queue have been executed. If the priority of a job is changed on the reader queue, the steps on the output queue(s) have the new priority, unless they contain a PRIORITY parameter on a PRINTER or PUNCH statement.

The partition assignment for the job on the reader queue can be changed. For example, the command CHANGE PTN,2,RQ,DAILY changes the partition in which the job DAILY is to be executed to P2.

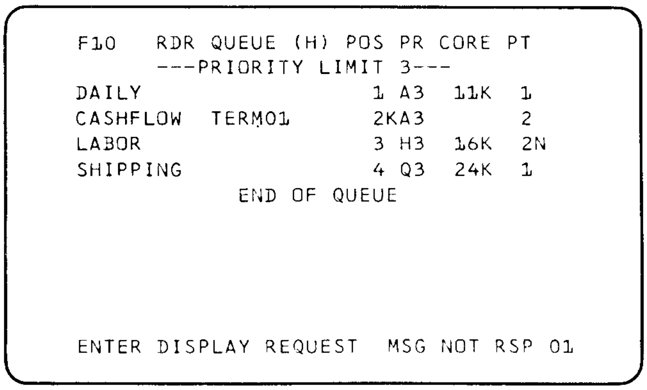
The main storage requirements of a job can be changed by specifying a number between the limits of 0 and 238 for 5704-SC1 or 0 and 488 for 5704-SC2 in the CHANGE core command. For example, the command CHANGE CORE, 48,RDRQ,DAILY changes the main storage requirement for the job DAILY to 48K. The command CHANGE CORE,0,RDRQ,DAILY removes any main storage requirement for the job DAILY and allows it to execute in any main storage size.

KEEP: The KEEP command is used to prevent a job from leaving the reader queue after all steps of the job have executed. Rather, the job remains on the queue in the hold state.

Displaying the Reader Queue

The DISPLAY RDRQ command can be very useful for the operator to monitor the reader queue (see Chapter 5).

The following is an example of how a display of the reader queue might appear. The information displayed for the reader queue is jobname, stepname, position on queue (POS), priority (PR), required main storage (CORE), and partition (PT). Main storage is given only when specified on the JOB OCL statement or through the CHANGE command. For 5704-SC2, jobs placed on the reader queue by $QCOPY under CCP will have the terminal name displayed in the stepname position.



```
F10   RDR QUEUE (H) POS PR CORE PT
      ---PRIORITY LIMIT 3---
DAILY                 1 A3  11K  1
CASHFLOW  TERM01      2KA3       2
LABOR                 3 H3  16K  2N
SHIPPING              4 Q3  24K  1
            END OF QUEUE




ENTER DISPLAY REQUEST  MSG NOT RSP 01
```

An (H) on the first line of the display indicates that the entire reader queue is held via the HOLD command.

For 5704-SC2, if the priority limit for jobs placed on the reader queue by $QCOPY under CCP is less than five, the limit is displayed on the second line of the display.

One or two of the following characters can precede the priority number in the display:

A – Indicates that the job is currently executing in the partition shown in the PT column (5704-SC2 only)
H – Indicates that the job is being held via the HOLD command
K – Indicates that the job is in a keep state via the KEEP command (5704-SC2 only)
Q – Indicates that the job is being copied by $QCOPY (5704-SC2 only)

For 5704-SC2, an N following the partition number indicates that QCOPY-NO was specified on the JOB OCL statement.

EXECUTING JOBS

After at least one job is placed in the reader queue, the operator can initiate job execution by responding to the end of job message displayed on the CRT for the partition. Spooled input records are read from the reader queue (on disk) and passed to the requesting partition; to the partition, it appears as though the records had been read from the card reader. All steps within a job are scheduled for sequential execution in that partition (see Figure 2-2).

Even though job execution has been started, the spooled reader does not stop. Execution within the program partition does not affect the spool reader, which continues to place jobs on the reader queue until (1) the operator enters a STOP RDR command, (2) the end of the job stream is reached, or (3) the operator selects a cancel option for a halt that is issued because of a reader device error. If a job is not in the keep state, it is removed from the reader queue when all steps of the job have been executed, or when the job is canceled before completion.

As job steps execute in the partitions, punched output is placed in the punch queue, and printed output is placed in the print queue (see Figure 2-3). The print and punch records are blocked (with trailing blanks removed) and written on their respective queues. Instructions for printing and punching the records (for example, forms length, spacing, card type) are also stored.

Identification on the Queue

The records are identified on the queue by jobname and stepname. If a stepname was not given, the records are identified by the program name as given on the LOAD statement, plus two appended digits that denote the step's position within the job. For example, JOBX with programs PRO1, PRO2, and PRO3 is identified as follows:

JOBX	PRO101
JOBX	PRO202
JOBX	PRO303

The records are identified by jobname and stepname, which allow the operator to control a job's output on a step basis.

For 5704-SC2, when the records on the queue are produced by a program running under CCP, the terminal name is generally used as the jobname and a stepname is generated by using the program name and a sequence number. The sequence numbers start at 01 and are increased by 1 for each job step placed on the queue. After 99, the sequence number is reset to 00. Separate number sequences are used for the print and punch queues.

For more information about the jobname and stepname used on the spool queues for CCP, see the *IBM System/3 Model 15 Communications Control Program System Reference Manual*, GC21-7620.

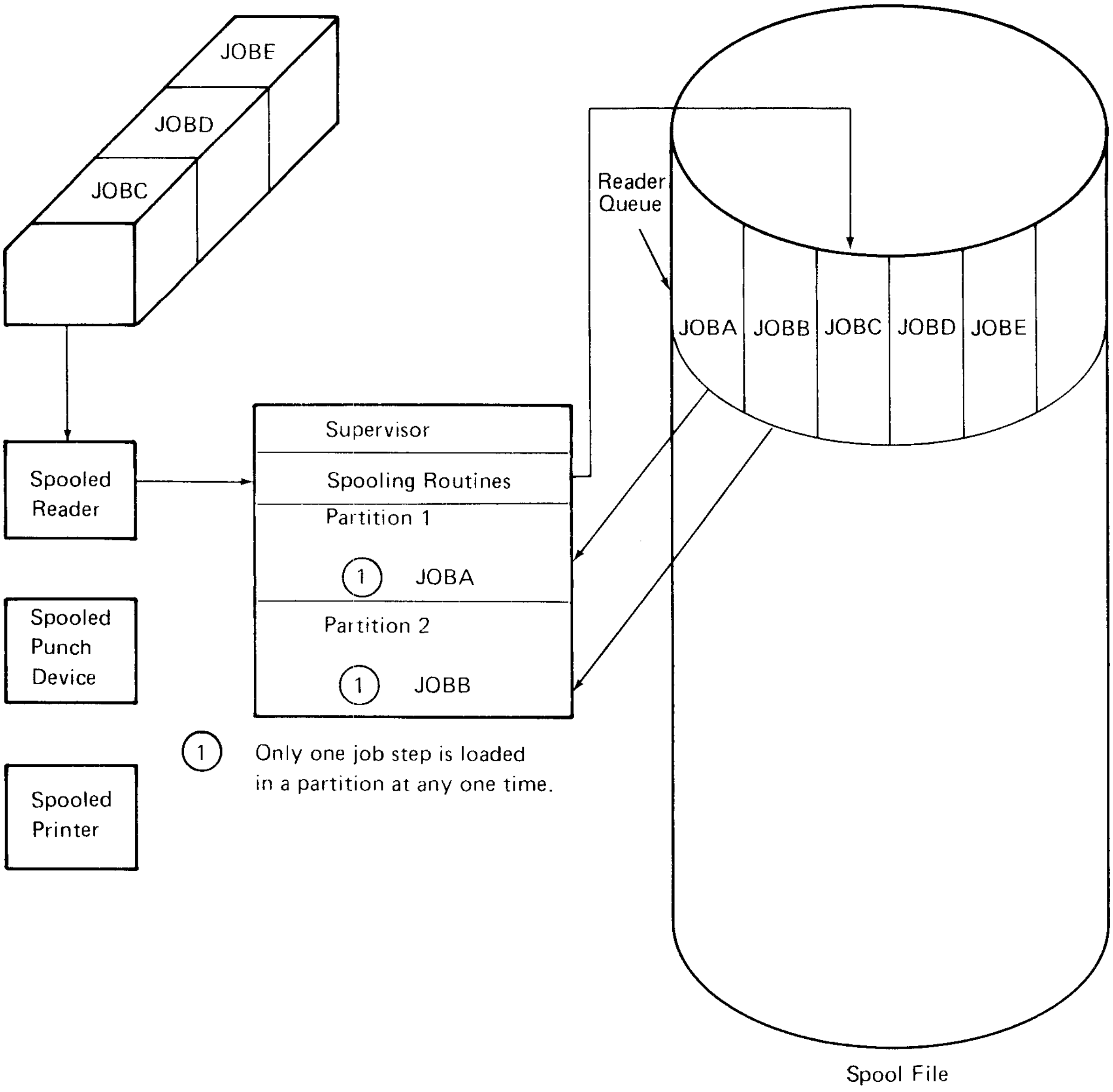


Figure 2-2. Initiating Jobs in the Partitions

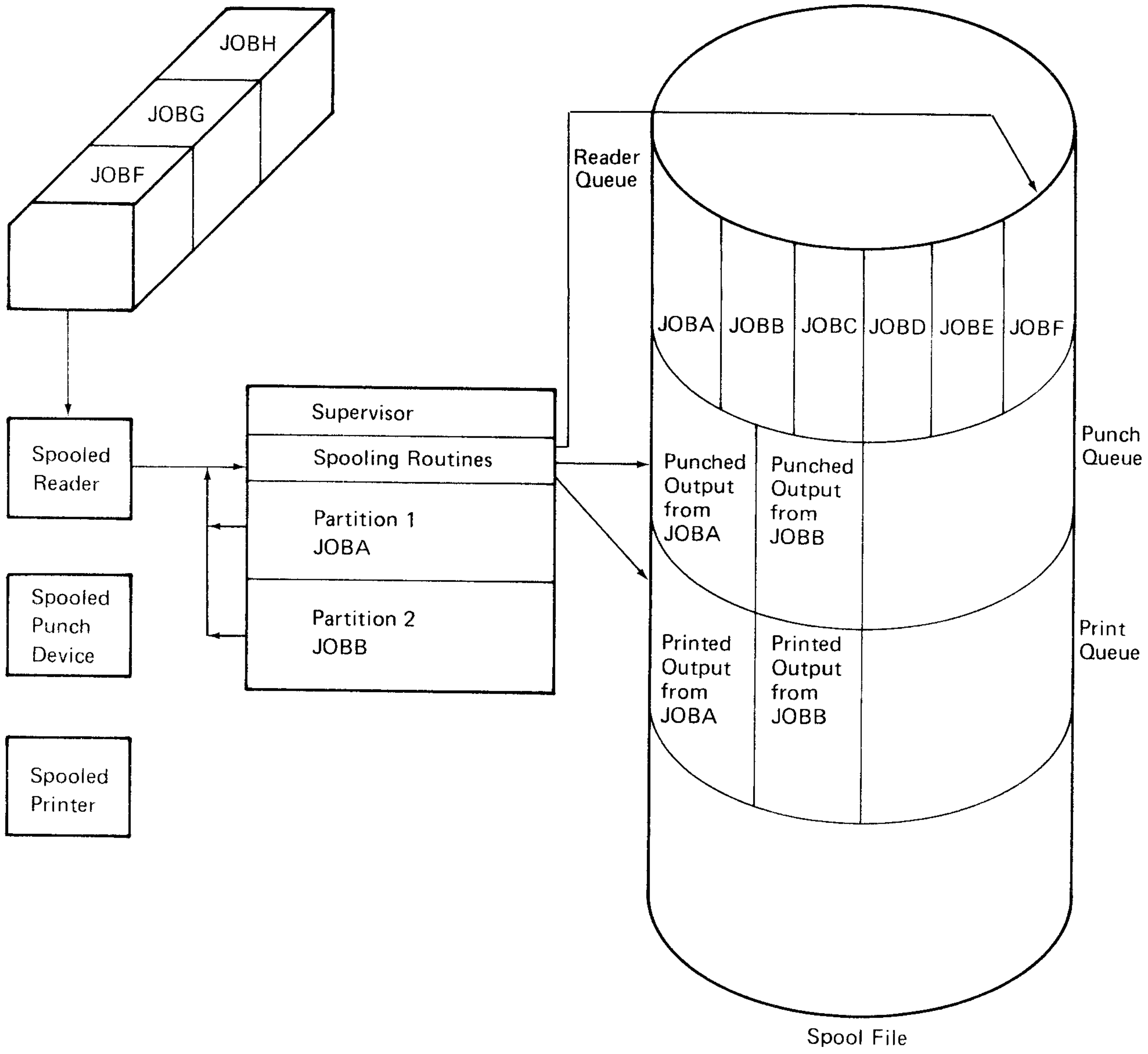


Figure 2-3. Building the Print Queue and Punch Queue

Device Contention

When a system has spooling in only one partition, some device contention may occur. Consider the following example:

	Partition 1	Partition 2
Jobname	JOBA	JOBB
Input device	MFCM1 (spooled)	2501 (not spooled)
Punch device	MFCM2 (spooled)	MFCM2 (not spooled)
Printer	1403 (spooled)	1403 (not spooled)

JOBA uses all three partition 1 devices, and JOBB uses all three partition 2 devices.

If JOBB needs to punch cards on the MFCM2 while input is still being spooled for JOBA on the MFCM1, contention occurs, and partition 2 issues the message MFCM NOT AVAILABLE. JOBB cannot use MFCM2 until the spooled reader has been terminated. At that time, the operator can respond to the partition 2 message initiating punching for JOBB.

If JOBA input is spooled to disk before JOBB requests the services of MFCM2, the above contention problem does not occur.

A similar contention problem can occur if the print or punch writer is started for partition 1 and JOBB needs to punch cards or use the printer. If one of these output writers has control, it must be terminated before JOBB can use the device. If JOBB has control, the output writer cannot be started until partition 2 has reached the end of job step.

With the preceding system setup, it is recommended that partition 1 be used for jobs that are mostly limited by unit-record I/O operations (reading cards, printing, and punching) and partition 2 be used for jobs that are limited by I/O operations other than unit-record (utilities, sorts, disk and tape activity). If both partitions had spool support in the preceding example, contention may not have occurred.

Considerations

The following considerations apply to job execution in a spooling environment:

- An execution-time diagnostic is issued if you attempt to spool a combined file. A combined file is a card file in which both input and output operations are performed (such as reading and punching the same card).
- Output operations (such as stacker selection) cannot be performed on input files. If your program specifies stacker selection, it is not necessary to recompile it. Spooling simply ignores the stacker select commands. Thus, if program input is not spooled, stacker selection occurs; if program input is spooled, stacker selection does not occur. If your application requires that input cards be separated, then the program can be rewritten to punch exceptions, or it can be executed nonspooled.
- The read card image operation is not supported.

Operator Control

The following OCC commands can be used during job execution; they are described because they have spooling implications. These commands are CANCEL, DUMP, and DISPLAY (see Chapter 5).

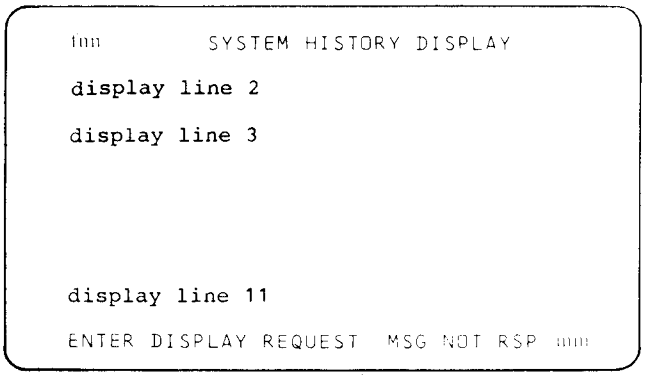
CANCEL: The operands P1, P2, or P3 specify which partition is to be canceled. A job currently being executed from the reader queue cannot be canceled by jobname; it can be canceled only by specifying the partition in which it is executing.

When a job is canceled by specifying the partition, its spooled output is not lost. The output is produced by the print and/or punch writer up to the last record prior to cancellation. The output is canceled only by entering a CANCEL command for that queue (such as CANCEL PRTQ,jobname).

DUMP: If the system is spooling the printed output for the partition being dumped, the dump output also is spooled. This dump terminates with the normal end-of-step message; processing of the next job for that partition can be resumed by responding to the end-of-job message.

DISPLAY: It is often helpful for the operator to display the system history area to determine the past or current activity of the partition (DISPLAY HISTORY).

The HISTORY operand causes the contents of the system history area to be displayed on the CRT similar to the following example. For 5704-SC2, history can also be displayed by individual tasks.



```
fnn          SYSTEM HISTORY DISPLAY

display line 2

display line 3




display line 11

ENTER DISPLAY REQUEST  MSG NOT RSP nnnn
```

The top line controls the direction of the display.

$$f = \left\{\begin{matrix} F \\ B \\ C \end{matrix}\right\} \begin{matrix} \text{Forward} \\ \text{Backward} \\ \text{Cancel the display} \end{matrix}$$

nn applies to the F and B functions and specifies how many lines to roll each time the ENTER key is pressed. For example, F10 means to roll the display forward 10 lines. If nn is not entered, 10 is assumed.

PRINTING

The print writer is the portion of spooling support that prints records on the 1403 Printer from the print output queue that was built during job execution (see Figure 2-4).

Once the print writer has begun printing output from the print queue, it continues until one of the following occurs:

1. All steps on the queue have been printed, or all steps on the queue of the specified forms type have been printed.

2. The operator enters a STOP PRT command to terminate printing.

3. A cancel option is selected for a printer diagnostic message.

4. The print queue is held (HOLD command) and at a later time is released (RELEASE command).

5. Each job step on the print queue is held (HOLD command).

If autowrite was specified during system generation, operator intervention is not required to resume printing for 1, 4, and 5. The print writer resumes printing automatically when output is available on the print queue.

If autowrite was not specified during system generation, the operator responds to the message PRTQ IS EMPTY/HELD to resume printing for 1, 4, and 5.

For 2 and 3, operator intervention is always required to resume printing; this is done by entering a START PRT or RESTART PRT command.

Output on the print queue is identified by jobname and stepname. Jobs, job steps, or the entire queue can be held, canceled, or displayed. If a job step is not in the keep state, it is removed from the print queue after all copies for that step have been printed.

The priority of a step on the print queue is the same as that for the executed job, unless a different priority is specified with a PRINTER OCL statement. For example, if a job has a priority of 3, then the priority of each step of that job on the print queue is also 3; but if a step has a priority of 1 assigned via a PRINTER OCL statement, all but that step will have a priority of 3. By changing the priority of a step on the queue, the operator can print a step's output in a sequence different than the sequence in which it was placed on the queue (CHANGE command).

If autostart was specified during system generation, the print writer begins when spooling is activated at IPL. If print records are not available and autowrite was specified during system generation, the print writer waits to begin printing output until output becomes available on the print queue. If print records are not available and autowrite was not specified during system generation, the message PRTQ IS EMPTY/HELD is displayed.

If autostart was not specified, the print writer must be started by the operator using the START PRT command. Printed output is taken off the print queue on a step basis allowing the operator to print a step's output out of the normal sequence.

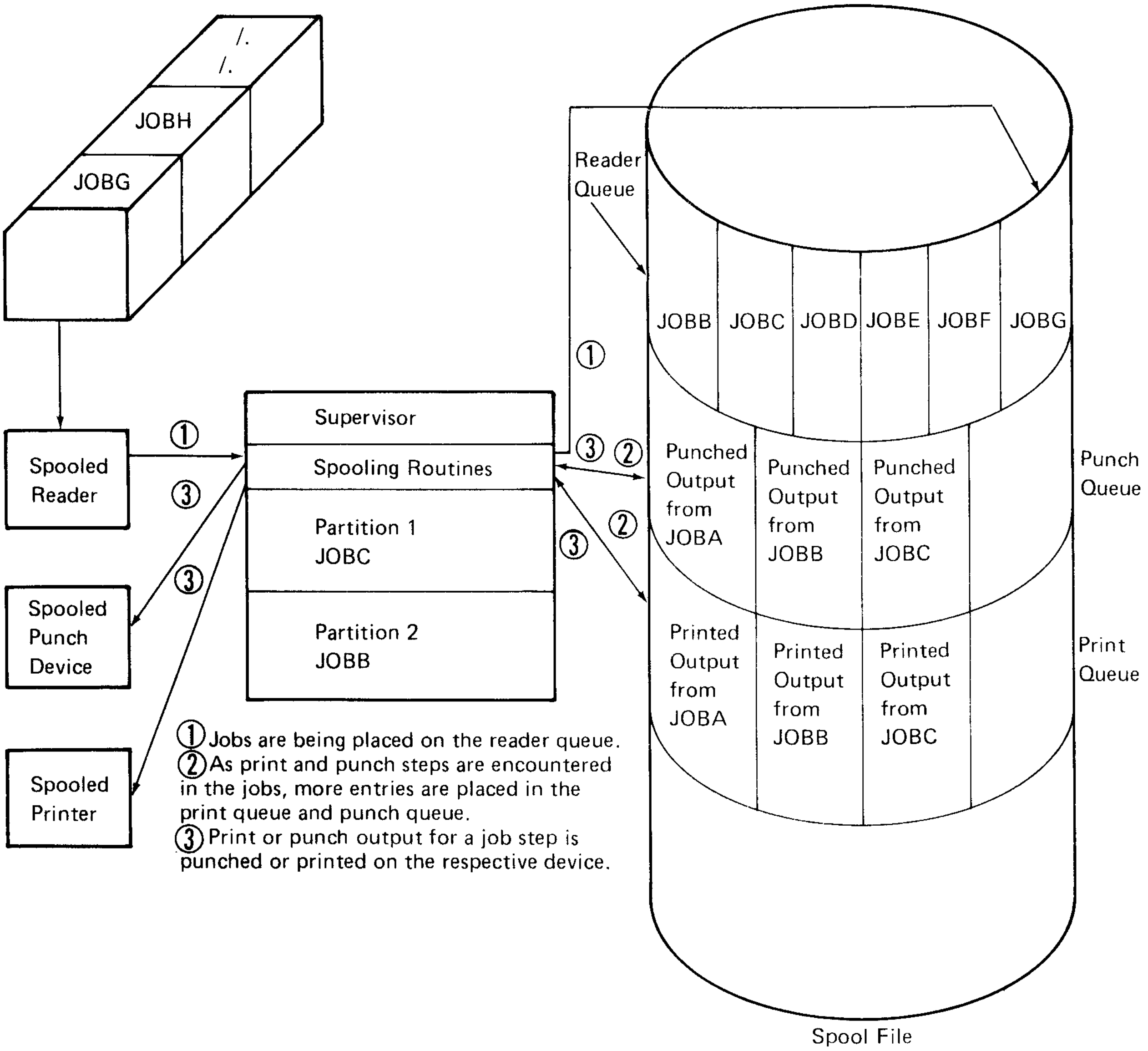
If time recording was specified during system generation, a message is written in the system history area each time the print writer completes printing a job step's output. The message will contain the date and time that the first record was printed, and the time the last record was printed.

More than one time recording message can be written in the system history area for each print step.

When a HOLD, RESTART, or STOP command is entered for an active print step, a time recording message is written in the system history area. If more than one of these commands is entered in succession, a time recording message is written only for the first command. A message is also written if a cancel option is taken to a printer error message.

If multiple copies are printed, a separate message is written for each copy.

When a retry option is taken to the ALIGN FORM TO LINE 1 message, the time recording message is not written even though the job step's printed output is again started from the beginning.



① Jobs are being placed on the reader queue.
② As print and punch steps are encountered in the jobs, more entries are placed in the print queue and punch queue.
③ Print or punch output for a job step is punched or printed on the respective device.

Figure 2-4. Starting the Print and Punch Devices

DEFER Option

Although the print writer normally prints output produced by job steps that have completed execution, it has the capability of printing a step's spooled print records during the execution of the step. The DEFER parameter can be specified in the PRINTER OCL statement to indicate that printing of the spooled output can begin while the step is still executing. When DEFER-NO is specified, the print writer begins printing the step's output if it is next on the print queue. If the DEFER parameter is not given, the system defaults to DEFER-YES, and the step's output is not printed until the step has completed execution.

DEFER-YES allows the operator to be sure that a job step has been successfully executed or terminated before printed output begins. If a problem occurs during execution, the printed output can be canceled without wasting time and paper printing the report.

A throughput improvement can be realized when using DEFER-NO for long-printing job steps. For example, a job step that executes 1 hour and prints 1 hour requires 2 hours with DEFER-YES – 1 hour of execution followed by 1 hour of spooled printing. With DEFER-NO, the print writer can be started shortly after execution starts; now, the time required for execution and printing is slightly more than 1 hour.

Spool File Security

For 5704-SC2, the QCOPY parameter of the PRINTER OCL statement allows print steps to be protected from access by the spool file copy program. If QCOPY-NO is specified, that print step cannot be copied by the spool file copy program; if QCOPY-YES is specified, the copy is allowed.

Grouping Job Steps

For 5704-SC2, the CLOSE parameter of the PRINTER OCL statement allows you to group several of a job's print steps as one step with one stepname. If CLOSE-NO is specified, no further print steps of that job are closed at end of step; but they are closed at end of job. If CLOSE-YES is specified, a print step is closed at end of step.

When a print step is not closed at end of step, a command to cause a skip to a new page is placed in the spool file.

If a previous print step specified CLOSE-NO, a PRINTER OCL statement with an ALIGN, COPIES, DEFER, FORMSNO, PRIORITY, or QCOPY parameter is ignored and causes an information message.

Restarting the Print Writer

The RESTART PRT command causes the print writer to stop printing output from the current job step, then restart from the beginning of any page of that step. A STOP PRT command is not required prior to issuing a RESTART PRT command; however, a RESTART PRT command can be issued following a STOP PRT command.

Prior to restarting the current job step's printed output, the print writer issues a message (PRT RESTART PENDING), giving the operator an opportunity to correct the condition that caused the RESTART command to be entered. Response to this message restarts the printing.

Changing Forms

Spooling support for the printer also has facilities available to aid the operator in changing forms. The forms required by a job step may be specified as the FORMSNO parameter on the PRINTER OCL statement when the step is submitted for execution. When the FORMSNO parameter is not given, the system defaults to the forms type specified during system generation. Thus, each job step on the print queue has a forms type associated with it.

The forms type can be changed for a job step through the use of the CHANGE forms type command (see Chapter 5).

On nonspooled systems, it is common practice to use a PAUSE statement in the job stream to remind the operator to change forms. Remember that on a spooled system, the PAUSE occurs at execution time, not when printing begins. Therefore, PAUSE statements used for this purpose should be changed to PRINTER statements.

The print writer can be started by keying the START PRT command with or without the optional forms type operand. However, when starting without the forms type operand, the print writer must determine if the forms required for this step differ from the forms used by the previously printed step. (Note that the previous printed step may or may not be in the same job.)

If a forms change is necessary, a message CHANGE FORM TO TYPE nnn is displayed on the CRT identifying the forms to be placed in the printer. The print writer does not continue until the operator has responded that the proper forms have been inserted in the printer.

The response to the forms change message also indicates if separator pages (Figure 2-5) are desired. A 0-option provides no separator pages; a 1-option provides three separator pages before the output of each job step and contains the jobname and stepname. Separator pages are produced for each step until another message (CHANGE FORM TO TYPE nnn) is displayed.

```
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX JOBNAME  STEPNAME  XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX JOBNAME  STEPNAME  XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX JOBNAME  STEPNAME  XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX JOBNAME  STEPNAME  XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX JOBNAME  STEPNAME  XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX JOBNAME  STEPNAME  XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

Figure 2-5. Separator Page Output

Starting the Print Writer Using a Forms Type

The print writer can be started by placing the forms type operand in the START command. For example, START PRT,forms type causes the print writer to print only those job steps in which the forms type matches that given in the START command. A message, CHANGE FORM TO TYPE nnn, is displayed on the CRT only for the first job step to print after the START command is given. If the printer has printed a job step with the same forms type as specified in the START PRT,forms type command, the CHANGE FORM TO TYPE nnn message is not displayed.

If autowrite has not been specified and the print writer has printed all job steps on the print queue with the specified forms type, the message PRTQ IS EMPTY/HELD TYPE nnn is written to the CRT. Additional steps placed on the print queue of the forms type nnn can be printed by responding to this message.

When autowrite has been specified, the message is not issued, and the print writer waits for additional jobs of forms type nnn to be placed on the print queue before it resumes printing.

To change the forms type with which the print writer is printing, the print writer must first be terminated (STOP PRT), then started using a new forms type with the START command. A job step being printed when the STOP PRT command is entered is retained on the print queue.

Forms Alignment

Two aids are available for aligning forms prior to and during a job step's printed output:

1. The ALIGN parameter can be specified on the PRINTER OCL statement when the step is submitted for execution. This allows forms alignment prior to printing a job step's output.

2. The RESTART PRT command can be used to restart a job step's printed output.

If ALIGN-YES was specified on the PRINTER OCL statement, the print writer prints the first line and displays a message, ALIGN FORM TO LINE 1, on the CRT. The operator, by responding to this message, may choose to repeat the alignment procedure or continue printing the step's output. (ALIGN-YES also applies to multiple copies and if output is restarted at the beginning.)

If the forms must be aligned during printing, the STOP PRT command may be entered to stop the print writer. At this time the forms can be aligned, after which the START PRT command is entered, causing the print writer to continue printing the output.

The RESTART PRT command can be entered to restart the step's printed output from the beginning. Prior to printing the step's output, a PRT RESTART PENDING message is issued. Response to this message restarts the printing.

If first page (1P) forms alignment is specified in an RPG II program and the printed output is being spooled, the RPG II forms alignment is bypassed during execution. Unless a PRINTER OCL statement with an ALIGN-YES parameter is included in the OCL for the execution of that program, the system does not halt to allow the operator to align the forms. The OCL option, used with spooling, operates in the same manner as the RPG II option.

Another way to allow the operator to adjust forms prior to spooled output is to use a different forms type for that step. This does not give the operator a chance to reprint the first line, but the halt allows checking of the forms alignment before beginning the printing.

Multiple Copies

Either the COPIES parameter on the PRINTER OCL statement, or the CHANGE copies command can cause more than one copy of printed output to be produced from the single execution of a job step. All requested copies for a job step are produced by the print writer before continuing to print the next job step on the queue. Although multiple copies are requested, only one copy of the output is in the print queue.

When a dump of main storage has been requested during the execution of a step by means of the DUMP command or by taking a D option to a message, the dump is included with the step's output in the print queue. In this case, only one copy of the output is printed, even though more copies were requested.

Changing the Print Chain/Train

No support is provided by spooling for changing the chain image for printed output. Print records that are spooled are printed using the currently loaded chain image. If a special print chain or train is required for a job on a system supporting spooling of printed output, the operator should use the following procedure (the example assumes that partition 1 is being used):

1. Enter a STOP SPOOL,P1 command. This causes the spooling support for the specified partition to terminate at the next end-of-job condition. If the partition is at end of job, spooling terminates immediately.

2. Enter a STOP PRT,PAGE command to terminate the printing of spooled print records at the end of a page and release the printer; or enter a STOP PRT,STEP command to allow the current print step to complete and then release the printer.

3. Stop the reader or punch writer if those devices are being used by spooling and are required by the special job.

4. Execute the job requiring the special print chain. It is suggested that the JOB statement for this job specify SPOOL-NO to ensure that the spooling functions in that partition have been terminated. Following the job's last step, the IMAGE statement must be given to restore the print chain image for the spool printed output. The IMAGE statement(s) should be followed by a /. statement.

5. At end of job, spooling for the partition can be restarted by entering the START SPOOL,P1 command. The desired spool functions (such as reader, print writer, punch writer) can then be started.

Error Conditions

When errors (such as a carriage sync check) occur on the printer while the spool print writer is active, the operator responds to the error as on a nonspooling system. When a 0- or 1-option is taken, the print writer continues printing from where it stopped. A 2- or 3-option causes the print writer to terminate; however, the job step remains on the print queue. (A CANCEL command removes the job step from the print queue.) It is important to note that one line of printing may be lost due to the type of error encountered. If this occurs, the operator can enter a RESTART PRT,PAGE command to start printing from the beginning of the page on which the error occurred.

Unprintable characters within the spooled printed output are handled in the manner requested by the user program. If the user program requests a message when unprintable characters are encountered, a message will be displayed if unprintable characters are found when the spooled output is printed. However, there are some exceptions:

1. A message will not occur if the step's printed output is preceded by OCL logged to the printer (LOG statement).

2. If the program requests a message in the first record of a step to be printed, an unprintable character in each succeeding record of the step will also cause a message even if one is not requested. If a message is not requested in the first record of a step to be printed, no message will be given, even if the program requests a message in a succeeding record.

3. A message will not occur if the step's printed output contains a dump and the printing of the step is started after end of step occurs. However, a message can occur if the step's printed output contains a dump and the printing of the step is started before end of step occurs.

Operator Control

Operator control commands are used to manipulate and monitor the print queue (see Chapter 5). These commands are CANCEL, HOLD, RELEASE, CHANGE, STOP, REUSE, DISPLAY, and KEEP.

For 5704-SC2, the CANCEL, HOLD, RELEASE, CHANGE, and KEEP commands can also be used to manipulate multiple jobs and/or job steps by specifying, in place of the jobname/stepname, one to seven characters followed by two asterisks (**). Spooling will then perform the requested function on all jobs/steps on the queue whose names begin with the specified characters. The following are some command examples and the results on the print queue if these commands are entered:

Command	Result
HOLD PRTQ,ABC**	All jobs whose jobnames begin with ABC are held.
HOLD PRTQ,ABC,12**	All job steps whose jobnames are ABC and whose stepnames begin 12 are held.
HOLD PRTQ,ABC***,12***	All job steps whose jobnames begin with ABC* and whose stepnames begin with 12* are held.

CANCEL: The operator can cancel a job on the print queue, a step on the queue, or the entire print queue. CANCEL *PRTQ does not affect the step currently printing or a step currently being placed on the print queue; however, a CANCEL PRTQ,jobname command or a CANCEL PRTQ,jobname,stepname command can be used to immediately terminate the current job or step printing.

HOLD: This command is used to prevent a job or step from printing. It can prevent the entire print queue from being printed or it can immediately interrupt the step currently being printed.

RELEASE: The RELEASE command allows the operator to release a job, job step, or the entire print queue from the hold state. The released job or job step retains the last priority assigned to it.

Jobs or job steps individually held are not released by a RELEASE PRTQ command; each job or job step must be released individually, by name (for example, RELEASE PRTQ,jobname).

If the entire queue is held, individual jobs or steps cannot be released. (The entire queue must be released.)

CHANGE: The CHANGE command allows the operator to alter the priority, forms type, or number of copies of a job or job step on the print queue. The CHANGE command does not affect a step currently being placed on the print queue. The uses of the CHANGE commands for the print queue are as follows:

- The CHANGE priority command is useful when the operator wants to obtain the output in a sequence different from which the output was put on the queue.
- The CHANGE forms type command is used to indicate the forms type required by the job step if the forms type required is different from, or was not specified on, the PRINTER OCL statement. The CHANGE forms type command does not affect a step currently printing.
- The CHANGE copies command is used to change the number of output copies to be produced by the job step.

STOP: By using the STOP PRT command, the operator causes the print writer to terminate and free the printer. The job step that was currently being printed is kept on the print queue and can be continued from the point of termination by using START PRT (or START PRT, forms type). The print writer can also be restarted from the beginning of any page by using the RESTART command.

The following example illustrates a valuable use for this command: Partition 1 has spooling support and the print writer is currently printing. Partition 2 does not have spooling support and the current job step has just requested use of the printer (the same printer is used by both partitions). The CRT displays the message 1403 NOT AVAILABLE. The operator enters the STOP PRT command and the message PRT WRITER TERMINATED is displayed. The operator responds to the 1403 NOT AVAILABLE message. The job step in partition 2 now executes using the printer. When this step reaches ES (end of step), the operator can resume printing from the print queue with either a START PRT or a RESTART PRT,PAGE command.

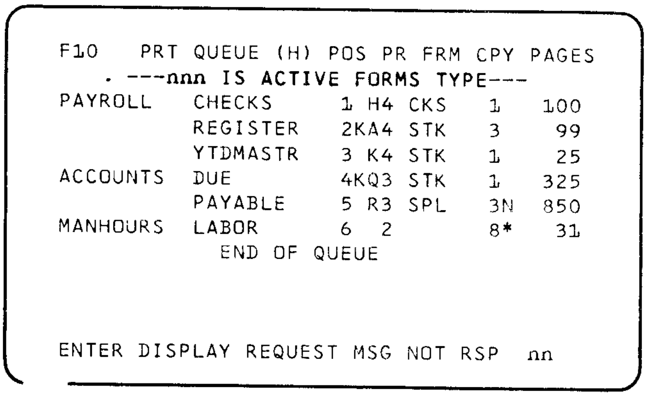
REUSE (5704-SC2 only): The REUSE command frees space in the spool file from the currently printing job step (before printing is complete), making that space available for other uses. A job step that is in the keep state cannot be reused.

DISPLAY: The operator determines the status of the print queue with the DISPLAY PRTQ command.

The following is an example of how a display of the print queue might appear: Information displayed for the print queue is jobname, stepname, position on queue (POS), priority (PR), forms number (FRM), number of copies (CPY), and number of pages (PAGES).

PAGES is a display of the total number of pages required to print the job step. The spool printer intercept routine maintains the page count for the job step as output is being placed onto the print queue. The page count is incremented by 1 when spooling detects a space or a skip to a new page. If a space or a skip to a new page is the last printer command issued by the job step (as in the case of logging the end-of-job statement to the printer which issues a skip after printer command), the page count appears to be incorrect by 1. This last movement to a new page is counted even though no further printing occurs. A page count up to 65,535 pages can be displayed. When 65,536 is reached, the page counter is reset to 0; thus, 0 indicates page 65,536 and 1 indicates 65,537.

If the print writer was started using a forms type, that forms type will be displayed in the message nnn IS ACTIVE FORMS TYPE.



```
F10    PRT QUEUE (H) POS PR FRM CPY PAGES
   . ---nnn IS ACTIVE FORMS TYPE---
PAYROLL   CHECKS      1 H4 CKS  1   100
          REGISTER    2KA4 STK  3    99
          YTDMASTR    3 K4 STK  1    25
ACCOUNTS  DUE         4KQ3 STK  1   325
          PAYABLE     5 R3 SPL  3N  850
MANHOURS  LABOR       6  2      8*   31
            END OF QUEUE



ENTER DISPLAY REQUEST MSG NOT RSP  nn
```

Line 1 controls the scrolling of the display.

If, after displaying the contents of one queue using the DISPLAY command, the operator wants to display the contents of another queue, the operand for the alternate queue is entered in the first two or four positions of line 1.

For example, to display the reader queue, RDRQ is entered in the first four positions, or RQ is entered in the first two positions. The ENTER key is pressed, and the reader queue is displayed.

An (H) on the first line of the display indicates that the entire print queue is held via the HOLD command.

One or two of the following characters can precede the priority number in the display:

A -- Indicates that the job step is currently being printed
H -- Indicates that the job or job step is being held via the HOLD command
K -- Indicates that the job or job step is in a keep state via the KEEP command (5704-SC2 only)
Q -- Indicates that the job or job step is being copied by $COPY (5704-SC2 only)
R -- Indicates that the job step was partially canceled with a REUSE command and is no longer being printed (5704-SC2 only)

An asterisk (*) after the number of copies indicates that the job step is executing and DEFER-NO was specified on the PRINTER OCL statement for the job step. A job step that is executing without DEFER-NO specified is not displayed. For 5704-SC2, an N after the number of copies indicates that the job step cannot be copied with the spool file copy program (QCOPY-NO specified on the PRINTER OCL statement).

KEEP (5704-SC2 only): The KEEP command is used to prevent a job step from leaving the print queue after a copy for that step has been printed. Rather, the job step remains on the queue in the hold state. If the job step has multiple copies, one copy is printed, the number of copies is reduced by one, and the job step remains on the queue in the hold state. A job step that has been reused with the REUSE command cannot be kept.

Displaying the Current Printed Page Count

To determine the number of the page currently being printed, the operator can enter a STOP PRT command. This causes a
PRT WRITER TERMINATED PAGE nnnnn
message, where nnnnn is the number of the current page.

If the operator enters a START PRT command following this message, the print writer continues from where it was stopped.

Printing the Last Pages First

To check the end of a long printed report before all of the report is printed, you should:

1. Use a CHANGE command to increase the number of copies by 1. The number of copies will be decreased by 1 after the last page is printed.

2. Start printing the first page to ensure that the report is the next one on the queue to be printed.

3. Enter a RESTART PRT,nnnnn command, where nnnnn is a page number close to the end of the report.

4. To stop the writer at the end of this copy, enter a STOP PRT,STEP command at the PRT RESTART PENDING message or while the pages are being printed. This allows you to inspect the end of the report before deciding whether to print the entire report at this time, print it later, or cancel it from the queue.

Skipping Back Several Pages

You may wish to skip back several pages after a forms jam or any other printer error. To do this, take a 2- or 3-option to the printer error message; this action causes a PRT WRITER TERMINATED PAGE nnnnn message, where nnnnn is the current page number. You can now subtract the number of pages to be skipped back and enter a RESTART PRT,nnnnn command, where nnnnn is the resulting page number.

PUNCHING

The punch writer is the portion of spooling support that punches cards from the punch output queue that was built during job execution.

When the punched output of a step or a copy of punched output is complete, a message, END OF PCH STEP, is displayed on the CRT, notifying the operator that the punched cards can be removed from the stacker(s). On the 2560 or 1442, cards are fed from the hopper causing cards in the feed path to be placed in the stacker(s). The operator does not need to initiate nonprocess runout (NPRO). The response to the message causes the next copy of the job step or the next job step on the punch queue to be started. Separator cards are not inserted between steps.

The punch writer continues to produce output in this manner until one of the following occurs:

1. All steps on the punch queue are punched.

2. A STOP PCH command is entered.

3. A cancel option is selected for a punch diagnostic message.

4. The entire queue is held (HOLD command) and at a later time is released (RELEASE command).

5. Each step is held (HOLD command).

If autowrite was specified during system generation, operator intervention is not required to resume punching for 1, 4, and 5. The punch writer resumes punching automatically when output is available on the punch queue.

If autowrite was not specified during system generation, the operator responds to the message PCHQ IS EMPTY/HELD to resume punching for 1, 4, and 5.

For 2 and 3, operator intervention is always required to resume punching; this is done by entering a START PCH or RESTART PCH command.

Output on the punch queue is identified by jobname and stepname. Jobs, job steps, or the entire queue can be held, canceled, or displayed. If a job step is not in the keep state, it is removed from the punch queue after all copies for that step have been punched.

The priority of a step on the punch queue is the same as the job that was executed, unless a different priority is specified with a PUNCH OCL statement. For example, if a job has a priority of 3, then the priority of its output on the punch queue is also 3; but if a step has a priority of 1 assigned via a PUNCH OCL statement, all but that step will have a priority of 3. By changing the priority of a step on the queue, an operator can punch a step's output in a sequence different from the sequence in which it was placed on the queue (CHANGE command).

If autostart was specified during system generation, the punch writer begins when spooling is activated at IPL. If punch records are not available and autowrite was specified during system generation, the punch writer waits to begin punching output until output becomes available on the punch queue. If punch records are not available and autowrite was not specified during system generation, the message PCHQ IS EMPTY/HELD is displayed.

If autostart was not specified, the punch writer must be started by the operator using a START PCH command. Punching of spooled data cannot begin until the reader has terminated.

If time recording was specified during system generation, a message is written in the system history area each time the punch writer completes punching a job step's output. The message will contain the date and time that the first record was punched, and the time the last record was punched.

More than one time recording message can be written in the system history area for each punch step.

When a HOLD, RESTART, or STOP command is entered for an active punch step, a time recording message is written in the system history area. If more than one of these commands is entered in succession, a time recording message is written only for the first command. A message is also written if a cancel option is taken to a punch error message.

When multiple copies are punched, a separate message is written for each copy.

DEFER Option

While the punch writer normally punches output produced by job steps that have completed execution, it can also punch a step's spooled punch records during the execution of the step. The DEFER parameter can be specified in the PUNCH OCL statement to indicate that punching of the spooled output can begin while the step is still executing. When DEFER-NO is specified, the punch writer begins punching the step's output if it is next on the punch queue. If the DEFER parameter is not given, the system defaults to DEFER-YES, and the step's output is not punched until the step has completed execution.

DEFER-YES allows the operator to be sure that a job step has been successfully executed or terminated before punched output begins. If a problem occurs during execution, the punched output can be canceled without wasting the time and cards needed to punch the output.

Use of DEFER-NO can improve throughput. For example, a job step that executes 1 hour and punches 1 hour requires 2 hours with DEFER-YES – 1 hour of execution followed by 1 hour of spooled punching. With DEFER-NO, the punch writer can be started shortly after execution starts. Now, the time required for execution and punching is slightly more than 1 hour.

Spool File Security

For 5704-SC2, the QCOPY parameter of the PUNCH OCL statement allows punch steps to be protected from access by the spool file copy program. If QCOPY-NO is specified, that punch step cannot be copied by the spool file copy program; if QCOPY-YES is specified, the copy is allowed.

Restarting the Punch Writer

The RESTART PCH command causes the punch writer to stop punching from the current job step and immediately restart from the beginning of that step. A STOP PCH command is not required prior to entering a RESTART PCH command; however, a RESTART PCH command can be entered following a STOP PCH command.

Prior to restarting, the punch writer issues a message (PCH RESTART PENDING) to give the operator an opportunity to correct the condition that caused the RESTART command to be entered.

Reading and Punching

The punch writer and input reader cannot operate simultaneously. When a job stream is being read to the reader queue, the punch writer cannot punch from the punch queue. Likewise, when the punch writer is operating, the reader cannot put jobs on the reader queue. OCC commands are available for stopping and starting the punching and reading functions.

Changing Card Types

Spooling support for the punch has facilities that inform the operator to use a different card type for a particular step. The card type required by a job step can be specified in the CARDNO parameter on the PUNCH OCL statement when the step is submitted for execution or with the CHANGE card type command after the job step has been placed on the punch queue. When the CARDNO parameter is not given, or the card type has not been changed by the CHANGE command, the system defaults to the card type specified during system generation. Thus, each job step on the punch queue has a card type associated with it.

When the punch writer prepares to process the output from a job step, it determines whether the card type required for the current step agrees with the card type used by the previously punched step. (Note that the previously punched step might not be from the same job.)

If the card types are not the same, the punch writer issues a message, CHANGE CARD TO TYPE nnn, informing the operator that a different card type is required. The punch writer does not continue until the operator has responded to the message.

Multiple Copies

The COPIES parameter on the PUNCH OCL statement can cause multiple copies of the punched output to be produced from the single execution of a job step. After the job step has been placed on the punch queue, the CHANGE copies command can be used to change the number of copies of output produced by a job step. Although multiple copies are requested, only one copy of the output is in the punch queue.

After punching one copy of a job step's punched output, the punch writer issues a message (END OF PCH STEP) allowing the punched cards to be removed from the stacker(s). Response to the message causes the punch writer to produce the next copy and continue in this manner until all copies have been punched.

When a dump of main storage has been requested during the execution of a step by means of the DUMP command or by taking a D-option to a message only one copy of the output is punched, even though more copies were requested.

Error Conditions

When errors (such as a feed check) occur on the punch device and the punch writer is active, the operator responds to the error in the same way as on a nonspooling system. When a 0- or 1-option is taken, the punch writer continues punching from where it stopped. A 2- or 3-option causes the punch writer to terminate; however, the job step remains on the punch queue. (A CANCEL command removes the job step from the punch queue.) Depending on the error condition and recovery procedure, one punched card may be invalid or lost. If this occurs, the operator can enter a RESTART PCH command to start punching from the beginning of the job step.

Operator Control of Punch Spooling

Operator control commands are used to manipulate and monitor the punch queue (see Chapter 5). These commands are CANCEL, HOLD, RELEASE, CHANGE, STOP, REUSE, DISPLAY, and KEEP.

The CANCEL, HOLD, RELEASE, CHANGE, and KEEP commands can also be used to manipulate multiple jobs and/or job steps by specifying, in place of the jobname/stepname, one to seven characters followed by two asterisks (**). Spooling will then perform the requested function on all jobs/steps on the queue whose names begin with the specified characters. The following are some command examples and the results on the punch queue if these commands are entered:

Command	Result
HOLD PCHQ,ABC**	All jobs whose jobnames begin with ABC are held.
HOLD PCHQ,ABC,12**	All job steps whose jobnames are ABC and whose stepnames begin with 12 are held.
HOLD PCHQ,ABC***,12***	All job steps whose jobnames begin with ABC* and whose stepnames begin with 12* are held.

CANCEL: The operator can cancel a job on the punch queue, a step on the queue, or the entire punch queue. CANCEL *PCHQ does not affect the step currently being punched or a step currently being placed on the punch queue; however, a CANCEL PCHQ,jobname command or a CANCEL PCHQ,jobname,stepname command can be used to immediately terminate the current job or step punching.

HOLD: The HOLD command can prevent all jobs and steps on the punch queue from being punched. A single step, a complete job, or the entire punch queue can be held. If a job or step is held on the punch queue when that job or step is being punched, it is put back on the queue and the next step is punched.

RELEASE: The RELEASE command provides the means to free a job, a step, or the entire punch queue after it was put in the hold state. Jobs or steps that were individually held must be released individually; they cannot be released with one RELEASE PCHQ command. If the entire queue was held, individual jobs or steps cannot be released; the entire queue must be released.

CHANGE: The CHANGE commands allow the operator to alter the priority, card type, or number of copies of a job or job step on the punch queue. The CHANGE command does not affect a step currently being placed on the punch queue.

The uses of the CHANGE commands for the punch queue are as follows:

- The CHANGE priority command is useful when the operator wants punched output in a sequence different from the sequence on the queue.
- The CHANGE card type command is used to indicate the card type required by the job step if the card type required is different from, or was rot specified, on the PUNCH OCL statement. The CHANGE card type command does not affect a step currently punching.
- The CHANGE copies command is used to change the number of output copies to be produced by the job step.

Consider this example, which illustrates how the CHANGE priority command might be used:

Partition 1 is supported by punch spooling and the punch queue has two steps: STEP1 requires 30 minutes of punching and STEP2 requires 5 minutes of punching.

Partition 2 is not supported by spooling and is currently executing JOBA which requires another 10 minutes to be completed. When JOBB starts executng, it will require the data contained in STEP2.

The operator can use the CHANGE priority command to give STEP2 a higher priority than STEP1. STEP2 can now be punched and the output be made available for JOBB.

STOP: The STOP PCH command allows the operator to terminate the punch writer and free the spooled punch device. The step that was currently being punched is kept on the queue and can be continued from the point of termination with the START PCH command or restarted from the beginning of the step with the RESTART PCH command. When STOP PCH,STEP is entered, the punch writer terminates after the step or copy has completed punching.

The STOP command can minimize device contention with both a nonspooled partition and a spooled reader device. Consider the following two examples:

1. The punch writer is currently punching cards on the 1442 for partition 1. The program in partition 2 is not spooled and has requested the use of the 1442.

   A message is displayed on the CRT indicating that the 1442 is not available for partition 2. The operator enters a STOP PCH,STEP command to stop the punch writer when the punching is completed for the current step or copy. When completed, the message PCH WRITER TERMINATED is displayed. The operator then responds to the partition 2 message, and the job step in partition 2 can now use the 1442. At the completion of this step, the operator enters a START PCH command to cause punching to resume from the punch queue.

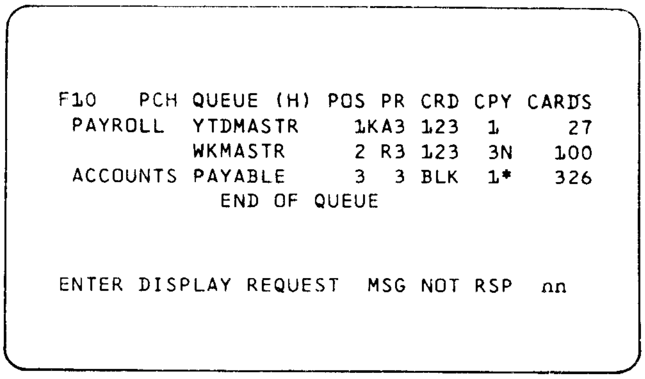
2. The system has input and punch spooling, with the 2560 specified as both the reader and punch device. (Partition 2 is being used for an online inquiry application.) The punch writer is currently punching and the operator has just received a top priority job to be executed immediately.

   The STOP PCH command terminates punching immediately. The rush job is placed in the 2560, and a START RDR command is entered. After the job has been spooled to the reader queue, the message SPOOL RDR TERMINATED is displayed, and the operator responds to the ES or EJ message to start job execution. A START PCH command can be entered to resume punching from the punch queue.

REUSE (5704-SC2 only): The REUSE command frees space in the spool file from the currently punching job step (before punching is complete), making that space available for other uses. A job step that is in the keep state cannot be reused.

DISPLAY: The DISPLAY command is useful in displaying the current status of the punch queue. In the preceding examples, the display could have been used to find out what steps were on the punch queue and what their priorities were. The operator could also have used the display function after manipulating the queue to ensure that everything was correct.

The following is an example of the punch queue display. Information displayed for the punch queue is jobname, stepname, position on queue (POS), priority (PR), card number (CRD), number of copies (CPY), and number of cards (CARDS).



```
F10    PCH QUEUE (H) POS PR CRD CPY CARDS
 PAYROLL  YTDMASTR     1KA3 123  1     27
          WKMASTR      2 R3 123  3N   100
 ACCOUNTS PAYABLE      3  3 BLK  1*   326
            END OF QUEUE


ENTER DISPLAY REQUEST  MSG NOT RSP  nn
```

An (H) on the first line of the display indicates that the entire punch queue is held via the HOLD command.

One or two of the following characters can precede the priority number in the display:

A – Indicates that the job step is currently being punched
H – Indicates that the job or job step is being held via the HOLD command
K – Indicates that the job or job step is in a keep state via the KEEP command (5704-SC2 only)
Q – Indicates that the job or job step is being copied by $QCOPY (5704-SC2 only)
R – Indicates that the job step was partially canceled with a REUSE command and is no longer being punched (5704-SC2 only)

An asterisk (*) after the number of copies indicates that the job step is executing and DEFER-NO was specified on the PUNCH OCL statement for the job step. A job step that is executing without DEFER-NO specified is not displayed. For 5704-SC2, an N after the number of copies indicates that the job step cannot be copied by the spool file copy program (QCOPY-NO specified on the PUNCH OCL statement).

KEEP (5704-SC2 only): The KEEP command is used to prevent a job step from leaving the punch queue after a copy for that step has been punched. Rather, the job step remains on the queue in the hold state. If the job step has multiple copies, one copy is punched, the number of copies is reduced by one, and the job step remains on the queue in the hold state. A job step that has been reused with the REUSE command cannot be kept.

Chapter 3. Processing without Spooling

There may be situations when the operator or system programmer chooses to run without spool support even though it is included in the system. The following are some reasons for not using spool support:

- A job stream has very little card input or card or printer output.
- Jobs require combined files (specifying print or punch operations on the input card files).
- The input card files use stacker selection.
- The printer chain image must be changed frequently.
- Card files are merged.

There are two ways to process a job without spooling:

- At IPL, the operator can enter a CANCEL SPOOL command; this command is in effect until another IPL.
- The operator can enter a STOP SPOOL command; this command is in effect until a START SPOOL command is issued.

The JOB OCL statement can contain the SPOOL-NO parameter. Use of this parameter only ensures that the job is not executed when spooling is active. (SPOOL-NO does not affect spooling.)

CHECKPOINT/RESTART

Checkpoint/restart is an optional facility of the system control program (SCP) that can be included in your system during system generation. Used with an object program, such as COBOL, the checkpoint function records the status of the program at specified intervals. If processing is terminated before normal end of job, the program can be restarted from the last checkpoint rather than from the beginning.

The following conditions are necessary to allow checkpoints to be taken. All of these conditions are necessary for taking a checkpoint; if any of these conditions do not exist, the checkpoint program can be executed, but checkpoints are ignored:

- The checkpoint program must be executed in partition 1.
- Spooling must not be active in partition 1.
- Unit-record devices required by the checkpoint program cannot be used by the spool reader or writers.

If spooling is active, the job can be prevented from running by specifying SPOOL-NO on the JOB statement. If SPOOL-YES is specified (or if SPOOL-NO is not specified), and if the checkpoint program is allowed to run with spooling active in partition 1, then checkpoints are ignored.

Prior to executing a checkpoint program, the operator can stop spooling in the partition by using the STOP SPOOL,P1 command.

The JOB statement should also specify PARTITION-1. This prevents the checkpoint program from being allowed to run in another partition where checkpoint requests are ignored.

The following example suggests an operating procedure that can be used when executing a checkpoint program:

Assume that the 2501 is the spooled reader, the 1442 is the spooled punch, and the 1403 is the spooled printer; also, assume that the checkpoint program reads cards from the 1442 and prints on the 1403.

The OCL could be the following:

```
// JOB01 JOB SPOOL-NO,PARTITION-1
// PAUSE ENTER . . . STOP,PRT . . . STOP,PCH
// LOAD CHKPT,F1
// RUN
      .
      .
      .
```

The JOB statement ensures that the program is executed only in partition 1 with spooling stopped in that partition.

The PAUSE statement reminds the operator to stop the spooled punch and printer if they are active. When this is done, the operator responds to the PAUSE message, and the checkpoint program can be executed.

To reactivate spooling after the checkpoint program is finished, the operator can use the following commands:

```
START SPOOL,P1
START PRT
START PCH
```

When restarting from a checkpoint, the required devices must be available and spooling must be stopped in partition 1. The OCL could be the following:

```
// JOB02 JOB PARTITION-1,SPOOL-NO
// PAUSE ENTER . . . STOP,PRT . . . STOP,PCH
// LOAD $$RSTR,F1
// RUN
      .
      .
      .
```

If spooling is active in partition 1 when restarting a program, a diagnostic message is issued to force the restart request to be canceled. The operator can enter STOP SPOOL,P1 and reenter the restart job.

ROLLOUT/ROLLIN

Rollout/rollin is an optional facility of the system control program (5704-SC1 only) and can be included in your system during system generation. Used with an object program, such as RPG II or FORTRAN, the rollout/rollin function provides a means of interrupting (rolling out) one program to allow another program to be executed. When the interrupting program is finished, the interrupted one is reinstated (rolled in) to continue executing.

A rollout is requested by means of the operator control command ROLLOUT, or simply RO. A rollout can be requested even while spooling is active; however, any unit-record devices being used by spooling must be released via OCC and made available to the interrupting program.

Note: Rollout/rollin can be used in partition 1 only.

Consider an example in which the 1442 is the spooled punch device and the 1403 is the spooled print device. In this example, the interrupting program reads cards from the 1442 and prints on the 1403:

1. The user enters the ROLLOUT command. A message (CC JY RD) is displayed, giving the user the opportunity to either cancel the rollout request or continue with the interruption. A 0-option allows the rollout request to continue.

2. The user enters commands to stop the print and punch writers (STOP PRT and STOP PCH).

   As a result, the 1403 and 1442 can be used by the interrupting program.

3. When the system displays ENTER READER DATA P1, the user can enter OCL statements for the interrupting program. Since the interrupting program reads cards from the 1442, it may be necessary to change the system input device by entering a READER statement.

4. The interrupting program is executed. Because rollin is pending, output from the interrupting program cannot be spooled.

5. At the completion of the interrupting program, a message (CC J' RD) is displayed, informing the user that the interrupted program is about to be resumed. After preparing the 1442 and 1403, the user responds to the message with a 0-option.

6. The user enters commands to restart the print and punch writers (START PRT and START PCH).

7. The execution of the interrupted program resumes.

Chapter 5. Operator Control Commands for Spooling

Operator control commands (OCC) are available for monitoring, controlling, and communicating with the spooling function. The operator enters OCC using the CRT/keyboard. The following commands can be entered in a spooling environment:

CANCEL

CHANGE

DISPLAY

HIPTY

HOLD

KEEP

PTY

READER

RELEASE

RESTART

REUSE

START

STOP

The following conventions are used to illustrate the commands:

1. Uppercase letters and punctuation marks (except brackets, braces, and parentheses) must be entered exactly as shown.

2. Lowercase letters and terms represent variable information that you supply depending on the operation you want to perform.

3. One or more items within brackets [ ] indicate an optional choice. You can include one item or omit the entire operand as required by the operation you are performing.

4. Items in braces { } indicate a required choice. You must enter one of the items.

5. Optional items that are underlined indicate the default value. If none of the items is entered, the default value is used.

6. Abbreviations, if applicable, are enclosed in parentheses () below the operation or operand. These abbreviations can be entered in place of the keywords.

7. Parameters in the commands are positional and must be entered in the order shown in this chapter. If a positional parameter is left out between other parameters, the comma must not be omitted.

In the following OCC descriptions, when a jobname or a jobname and stepname are used, the requested action is performed on all jobs or steps on the queue that use the name you enter.

Note: If the jobname or stepname in the OCL contains characters which are not on the 3277 keyboard, that job or job step cannot be referenced by name when using OCC.

CANCEL

This command takes effect immediately.

Functions

- Cancels spooling.
- Cancels the reader queue or one or more jobs on the reader queue.
- Cancels the print queue, one or more jobs on the print queue, or one or more jobs steps on the print queue.
- Cancels the punch queue, one or more jobs on the punch queue, or one or more job steps on the punch queue.

  Note: The preceding functions are valid only for systems that support spooling.

- Cancels a program that is executing in a partition.

Format

```
Operation   Operand

CANCEL    ⎧ SPOOL                                            ⎫
(CN)      ⎪ (SP)                                             ⎪
          ⎪                                                  ⎪
          ⎪ RDRQ, {jobname      }                            ⎪
          ⎪ (RQ)  {characters** }                            ⎪
          ⎪                                                  ⎪
          ⎪ PRTQ, {jobname      }  [,stepname     ]          ⎪
          ⎪ (WQ)  {characters** }  [,characters** ]          ⎪
          ⎪                                                  ⎪
          ⎪ PCHQ, {jobname      }  [,stepname     ]          ⎪
          ⎪ (PQ)  {characters** }  [,characters** ]          ⎪
          ⎪                                                  ⎪
          ⎪ *RDRQ                                            ⎪
          ⎨ (*RQ)                                            ⎬
          ⎪                                                  ⎪
          ⎪ *PRTQ                                            ⎪
          ⎪ (*WQ)                                            ⎪
          ⎪                                                  ⎪
          ⎪ *PCHQ                                            ⎪
          ⎪ (*PQ)                                            ⎪
          ⎪                                                  ⎪
          ⎪ P1[,2]                                           ⎪
          ⎪   [,3]                                           ⎪
          ⎪                                                  ⎪
          ⎪ P2[,2]                                           ⎪
          ⎪   [,3]                                           ⎪
          ⎪                                                  ⎪
          ⎪ P3[,2]                                           ⎪
          ⎩   [,3]                                           ⎭
```

Contents

SPOOL (SP): This operand can be specified at IPL prior to spooling being activated. The main storage used by the spooling routine is made available to any partition. (For a partition to use this additional main storage a SET command must be entered.) IPL must be performed again to activate spooling. Once spooling is activated, it cannot be canceled.

RDRQ (RQ): This operand specifies the queue that contains the job to which the cancel applies. The jobname operand specifies the job that is canceled from the reader queue. The RDRQ operand is valid only on systems that support input spooling.

PRTQ (WQ): This operand specifies the queue that contains the job or job step to which the cancel applies. When the jobname operand is used, all steps of the specified job are canceled from the print queue. When both the jobname and stepname are used, only the specified step is canceled. If the job or step you want to cancel is currently being printed, it is terminated immediately; and the next step on the queue is processed.

PCHQ (PQ): This operand specifies the queue that contains the job or job step to which the cancel applies. When the jobname operand is used, all steps of the specified job are canceled from the punch queue. When both the jobname and stepname are used, only the specified step is canceled. If the job or step you want to cancel is currently being punched, it is terminated immediately; and the next step on the queue is processed. The PCHQ operand is valid only on systems that support punch spooling.

**RDRQ (*RQ):* This operand cancels the entire reader queue but does not affect the job currently executing or the job currently being placed on the reader queue. The *RDRQ operand is valid only on systems that support input spooling.

**PRTQ (*WQ):* This operand cancels the entire print queue but does not affect the job step currently being printed or the output being placed on the queue by an executing program.

**PCHQ (*PQ):* This operand cancels the entire punch queue but does not affect the job step currently being punched or the output being placed on the queue by an executing program. The *PCHQ operand is valid only on systems that support punch spooling.

P1, P2, or P3: This operand cancels the program currently executing in the specified partition (1, 2, or 3). The optional 2 operand causes a controlled cancel to be taken; in this case files in the program are closed. The optional 3 operand causes an immediate cancel. If neither a 2 or 3 is specified, 2 is assumed. The P1, P2, or P3 operand can be used whether or not spooling is supported by the system. P3 is supported by 5704-SC2 only.

jobname: This operand specifies the job to which the cancel applies.

stepname: This operand specifies the job step to which the cancel applies.

*characters** (5704-SC2 only):* This operand specifies the jobs and/or job steps to which the cancel applies. Multiple jobs/steps can be referenced by specifying one to seven characters followed by two asterisks (**). All jobs/steps on the queue whose names begin with the specified characters will be canceled.

Considerations

A job executing in a partition cannot be canceled by specifying its name. It can be canceled only by specifying the partition in which it is executing.

Consider the following example:

```
CANCEL PRTQ,JOBA,STEP1
or
CN WQ,JOBA,STEP1
```

STEP1 of JOBA on the print queue is removed from the queue.

CHANGE CARD TYPE

Function

Changes the card type characters associated with one or more jobs or job steps on the punch queue. This command is valid only if spooling is active.

Format

```
Operation  Operand

CHANGE     CRD,card type, [ PCHQ ] , { jobname       } [ ,stepname     ]
(G)                       [ (PQ) ]   { characters**  } [ ,characters** ]
```

Contents

CRD: This operand indicates a change in the card type characters for a job or job step on the punch queue.

card type: This operand may be any combination of one to three characters (except commas or blanks) to indicate the card type used. These characters are examined by the punch writer when punching spooled output. If the card type differs from that of the previous punch step, the system issues a message to the operator indicating the card type to be used.

PCHQ: This operand specifies the queue that contains the job to which the change in the card type applies. Since only the punch queue is valid for this function, the queue operand can be omitted as shown in the following example:

```
CHANGE CRD,ABC,,JOB1,STEP1
```

jobname: This operand specifies the job to which the change of card type applies.

stepname: This operand specifies the job step to which the change of card type applies.

*characters** (5704-SC2 only):* This operand specifies the jobs and/or job steps to which the change of card type applies. Multiple jobs/steps can be referenced by specifying one to seven characters followed by two asterisks (**). All jobs/steps on the queue whose names begin with the characters specified will have the card type changed.

Considerations

Unless the CHANGE card type command specifies stepname, all job steps within a job on the punch queue receive the change.

The CHANGE card type command does not affect a punch step that is currently punching.

CHANGE COPIES

Function

Changes the number of output copies produced by one or more jobs on an output queue. This command is valid only if spooling is active.

Format

Operation   Operand

```
                          ( PRTQ )
                          ) (WQ) (    ( jobname       )   [,stepname      ]
CHANGE   CPY, [number] ,  < PCHQ >  , < characters**  >   [,characters**  ]
(G)                       ( (PQ) )
```

Contents

CPY: This operand indicates a change in the number of copies produced by a job on the output queue.

number: This operand specifies the number (1-99) of output copies to be produced by the job or job step. If the number of copies is altered while the job step is being punched or printed, the total number of copies produced will be the sum of the new number and the number of copies already completed. A default value of one copy is assumed if the operand is not specified.

PRTQ or *PCHQ:* This operand specifies the queue that contains the job or job step to which the change in the number of copies applies.

jobname: This operand specifies the job to which the change in the number of copies applies.

stepname: This operand specifies the job step to which the change in the number of copie's applies.

*characters** (5704-SC2 only):* This operand specifies the jobs and/or job steps to which the change in the number of copies applies. Multiple jobs/steps can be referenced by specifying one to seven characters followed by two asterisks (**). All jobs/steps on the queue whose names begin with the specified characters will have the number of copies changed.

Considerations

Unless the CHANGE copies command specifies stepname, all job steps within a job on the output queue receive the change.

CHANGE CORE

Function

Changes the amount of main storage required for one or more jobs on the reader queue. This command is valid only if spooling is active.

Format

```
Operation    Operand

CHANGE       CORE,size, ┌ RDRQ ┐ , ┌ jobname      ┐
(G)                     └ (RQ) ┘   └ characters** ┘
```

Contents

CORE: This operand indicates that a job on the reader queue is to have its main storage size requirements changed.

size: This operand specifies the amount of main storage required to execute the largest step in the job. (It does not assign main storage to the partition; the SET command must be used for this purpose.) The number specified (1-3 digits) can range from 0 to 238 (for 5704-SC1) or from 0 to 488 (for 5704-SC2). If 0 is specified, the current partition size is assumed.

RDRQ: This operand specifies the queue that contains the job to which the change applies. Since only the reader queue is valid for this command, this operand can be omitted as shown in the following example:

```
CHANGE CORE,8,,SAMPLE
```

jobname: This operand specifies the job to which the change of main storage size applies.

*characters** (5704-SC2 only):* This operand specifies the jobs to which the change of main storage size applies. Multiple jobs can be referenced by specifying one to seven characters followed by two asterisks (**). All jobs on the queue whose names begin with the specified characters will have the storage requirement changed.

Considerations

You cannot specify stepname to change main storage requirements; jobs on the reader queue can be controlled only on a job basis. If the JOB statement for the job to be changed is in error, the change is not allowed.

CHANGE FORMS TYPE

Function

Changes the forms type characters associated with one or more jobs or job steps on the print queue. This command is valid only if spooling is active.

Format

```
Operation  Operand

CHANGE     FRM,forms type,  ┌PRTQ┐ , ┌jobname      ┐  ┌,stepname     ┐
(G)                         └(WQ)┘   └characters** ┘  └,characters** ┘
```

Contents

FRM: This operand indicates a change in the forms type characters for a job or job step on the print queue.

forms type: This operand can be any combination of one to three characters (except commas or blanks) to indicate the forms type used. These characters are examined by the print writer when printing spooled output. If the forms type for the current print step is different from that of the previous print step, the system issues a message to the operator indicating the forms type to be used. The response taken to this message determines if separator pages are to be printed between print steps.

PRTQ: This operand specifies the queue that contains the job to which the change in the forms type applies. Since only the print queue is valid for this function, the queue operand can be omitted as shown in the following example:

```
CHANGE FRM,ABC,,JOB1,STEP1
```

jobname: This operand specifies the job to which the change of forms type applies.

stepname: This operand specifies the step to which the change of forms type applies.

*characters** (5704-SC2 only):* This operand specifies the jobs and/or job steps to which the change of forms type applies. Multiple jobs/steps can be referenced by specifying one to seven characters followed by two asterisks (**). All jobs/steps on the queue whose names begin with the specified characters will have the forms type changed.

Considerations

Unless the CHANGE forms type command specifies stepname, all job steps within a job on the print queue receive the change.

The CHANGE forms type command does not affect a print step that is currently printing.

CHANGE PARTITION

Function

Changes the partition assignment of one or more jobs on the reader queue. This command is valid only is spooling is active.

Format

```
Operation   Operand

CHANGE      PTN, [1], [RDRQ] , {jobname      }
(G)              [2]  [(RQ)]   {characters** }
                 [3]
                 [A]
                 [B]
                 [C]
                 [D]
```

Contents

PTN: This operand indicates that the partition assignment for a job on the reader queue is to be changed.

```
[1]
[2]
[3]
[A]
[B]
[C]
[D]
```

: This operand indicates the partition to which the job will be assigned for execution:

1	–	Partition 1
2	–	Partition 2
3	–	Partition 3
A	–	Partition 1 or 2
B	–	Partition 1 or 3
C	–	Partition 2 or 3
D	–	Partition 1, 2, or 3

If A, B, C, or D is used, spool allows a job to be executed in whichever of the specified partitions next requests a job for execution (see *JOB Statement* in Chapter 6).

RDRQ: This operand specifies the queue to which the change applies. Since only the reader queue is valid for this command, the queue operand can be omitted as shown in the following example:

```
CHANGE PTN,2,,SAMPLE
```

jobname: This operand specifies the job to which the change of partition assignment applies.

*characters** (5704-SC2 only):* This operand specifies the jobs to which the change of partition assignment applies. Multiple jobs can be referenced by specifying one to seven characters followed by two asterisks (**). All jobs on the queue whose names begin with the specified characters will have the partition changed.

Considerations

You cannot specify stepname to change partition; jobs on the reader queue can be controlled only on a jobname basis. If the JOB statement for the job to be changed is in error, the change is not allowed.

CHANGE PRIORITY

This command is valid only for systems that support spooling, and only when spooling is active.

Functions

- Changes the priority of one or more jobs on the reader queue.
- Changes the priority of one or more jobs or job steps on the print or punch queue.

Format

```
Operation   Operand

CHANGE                  ⎧ RDRQ,  {jobname      }
(G)                     ⎪ (RQ)   {characters** }
                 ⎧1⎫    ⎪
                 ⎪2⎪    ⎪ PRTQ,  {jobname      }  [,stepname     ]
        PTY,    <3> ,  <  (WQ)   {characters** }  [,characters** ]
        (Y)      ⎪4⎪    ⎪
                 ⎩5⎭    ⎪ PCHQ,  {jobname      }  [,stepname     ]
                        ⎩ (PQ)   {characters** }  [,characters** ]
```

Contents

PTY (Y): This operand indicates that the priority of a job or job step on one of the queues is to be changed.

1, 2, 3, 4, or 5: This operand is the number of the priority to be assigned to the job or step. The highest priority is 5.

RDRQ (RQ): This operand indicates that the reader queue contains the job for which the priority is to be changed. The jobname parameter identifies the specific job. If the priority of a job on the reader queue is changed, the output from that job has the changed priority assigned to it, unless the job steps within that job contain the PRIORITY parameter on a PRINTER or PUNCH statement.

PRTQ (WQ): This operand indicates that the print queue contains the job or step for which the priority is to be changed. When only the jobname parameter is used, the priority of all steps in the job is changed to the same value. When both jobname and stepname parameters are used, the priority of only the specified step is changed.

PCHQ (PQ): This operand indicates that the punch queue contains the job or step for which the priority is to be changed. When only the jobname parameter is used, the priority of all steps in the job is changed to the same value. When both jobname and stepname parameters are used, the priority of only the specified step is changed.

jobname: This operand specifies the job to which the change of priority applies.

stepname: This operand specifies the job step to which the change of priority applies.

*characters** (5704-SC2 only):* This operand specifies the jobs and/or job steps to which the change of priority applies. Multiple jobs/steps can be referenced by specifying one to seven characters followed by two asterisks (**). All jobs/steps on the queue whose names begin with the specified characters will have the priority changed.

Considerations

All jobs or job steps are removed from the queue on a first-in, first-out basis within priority. When the priority is changed, the job or step is removed from the queue only after all other jobs (or steps) with the same priority have been removed.

Consider the following example:

```
CHANGE PTY,5,RDRQ,JOBA
or
G Y,5,RQ,JOBA
```

Assuming the priority of JOBA was 1, it is changed to 5 and scheduled for execution after all preceding priority 5 jobs have been executed. Output on the print or punch queues for JOBA is also changed from 1 to 5, unless the job steps within JOBA contain the PRIORITY parameter on a PRINTER or PUNCH statement.

DISPLAY

Functions

- Displays system status information.
- Displays the history of the entire system.
- Displays the history of one to seven tasks (5704-SC2 only).
- Displays the reader, print, and punch queues.

Format

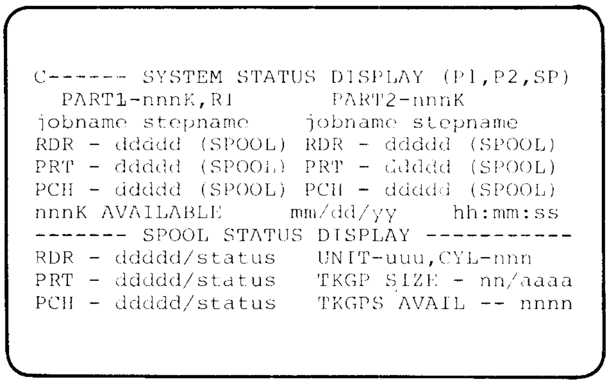
```
Operation    Operand

DISPLAY      ┌ STATUS  ┌,P1┐                 ┐
(D)          │         │,P2│                 │
             │         │,P3│                 │
             │         └,N ┘                 │
             │ HISTORY [,task]  [,task] . . .│
             │ (H)                           │
             │ RDRQ                          │
             │ (RQ)                          │
             │ PRTQ                          │
             │ (WQ)                          │
             │ PCHQ                          │
             └ (PQ)                          ┘
```

Contents

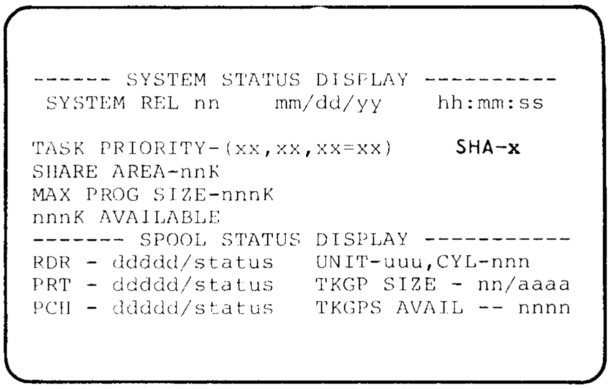
STATUS: When this operand is specified (or if no operand is specified), the status of the system is displayed on the CRT as follows:

SCP 5704-SC1



```
                                                   Line
C------ SYSTEM STATUS DISPLAY (P1,P2,SP)           1
  PART1-nnnK,R1          PART2-nnnK                2
jobname stepname     jobname stepname              3
RDR - ddddd (SPOOL) RDR - ddddd (SPOOL)            4
PRT - ddddd (SPOOL) PRT - ddddd (SPOOL)            5
PCH - ddddd (SPOOL) PCH - ddddd (SPOOL)            6
nnnK AVAILABLE      mm/dd/yy    hh:mm:ss           7
------- SPOOL STATUS DISPLAY -----------           8
RDR - ddddd/status   UNIT-uuu,CYL-nnn              9
PRT - ddddd/status   TKGP SIZE - nn/aaaa           10
PCH - ddddd/status   TKGPS AVAIL -- nnnn           11
```

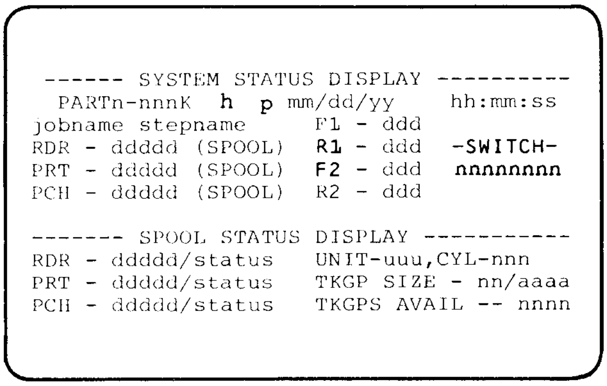
SCP 5704-SC2



```
                                                   Line
------ SYSTEM STATUS DISPLAY ----------            1
 SYSTEM REL nn    mm/dd/yy    hh:mm:ss             2
                                                   3
TASK PRIORITY-(xx,xx,xx=xx)    SHA-x               4
SHARE AREA-nnK                                     5
MAX PROG SIZE-nnnK                                 6
nnnK AVAILABLE                                     7
------- SPOOL STATUS DISPLAY -----------           8
RDR - ddddd/status   UNIT-uuu,CYL-nnn              9
PRT - ddddd/status   TKGP SIZE - nn/aaaa           10
PCH - ddddd/status   TKGPS AVAIL -- nnnn           11
```

The information displayed for SCP 5704-SC1 is described in Figure 5-1; information for SCP 5704-SC2 is in Figure 5-2. The spool status display information (lines 8-11) is identical for both program numbers and is described only in Figure 5-1.

SCP 5704-SC2 displays the status of each partition and the names of all main data areas and simulation areas separately. To move forward through the status display sequence (system, partition 1, partition 2, partition 3, names, system . . .), press ENTER in response to each display; or, to skip forward or backward to a specific display, key the partition number (1, 2, or 3), S (for system), or N (for names) over the cursor and press ENTER. The partition status is displayed as follows:



```
                                                   Line
------ SYSTEM STATUS DISPLAY ----------            1
  PARTn-nnnK  h  p mm/dd/yy    hh:mm:ss            2
jobname stepname      F1 - ddd                     3
RDR - ddddd (SPOOL)   R1 - ddd   -SWITCH-          4
PRT - ddddd (SPOOL)   F2 - ddd   nnnnnnnn          5
PCH - ddddd (SPOOL)   R2 - ddd                     6
                                                   7
------- SPOOL STATUS DISPLAY -----------           8
RDR - ddddd/status   UNIT-uuu,CYL-nnn              9
PRT - ddddd/status   TKGP SIZE - nn/aaaa           10
PCH - ddddd/status   TKGPS AVAIL -- nnnn           11
```

The partition status display is described in Figure 5-3.

To cancel the system status display for 5704-SC1, press ENTER. To cancel the 5704-SC2 system status displays, key a C over the cursor and press ENTER.

P1, P2, P3, or N: (5704-SC2 only) This operand specifies the first status to display and bypasses the system status display.

Line	Characters Displayed	Meaning
1	C	Cancel; if ENTER is pressed, display is canceled.
	SYSTEM STATUS DISPLAY	Identifies information in upper half of display.
	(xx,xx,xx)	Task priority, listed in descending order; for example, P1,P2,SP.
2	PART1-nnnK PART2-nnnK	Size of partitions 1 and 2.
	RI	Partition has been rolled out; a rollin is pending.
3	jobname stepname	Identity of jobs running in partitions 1 and 2.
4 5 6	RDR – ddddd PRT – ddddd PCH – ddddd	Reader, printer, and punch devices assigned to each partition by system generation, OCL statements, or OCC commands.
	(SPOOL)	Displayed if a device is assigned to spooling.
7	nnnK AVAILABLE	Main storage not assigned to partition 1 or 2.
	dd/mm/yy or mm/dd/yy	System date, in format specified during system generation.
8	SPOOL STATUS DISPLAY	Identifies information in lower half of display. Lines 8-11 are blank if spooling is not supported by the system.
9 (10) (11)	RDR – ddddd PRT – ddddd PCH – ddddd	Reader, printer, and punch devices assigned to spooling during system generation.
	/status	Status of the spool queues: ACT – Queue contains jobs or steps to process. EMP – Queue is empty. H – Queue is held via the HOLD command.
9	UNIT-uuu	Unit containing the disk space used by spool.
	CYL-nnn	Number of cylinders assigned to the spool area during system generation or IPL.
10	TKGP SIZE – nn/aaaa	Track group information: nn – Number of tracks per group (1, 2, 4, 5, 10) aaaa – Total number of track groups within the spool disk space
11	TKGPS AVAIL – nnnn	Number of track groups currently available.

Figure 5-1. Explanation of System Status Display (5704-SC1)

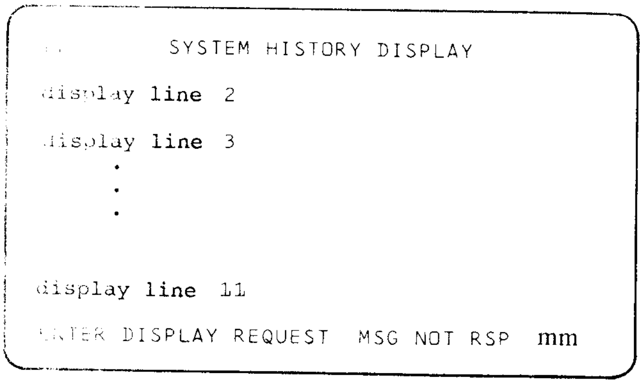
Line	Characters Displayed	Meaning
1	SYSTEM STATUS DISPLAY	Identifies information in upper half of display.
2	SYSTEM REL nn	Release level of the system program.
	mm/dd/yy or dd/mm/yy	System date, in format specified during system generation.
	hh:mm:ss	Time, in hours, minutes, and seconds.
4	TASK PRIORITY- (xx,xx,xx=xx)	Task priority, listed in descending order; for example, SP,P3,P2,P1, where SP is the highest priority and P1 is the lowest. Where equal priority has been specified in the PTY command, equal signs (=) appear instead of commas.
	SHA-x	System history area status: halt (H)/nohalt (N)/CCP auto (C)
5	SHARE AREA-nnnK	Size of the file share area assigned to the system.
6	MAX PROG SIZE-nnK	Maximum size program that can be run in a partition.
7	nnK AVAILABLE	Main storage not assigned to a partition or file share area.
8-11	SPOOL STATUS DISPLAY . . .	See Figure 5-1.

Figure 5-2. Explanation of System Status Display (5704-SC2)

Line	Characters Displayed	Meaning
1	SYSTEM STATUS DISPLAY	Identifies information in upper half of display.
2	PARTn-nnnK	Partition number and size.
	h	Halt (H)/nohalt (N) status.
	p	Partition stopped via OCC (P).
	mm/dd/yy or dd/mm/yy	System date, in format specified during system generation.
	hh:mm:ss	Time, in hours, minutes, and seconds.
3	jobname stepname	Identity of job running in this partition.
3 (4) (5) (6)	F1 - ddd R1 - ddd F2 - ddd R2 - ddd	Simulation areas assigned as F1, R1, F2, and R2 for this partition.
4 (5) (6)	RDR - ddddd PRT - ddddd PCH - ddddd	Reader, printer, and punch devices assigned to this partition.
4 (5) (6)	(SPOOL) (SPOOL) (SPOOL)	Displayed if the reader, printer, or punch device is assigned to spooling.
4 (5)	-SWITCH- nnnnnnnn	External indicator switch status.
8-11	SPOOL STATUS DISPLAY . . .	See Figure 5-1.

Figure 5-3. Explanation of Partition Status Display (5704-SC2)

HISTORY (H): This operand displays the contents of the system history area. The display is on 11 lines of the screen with lines 2 through 11 used for the actual display.



```
fnn          SYSTEM HISTORY DISPLAY

display line  2

display line  3
      .
      .
      .


display line  11

ENTER DISPLAY REQUEST  MSG NOT RSP  mm
```

Note: The top line controls the direction of the display.

f = { F Forward
B Backward
C Cancel the display }

nn applies to the F and B functions and specifies how many lines to roll each time the ENTER key is pressed. nn can range from 1 to 99.

task (5704-SC2 only): One to seven of the following operands specify that the history of a single task or of a combination of tasks is to be displayed:

Task Operand	**Displays**
P1	History of partition 1
P2	History of partition 2
P3	History of partition 3
S	System messages
SP	Spool messages
OCC	Operator control commands
CCP	History of all CCP tasks
CCU	History of all CCP *user* tasks (group 1 below)
CCS	History of all CCP *system* tasks (group 2 below)
Cxx	History of one or more of these individual CCP tasks:

Group 1

C44	C99	CVV
C55	CEE	CWW
C66	CGG	CXX
C77	CHH	CYY
C88	CUU	CZZ

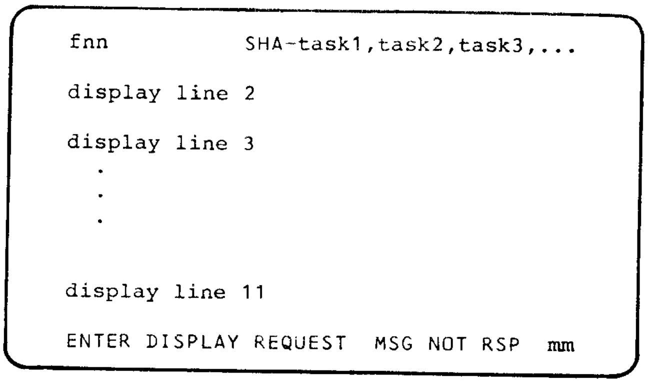
Group 2

CCC (communication manager)
CDD (display format facility—DFF)
CMM (communication scheduler)
CPP (command processor)
CTT (termination task)

The task operands are separated by commas, as in the following examples:

DISPLAY HISTORY,P1,P2,S,SP,OCC,CCU,CDD
or
D H,S,SP,CCP

When task operands are specified, the history of the tasks is displayed as follows:



```
fnn          SHA-task1,task2,task3,...

display line 2

display line 3
  .
  .
  .


display line 11

ENTER DISPLAY REQUEST  MSG NOT RSP  mm
```

RDRQ (RQ): This operand is used to display information about the reader queue (see *Considerations*).

PRTQ (WQ): This operand is used to display information about the print queue (see *Considerations*).

PCHQ (PQ): This operand is used to display information about the punch queue (see *Considerations*).

Considerations

Queue Display Format

The format of the display is:

```
fnn qqq QUEUE  (h)
display line 2
display line 3




display line 11

ENTER DISPLAY REQUEST  MSG NOT RSP  mm
```

You use line 1 to control scrolling of the display. To change the display, key the control codes into the appropriate positions, then press the ENTER key.

The meanings of the pqsitions in line 1 are:

f Enter F to roll the queue display, enter C to cancel the display. If neither F or C is entered, F is assumed.

nn Specifies the number of new lines to be added to the next display when the ENTER key is pressed.

qqq Indicates which queue is being displayed.

(h) Is H when the queue is currently being held. A queue is held by specifying it in the HOLD command. Lines 2 through 11 display the contents of the queue.

If, after displaying the contents of one queue, you want to display the contents of another, you can do so without entering another DISPLAY command. Enter the operand (RDRQ, RQ, PRTQ, WQ, PCHQ, or PQ) for the next queue you want to display in the first two or four positions of line 1 and press the ENTER key.

Reader Queue Display

The following is an example of a reader queue display containing the jobname, stepname, position on the queue, priority number, the amount of storage required, and the partition in which the job will be executed. The storage required is displayed only if it was specified on the JOB OCL statement or through the CHANGE command.

For 5704-SC2, jobs placed on the reader queue by $QCOPY under CCP will have the terminal name displayed in the stepname position.

```
F10     RDR QUEUE (H)   POS  PR CORE   PT
          ---PRIORITY LIMIT 3---
 FIRST                    1  A5  10K   1
 JOB-E                    2   5        1
 JOB-F                    3 HQ4   3K   2
 JOB-J                    4  H4        2
 JOB-L TERM01             5  H3  24K   3
 JOB-I TERM02             6 KA2        2
 JOB-M                    7  K1        1
 ACCOUNTS                 8   1 136K   3N
                END OF QUEUE
ENTER DISPLAY REQUEST       MSG NOT RSP 01
```

An (H) on the first line of the display indicates that the entire reader queue is held via the HOLD command.

For 5704-SC2, if the priority limit for jobs placed on the reader queue by $QCOPY under CCP is less than five, the limit is displayed on the second line of the display.

One or two of the following characters can precede the priority number in the display:

A – Indicates that the job is currently executing in the partition shown in the PT column (5704-SC2 only)

H – Indicates that the job is being held via the HOLD command

K – Indicates that the job is in a keep state via the KEEP command (5704-SC2 only)

Q – Indicates that the job is being copied by $QCOPY 5704-SC2 only)

For 5704-SC2, an N following the partition number indicates that QCOPY-NO was specified on the JOB OCL statement.

Print Queue Display

The following is an example of a display of the print queue containing the jobname, stepname, position on the queue, job priority, forms type, the number of copies to be printed, and the number of pages within each copy.

An (H) on the first line of the display indicates that the entire print queue is held via the HOLD command.

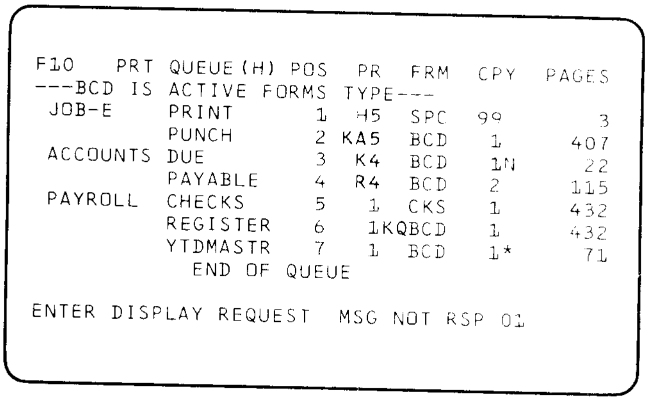
One or two of the following characters can precede the priority number in the display:

A – Indicates that the job step is currently being printed
H – Indicates that the job or job step is being held via the HOLD command
K – Indicates that the job or job step is in a keep state via the KEEP command (5704-SC2 only)
Q – Indicates that the job or job step is being copied by $QCOPY (5704-SC2 only)
R – Indicates that the job step was partially canceled with a REUSE command and is no longer being printed (5704-SC2 only)

For 5704-SC2, an N after the number of copies indicates that QCOPY-NO was specified on a PRINTER OCL statement.

An asterisk (*) after the number of copies indicates that the job step is currently executing and that a PRINTER statement with DEFER-NO was specified for the job step. (A job step that is executing without DEFER-NO specified is not displayed.)

The active forms type will be displayed in the message
  nnn IS ACTIVE FORMS TYPE
if the print writer was started using a forms type.



```
F10    PRT QUEUE(H) POS   PR   FRM   CPY   PAGES
---BCD IS ACTIVE FORMS TYPE---
 JOB-E     PRINT       1   H5   SPC   99       3
           PUNCH       2  KA5   BCD    1     407
 ACCOUNTS  DUE         3   K4   BCD    1N     22
           PAYABLE     4   R4   BCD    2     115
 PAYROLL   CHECKS      5    1   CKS    1     432
           REGISTER    6    1KQBCD     1     432
           YTDMASTR    7    1   BCD    1*     71
             END OF QUEUE

ENTER DISPLAY REQUEST  MSG NOT RSP 01
```

Punch Queue Display

The following is an example of a punch queue display containing the jobname, stepname, position on the queue, priority, card type, the number of copies to be punched, and the number of cards within each copy.

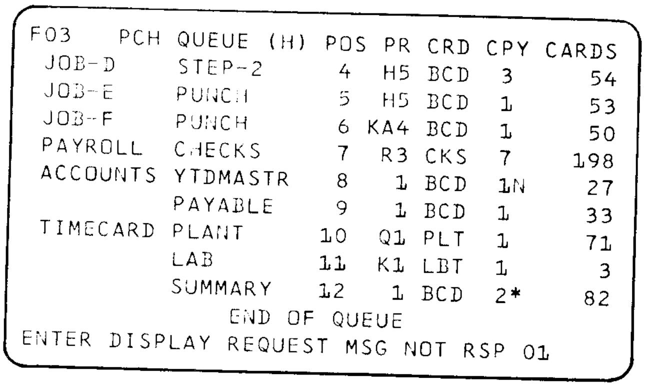
An (H) on the first line of the display indicates that the entire punch queue is held via the HOLD command.

One or two of the following characters can precede the priority number in the display:

A – Indicates that the job step is currently being punched
H – Indicates that the job or job step is being held via the HOLD command
K – Indicates that the job or job step is in a keep state via the KEEP command (5704-SC2 only)
Q – Indicates that the job or job step is being copied by $QCOPY (5704-SC2 only)
R – Indicates that the job step was partially canceled with a REUSE command and is no longer being punched (5704-SC2 only)

For 5704-SC2, an N after the number of copies indicates that QCOPY-NO was specified on a PUNCH OCL statement.

An asterisk (*) after the number of copies indicates that the job step is currently executing and that a PUNCH statement with DEFER-NO was specified for the job step. (A job step that is executing without DEFER-NO specified is not displayed.)



```
F03     PCH QUEUE (H) POS  PR CRD CPY CARDS
 JOB-D    STEP-2        4  H5 BCD  3     54
 JOB-E    PUNCH         5  H5 BCD  1     53
 JOB-F    PUNCH         6 KA4 BCD  1     50
 PAYROLL  CHECKS        7  R3 CKS  7    198
 ACCOUNTS YTDMASTR      8   1 BCD  1N    27
          PAYABLE       9   1 BCD  1     33
 TIMECARD PLANT        10  Q1 PLT  1     71
          LAB          11  K1 LBT  1      3
          SUMMARY      12   1 BCD  2*    82
               END OF QUEUE
ENTER DISPLAY REQUEST MSG NOT RSP 01
```

HIPTY

This command is valid only for 5704-SC2 systems that support spooled reading and only when spooling is active.

Function

Limits the priority of jobs placed on the active spool file reader queue by $QCOPY (spool file copy program) under CCP.

Format

Operation	Operand
HIPTY (HP)	high-priority

Contents

high-priority: This operand can be any value from 0 through 5; it specifies the high-priority limit of jobs placed on the active spool file reader queue by $QCOPY under CCP. For example, if you specify 3 as the high-priority operand, the highest priority allowed for jobs placed in the reader queue by $QCOPY under CCP is 3. If you specify 0, the priority is limited to 1 and held.

Considerations

A job with a JOB statement error will be placed on the queue with priority 5, regardless of the HIPTY limit.

A job being placed on the queue at the time the HIPTY command is entered will not be affected by the limit specified.

The limit specified remains in effect until another HIPTY command is entered or until an IPL is performed. After an IPL, the limit in effect is 5 unless otherwise specified by $CNFIG.

HOLD

This command is valid only for systems that support spooling, and only when spooling is active.

Functions

- Prevents one or more jobs from being executed from the reader queue.
- Prevents one or more jobs or job steps from being printed from the print queue.
- Prevents one or more jobs or job steps from being punched from the punch queue.
- Prevents all jobs from being executed, printed, or punched from the specified queues.

Format

```
Operation   Operand

HOLD       / RDRQ  [,jobname     ]                       \
(H)        | (RQ)  [,characters**]                        |
           |                                              |
           | PRTQ  [,jobname        [,stepname     ]]     |
           < (WQ)  [,characters**   [,characters**]]      >
           |                                              |
           | PCHQ  [,jobname        [,stepname     ]]     |
           | (PQ)  [,characters**   [,characters**]]      |
           \                                             /
```

Contents

RDRQ (RQ): This operand prevents the execution of any job from the reader queue; however, jobs can continue to be placed in the reader queue. Using the optional jobname parameter, only a particular job is held.

PRTQ (WQ): This operand prevents any output from being removed from the print queue; however, output from executing jobs or steps can continue to be placed in the print queue. Using the optional jobname parameter, all steps of the specified job are held. Using the jobname and stepname parameters, only a particular step is held.

PCHQ (PQ): This operand prevents any output from being removed from the punch queue; however, output from executing jobs or steps can continue to be placed in the punch queue. Using the optional jobname parameter, all steps of the specified job are held. Using the jobname and stepname parameters, only a particular step is held.

jobname: This operand specifies the job to be held.

stepname: This operand specifies the job step to be held.

*characters** (5704-SC2 only):* This operand specifies the jobs and/or steps to be held. Multiple jobs/steps can be referenced by specifying one to seven characters followed by two asterisks (**). All jobs/steps on the queue whose names begin with the specified characters will be held.

Considerations

The job or job step is held on the queue until released by a RELEASE command or canceled by a CANCEL command. If the entire queue is held, it remains held until released by a RELEASE command.

Holding a job on the reader queue has the same effect as specifying PRIORITY-0 on the JOB statement. The RELEASE command is required to remove a job from the hold state.

When a job that is still executing is held by jobname on the print or punch queue, only those steps of the job that are on the queue at the time of the HOLD command are held. Subsequent steps of the job placed on the queue are not held.

When used to hold jobs or steps, the HOLD command takes effect immediately. When used to hold an entire queue, the command does not take effect immediately. Consider the following examples:

1. HOLD PRTQ,JOBA
   or
   H WQ,JOBA

   If this command is entered while JOBA is being printed, the writer immediately terminates printing JOBA and continues with the next job on the queue. When JOBA is released, it is printed from the beginning, or from the beginning of the interrupted copy if multiple copies were requested. (This action also occurs for holding steps, as well as jobs, on the print and punch queues.)

2. HOLD PRTQ
   or
   H WQ

   When the entire queue is held, the step currently printing is completed before the queue is held. If multiple copies are requested, the queue is held when the current copy is complete. When the queue is released, printing begins with the next copy.

KEEP

This command is valid only for 5704-SC2 systems that support spooling and only when spooling is active.

Function

Causes one or more jobs or job steps to be placed back on the queue in the hold state after they have been printed, punched, or executed.

Format

```
Operation   Operand

KEEP       ⎧ RDRQ,  ⎧ jobname      ⎫                                ⎫
(KP)       ⎪ (RQ)   ⎩ characters** ⎭                                ⎪
           ⎪                                                        ⎪
           ⎪ PRTQ,  ⎧ jobname      ⎫  ⎡ ,stepname     ⎤             ⎪
           ⎨ (WQ)   ⎩ characters** ⎭  ⎣ ,characters** ⎦             ⎬
           ⎪                                                        ⎪
           ⎪ PCHQ,  ⎧ jobname      ⎫  ⎡ ,stepname     ⎤             ⎪
           ⎩ (PQ)   ⎩ characters** ⎭  ⎣ ,characters** ⎦             ⎭
```

Contents

RDRQ, PRTQ, or PCHQ: This operand specifies the queue that contains the job or job step to be kept.

jobname: This operand specifies the job to be kept.

stepname: This operand specifies the job step to be kept.

*characters** (5704-SC2 only):* This operand specifies the jobs and/or job steps to be kept. Multiple jobs/steps can be referenced by specifying one to seven characters followed by two asterisks (**). All jobs/steps on the queue whose names begin with the specified characters will be kept.

Considerations

The KEEP command cannot be used unless the job or job step is on the queue. The job or job step is kept in the queue until it prints, punches, executes, is released by a RELEASE command, or is canceled by a CANCEL command.

PTY

Function

Changes the priority of the following tasks in the system control program:

Spooling (reader, print writer, punch writer)
Partition 1
Partition 2
Partition 3

The priority of these tasks determines the order in which they receive the services of the system control program. The spooling task refers to the spooled reader and/or writer. The PTY command can be entered at any time.

At IPL, the priority of these tasks is SPOOL, P3, P2, P1, with SPOOL having the highest priority. If SPOOL is not used, the priority is P3, P2, P1, with P3 having priority over P2 and P1. If SPOOL is not used and P3 is not supported (5704-SC1), the priority is P2, P1, with P2 having priority over P1.

Format

Operation	Operand
PTY (Y)	$\left\{\begin{matrix}\text{SPOOL}\\ \text{(SP)}\\ \text{P1}\\ \text{P2}\\ \text{P3}\end{matrix}\right\}\left\{\begin{matrix},\\ =\\ -\end{matrix}\right\}\ldots$

Contents

SPOOL: This operand specifies the spooling task for a priority change.

P1, P2, or P3: These operands specify the partition to which the priority change applies.

Considerations

The order of the operand indicates the requested priority: the leftmost operand has the highest priority; the rightmost operand the lowest. Two or more tasks can be given equal priority by specifying an equal sign (=) or dash (-) instead of a comma (,) between the operands (5704-SC2 only).

The following rules apply to the operands on the PTY command:

1. Each operand can be specified only once.

2. SPOOL is an invalid operand if spooling is not supported or is canceled.

3. All operands must be specified; there are no defaults. The following examples illustrate the first three rules:

   ```
   P1,P2,P3,SP
   P1,P2,P3
   P1=P2=P3
   P1-P2-P3
   SP,P2,P1
   P1,P2
   ```

4. Only one group of operands may be equated. For example:

   SP=P3,P2=P1

   is not valid, but

   SP=P3=P2=P1 and SP,P3=P2=P1

   are valid.

5. The task that contains the communications control program (CCP) cannot be given equal priority with another task. For example:

   P3=SP,P2,P1

   is not valid if CCP is executing in P3, but

   P3,SP=P1=P2

   is valid.

To improve the throughput of nonspooling tasks, the priority of the spooling tasks can be lowered if:

- The spooled output (print or punch) is of low volume.

- Turnaround time is not important.

Changing the priority can increase the performance of one task at the expense of another task.

For example, to cause the spooling task to have a lower priority in the system control program than P1 or P2, but higher than P3, the following command can be entered:

```
PTY P2,P1,SPOOL,P3
or
Y P2,P1,SP,P3
```

READER

Function

Changes the system input device for a partition.

Format

```
Operation   Operand

READER      ┌P1┐
(RDR)       │P2│ , device
            └P3┘
```

(P1 is underlined as the default.)

Contents

P1, P2, or P3: This operand specifies the partition for which the system input device is to be changed. If the partition is not specified, P1 is assumed. The device parameter can be one of the following:

- MFCU1 – Primary hopper of the 5424
- MFCU2 – Secondary hopper of the 5424
- MFCM1 – Primary hopper of the 2560
- MFCM2 – Secondary hopper of the 2560
- 3741 – 3741 Data Station or Programmable Work Station (directly attached)
- 1442 – 1442 Card Read Punch
- 2501 – 2501 Card Reader
- CONSOLE – 3277 Display Station (CRT/Keyboard)

Considerations

When this command is entered, the specified partition must be at end of job and it must not have a system input device allocated to it at this time. The device assigned remains as the system input device until (1) changed by another READER command, (2) changed by a READER OCL statement, or (3) an IPL is performed.

The system input device for a spooled job stream can be changed; however, if the new system input device is different from the spooled input device, input spooling stops for that partition. Input for the partition comes from the new system input device instead of from the spooled input job stream.

To resume input spooling, the system input device must be changed back to the system input device that is accepting spooled input.

Consider the following example:

```
READER P1,MFCU1
or
RDR P1,MFCU1
```

The primary hopper of the MFCU is assigned as the system input device for partition 1.

RELEASE

This command is valid only for systems that support spooling, and only when spooling is active.

Functions

- Releases a queue or one or more jobs or job steps that were held by the HOLD command.
- Releases one or more jobs that were assigned a priority of 0 on the JOB statement.
- Releases one or more job steps that were assigned a priority of 0 on the PRINTER or PUNCH statement.

Format

```
Operation   Operand

RELEASE    / RDRQ  [,jobname      ]
(R)       |  (RQ)  [,characters** ]
          |
           \ PRTQ  [,jobname       [,stepname      ]]
           / (WQ)  [,characters**  [,characters**  ]]
          |
          |  PCHQ  [,jobname       [,stepname      ]]
          |  (PQ)  [,characters**  [,characters**  ]]
           \
```

Contents

RDRQ (RQ): This operand releases the reader queue. Any jobs on the queue that are in a hold state continue to be held. Using the optional jobname parameter, a particular job is released. The RDRQ operand is valid only on systems that support input spooling.

PRTQ (WQ): This operand releases the print queue. Any jobs or steps on the queue that are in a hold state continue to be held. Using the optional jobname parameter, all steps of the specified job are released. Using the jobname and stepname parameters, only a particular step is released.

PCHQ (PQ): This operand releases the punch queue. Any jobs or steps on the queue that are in a hold state continue to be held. Using the optional jobname parameter, all steps of the specified job are released. Using the jobname and stepname parameters, only a particular step is released. The PCHQ operand is valid only on systems that support punch spooling.

jobname: This operand specifies the job to be released.

stepname: This operand specifies the job step to be released.

*characters** (5704-SC2 only):* This operand specifies the jobs and/or job steps to be released. Multiple jobs/steps can be referenced by specifying one to seven characters followed by two asterisks (**). All jobs/steps on the queue whose names begin with the specified characters will be released.

Considerations

A released job or job step that was previously held by a HOLD command retains the priority assigned to it.

A job that was held by jobname is not released by using only a RDRQ, PRTQ, or PCHQ operand; it must be released by jobname.

A step that was held by jobname and stepname is not released by using only a RDRQ, PRTQ, or PCHQ operand; it must be released by jobname and stepname.

The RELEASE command does not always void all functions of the HOLD command. For example: A job step that was being printed or punched is held and then released while the writer is waiting or stopped; when the writer starts, it acts as if the job step is still held. The first available job step on the queue is printed or punched.

Consider the following example:

```
RELEASE RDRQ
or
R RQ
```

This command releases the reader queue previously held by a HOLD RDRQ command.

RESTART

This command is valid only for systems that support spooling, and only when spooling is active.

Functions

- If entered while the print or punch writer is active, restarts printed or punched output of that job step.
- If entered when the writer is not active, causes the writer to start with the job step that was stopped, if one was stopped, or from the beginning of the queue if no job steps were previously stopped.

Format

```
Operation     Operand
             ┌                  ┐
RESTART      │ PRT ┌,PAGE    ┐  │
(T)          │ (W) │(PG)     │  │
             │     │,number  │  │
             │     └         ┘  │
             │ PCH              │
             │ (P)              │
             └                  ┘
```

Contents

PRT (W): This operand causes the output of the step currently being printed to be restarted. If an operand is not specified for the RESTART command, PRT is assumed.

PAGE (PG): If entered when the print writer is active (not terminated), this operand causes the print writer to restart at the beginning of the current page. If entered when the print writer is not active, this operand causes the print writer to start printing from the beginning of the first page not previously finished. This operand will not be accepted after page 65535 has been printed. If PAGE is specified when starting the print writer after an IPL, the specification is ignored and page 1 is started.

number: This operand specifies the number (1 to 65535) of the page to be started. If the number specified is higher than any page in the job step, spool issues an error message (or waits for the specified page if the job step is still executing).

The page number reflects the pages in the spool queue, not necessarily the page numbers on the user report. When a step is terminated, the message indicates the page number at that time. Also, the display of the print queue shows the number of pages for that step in the print queue.

PCH (P): This operand causes output from the step currently being punched to be restarted from the beginning.

Considerations

When the RESTART PRT or RESTART PCH command is entered, the printing or punching of the current job step is immediately terminated. Depending on the command entered, the message PRT RESTART PENDING or PCH RESTART PENDING is displayed. The message allows the operator to correct any situation that may have caused the RESTART to be entered. Response to the message initiates the restart.

A STOP command is not required prior to issuing a RESTART command; however, a RESTART command can be entered following a STOP command.

When the print writer is restarted at a specified page, the restart may not occur immediately. The length of time required to find the page in the spool file varies depending on the page number required, the priority of the spool print writer in relation to the program partitions, and the track group size used in the spool file.

If the print or punch writer is producing output for a job step that is currently executing and a RESTART writer command is entered while the writer is waiting for more data from the job step, the writer is not restarted until the job step produces the required data for the writer or until the job step completes.

You should respond to any I/O error messages before entering the RESTART PRT or RESTART PCH command. The RESTART command is not executed until you have responded to these I/O error messages.

Consider the following example:

```
RESTART
or
T
```

This command causes the print writer to immediately terminate printing the current job step. When the operator responds to the message PRT RESTART PENDING, the print writer prints output of the job step starting with the first page.

REUSE

This command is valid only for 5704-SC2 systems that support spooling, and only when spooling is active.

Function

Frees space in the spool file from the currently printing or punching job step (before printing or punching is complete), making that space available for other uses.

Format

```
Operation   Operand

REUSE       { PRTQ                           }
(RU)        { (WQ) ,jobname[,stepname]       }
            {                                }
            { PCHQ                           }
            { (PQ) ,jobname[,stepname]       }
```

Contents

PRTQ or PCHQ: This operand specifies the queue that contains the currently printing or punching job step to which the reuse applies.

jobname: This operand specifies the job to which the reuse applies.

stepname: This operand specifies the job step to which the reuse applies.

Considerations

The REUSE command is most useful when the SPOOL FILE FULL message is issued. The space freed from the currently printing or punching job step can be used for any spooled input or output from any partition.

If spool is stopped after space is freed from a print step or if a print writer is stopped and then started with a different forms type operand, that print step can be restarted only if you enter a RESTART command with the number of a page that is still in the queue. To restart, you must:

1. Stop the print writer.

2. Move the reused job step to the front of the queue (either by changing the priority or by holding other jobs).

3. Enter the RESTART command.

If spool is stopped after space is freed from a punch step, that punch step cannot be restarted.

When a REUSE command is entered, the number of copies for the job step is set to 1 and cannot be changed. Also, a HOLD command is not allowed for that job step if the step is still being printed or punched.

After a REUSE command is entered for a job step, the spool file that contains that step can be used only by SCP 5704-SC2 (version 2 or later).

Consider the following example:

```
REUSE PRTQ,JOBA
or
RU WQ,JOBA
```

If this command is entered while JOBA is being printed, the command will be accepted and disk space in the spool file will be made available for use by other spooled input or output.

START

Functions

- Activates spooling immediately after IPL.
- Starts the input spooling routine (reader), print writer, or punch writer.
- Starts spooling in a partition that has had spooling stopped by the STOP command.

  Note: The preceding functions are valid only for systems that support spooling.

- Starts program execution in a partition (5704-SC2 only).

Format

```
Operation   Operand

START      /  SPOOL   ┌ ,P1                          ┐ \
(S)        |  (SP)    │ ,P2                          │  |
           |          │ ,P3                          │  |
           |          │ [,NEW]  [,unit]  [,size]     │  |
           |          │    [,track group size]       │  |
           |          └                              ┘  |
           |  RDR  [,ROLL]                              |
           |  (R)                                       |
           <                                            >
           |  PRT  [,forms type]                        |
           |  (W)                                       |
           |  PCH                                       |
           |  (P)                                       |
           |  P1                                        |
           |  P2                                        |
           \  P3                                        /
```

Contents

SPOOL (SP): This operand activates spooling at IPL or starts spooling after it was stopped in a partition. If used with the NEW, unit, size, and/or track group size parameters, the SPOOL operand can be entered only at IPL prior to initiation of the first job. If any of these values is omitted, the value specified during system generation is used. Once spooling is activated by the START command, spooling remains active until you enter a STOP command.

P1, P2, or P3: This optional parameter of the SPOOL operand is used to start spooling in a partition in which spooling was stopped by a STOP command. The START command specifying a partition can be entered only after the spooling function has been activated. If the START command with the partition parameter is entered at the end of a job when the system input device has been released, spooling for the partition begins immediately. If it is entered during the execution of a job, it takes effect at the next end of job (not end of step) that releases the system input device.

NEW: This optional parameter of the SPOOL operand causes the spooling area on disk to be reformatted. Previous input or output in this area is lost. When NEW is specified and data is found in the spool file, the message THE SPOOL FILE CONTAINS DATA warns that data will be destroyed if a continue option is taken. Options to the message allow the operator to continue or cancel the START SPOOL,NEW request. If NEW is not specified, existing data in the spooling area is not lost. If no data exists, the area is reformatted.

Unit: This optional parameter of the SPOOL operand changes the disk unit on which spooled records are written. The specified disk unit overrides the unit specified during system generation and remains in effect until the next IPL. If the same spool file is to be used after the next IPL, the unit parameter must again be specified. See Figure 1-1 for a list of possible disk units.

Size: This optional parameter specifies the size of the spooling area, in number of cylinders. The maximum size that may be specified is 199 for a 5445, 166 for a 3340 main data area, and 186 for a 3344 main data area. This parameter overrides the value specified during system generation only when NEW is specified, and it remains in effect until that spooling disk area is reformatted or a new area is built on the disk pack. Whenever size is specified, NEW must also be specified.

Track Group Size: This optional parameter of the SPOOL operand changes the size of the track groups within the spooling disk area. Valid sizes that can be specified at IPL are 1, 2, 4, 5, and 10. This parameter overrides the value specified during system generation only when NEW is specified, and it remains in effect until that spooling disk area is reformatted or a new area is built on the disk pack. Whenever track group size is specified, NEW must also be specified.

RDR (R): This operand activates the input spooling routine that reads from the spooled reader and places the jobs in the reader queue. This operand is used to start the spooled reader following IPL or after the reader has been terminated. Spooling must be active prior to entering a START RDR command and the punch writer must not be active.

ROLL: The optional ROLL parameter allows cards read on the MFCU to be placed in each of the four stackers before operator intervention is required. Stopping the reader with the STOP RDR command and then restarting without the ROLL parameter causes all the cards read to be placed in only one stacker.

PRT (W): This operand activates the print writer so printing of spooled records from the print queue can begin when the spooled printer is available. The PRT operand starts the print writer following IPL or after the print writer has been terminated. If used after a STOP command, printing begins at the point at which the writer was terminated. Spooling must be active prior to entering a START PRT command.

Forms Type: When the forms type operand is supplied, the print writer prints only those job steps in which the forms type matches that given in the START command. For example, START PRT,ABC causes only those job steps in which the forms type is ABC to be printed.

To change the forms type, the print writer must be terminated (STOP PRT), and a new START command specifying different forms type must be entered.

If the print writer is started without the forms type operand, it prints job steps in the order in which they occur on the print queue.

PCH (P): This operand activates the punch writer so punching of spooled records from the punch queue may begin when the spooled punch device is available. This operand starts the punch writer following IPL or after the punch writer has been terminated. When used after a STOP command, punching begins at the point in which the writer was terminated. Spooling must be active prior to entering a START PCH command, and the reader must not be active.

P1, P2, or P3: (5704-SC2 only) These operands are used to start program execution in a partition in which execution was stopped by a STOP command. The START Px command can be entered at any time and takes effect immediately.

Considerations

If the START SPOOL command is not entered following IPL, the system activates spooling according to the specifications given during system generation. This command is valid only on systems that support spooling.

If START PRT or START PCH is entered while the respective writer is active, the START command is ignored.

Once a disk drive has been specified as the spool unit, the disk pack on that drive should not be removed (unless an IPL procedure is to be performed) even though all of the spooling functions have been terminated.

Consider the following examples:

1. START SPOOL,NEW,D1,50,5
   or
   S SP,NEW,D1,50,5

   This command, given at IPL, reformats the spool area on disk. Existing data in the spool file is lost. The spool data area is built on drive 1, is 50 cylinders in size, and has a track group size of 5 tracks. The operator will be informed by the display if previous data exists in the spool file.

2. START SPOOL,, D1
   or
   S SP,, D1

   If there is any data in the spool queues, this command, given at IPL, does not reformat the spool area on disk; the number of cylinders used for spooled data is not changed. If there is no data in the spool queues, the spool area is reformatted, and the number of cylinders and track group size are the values specified during system generation. In either case, the disk unit is D1.

3. START SPOOL,P2
   or
   S SP,P2

   If spooling is stopped in partition 2, this command starts spooling in partition 2 after the next end of job (not end of step).

STOP

Functions

- Terminates spooling in a partition.

- Terminates the reader, print writer, or punch writer.

  Note: The preceding functions are valid only for systems that support spooling.

- Stops program execution in a partition (5704-SC2 only).

Format

```
Operation   Operand

STOP      /  SPOOL  { ,P1 }    \
(P)      |   (SP)   { ,P2 }     |
         |          { ,P3 }     |
         |                      |
         |   RDR                |
         |   (R)                |
         |                      |
         |         [ ,STEP ]    |
        <    PRT   [  (ES) ]     >
         |   (W)   [ ,PAGE ]    |
         |         [  (PG) ]    |
         |                      |
         |   PCH   [ ,STEP ]    |
         |   (P)   [  (ES) ]    |
         |                      |
         |   P1                 |
         |   P2                 |
          \  P3                /
```

Contents

SPOOL (SP): This operand causes spooling to be terminated in the specified partition: partition 1 (P1), partition 2 (P2), or partition 3 (P3). After IPL, the STOP SPOOL command can be entered at any time following initiation of spooling. The command takes effect at the next end of job (not end of job step) when the system input device is released. The command takes effect immediately if it is entered at end of job. Spooling remains stopped in that partition until a START SPOOL command is entered in the partition. When spooling has been stopped, jobs execute in the same manner as in a system without spool support; that is, the jobs use the unit-record devices directly.

RDR (R): This operand causes the input spooling routine (reader) to terminate immediately, releasing the input device for other use. The job currently being spooled is not put on the reader queue.

PRT (W): This operand causes the print writer to terminate, releasing the printer for other use. The job step currently being printed is kept on the queue; it can be continued from the point of interruption using a START PRT command, or the print writer can be started from the beginning of any page using a RESTART command.

The STEP (ES) operand causes the print writer to delay termination until the current step or copy is completed and removed from the queue. If the print writer is already at the end of a step, it ignores this operand and terminates immediately.

The PAGE (PG) operand causes the print writer to delay termination until the current page is completed. If the print writer is at the end of a print step, it ignores this operand and terminates immediately.

Note: If neither the STEP nor the PAGE operand is entered, the print writer terminates immediately.

PCH (P): This operand causes the punch writer to terminate, releasing the punch device for other use. The job step currently being punched is kept on the queue; it can be continued from the point of interruption using a START PCH command, or it can be started from the beginning using a RESTART PCH command.

The STEP (ES) operand causes the punch writer to delay termination until the current step or copy is completed and removed from the queue. If the punch writer is already at the end of a step, it ignores this operand and terminates immediately. If this operand is not entered, the punch writer terminates immediately.

P1, P2, or P3: (5704-SC2 only) These operands are used to stop program execution in a partition. This command cannot be entered for the partition in which the communications control program (CCP) is executing or when a partition is using certain system resources, such as the system interlocks. This command takes effect immediately. A START command is required to resume program execution after it has been stopped.

Note: The STOP Px command can also cause the spool writers to stop execution if they are printing or punching data from the job step when it is stopped. They will resume printing or punching after execution has been started in the partition.

Considerations

Once a disk drive has been specified as the spool unit, the disk pack on that drive should not be removed (unless an IPL procedure is to be performed) even though all of the spooling functions have been terminated.

If the print or punch writer is producing output for a job step that is currently executing and a STOP writer command is entered while the writer is waiting for more data from the job step, the writer is not stopped until the job step produces the required data for the writer or until the job step completes.

You should respond to any I/O error messages before entering the STOP RDR, STOP PRT, and STOP PCH commands. The STOP command is not executed until you have responded to these I/O error messages.

Consider the following example:

```
STOP SPOOL,P1
or
P SP,P1
```

This command causes spooling to be terminated when partition 1 is at end of job (not end of job step).

Chapter 6. OCL Considerations

The following Model 15 OCL statements contain information that has special considerations for a spooling environment:

- ASSIGN
- JOB
- PRINTER
- PUNCH
- READER
- /.

ASSIGN

The ASSIGN statement is supported only by SCP 5704-SC2.

Function

Reassigns logical simulation areas (F1, R1, F2, R2) to any of the supported physical simulation areas. Each logical 5444 unit (F1, R1, F2, R2) for a given partition must be assigned to a unique physical simulation area. The system pack cannot be reassigned; therefore, a maximum of nine user-assigned simulation areas is allowed (three per partition). Simulation area assignments remain in effect until another ASSIGN statement is received or until IPL. An ASSIGN statement placed in a job ensures that, no matter which partition the job is executed in, the correct simulation areas are assigned.

Placement

The ASSIGN statement may appear anywhere among the OCL statements. In a procedure, it must precede the RUN statement (if RUN is used).

Format

```
// ASSIGN F1-physical simulation area
          R1-physical simulation area
          F2-physical simulation area
          R2-physical simulation area
```

Example

```
// ASSIGN R2-D3E,F2-D2B
```

JOB

The JOB statement is required when spooling is active.

Function

Groups related job steps together to ensure sequential execution. (The system does not initiate the next step of a job until the previous step has come to a successful completion.)

Placement

The JOB statement must precede the first LOAD or CALL statement for a job. It cannot be used in a procedure. (Since a job cannot be executed as an inquiry program, the JOB and /. statements are ignored if a rollin is pending for that partition.)

Format

//jobname JOB parameters

jobname: This is a required entry used to uniquely identify a job. It must begin in position 3, may not exceed 8 characters, and may not contain a comma. Job (and step) names should contain only characters that are on the 3277 Display Station keyboard; otherwise, that job cannot be referenced by its name using OCC.

The jobname is displayed on the screen with system messages associated with the job. It is also the name by which jobs are identified on the queues.

When jobs are being read onto the reader queue, any errors on the JOB statement cause that job to be assigned a priority of 5. If the jobname is missing or invalid, the default jobname, JOB, is assigned to that job on the reader queue.

Contents

All parameters are optional. They are as follows (defaults are underlined):

- PRIORITY-$\left\{\begin{array}{c}0\\ \underline{1}\\ 2\\ 3\\ 4\\ 5\end{array}\right\}$

- PARTITION-$\left\{\begin{array}{c}\underline{1}\\ 2\\ 3\\ A\\ B\\ C\\ D\end{array}\right\}$

- CORE-nnn (1 to 3 digits)

- SPOOL-$\left\{\begin{array}{c}\underline{\text{YES}}\\ \text{NO}\end{array}\right\}$

- QCOPY-$\left\{\begin{array}{c}\underline{\text{YES}}\\ \text{NO}\end{array}\right\}$

PRIORITY: A priority can be assigned to a job to indicate its level of importance on the reader or output queues. The priority of each step of a job on the output queue is the same as the priority of that job on the reader queue, unless it is changed with a CHANGE priority command or specified separately on a PRINTER or PUNCH statement. Priority 0 is the same as priority 1 except that it indicates that the job is to remain on the reader queue until released by a RELEASE command. Priority 5 is the highest priority that can be assigned. Within a given priority, jobs are scheduled for execution on a first-in, first-out basis; that is, if two jobs have the same priority, the first one put on a queue is the first one removed. If this parameter is not specified, priority 1 is assumed.

PARTITION: This parameter specifies the partition in which a job is to be executed. If this parameter is not specified and input spooling is not used, the system assumes that the job can be executed in any partition. When input spooling is used, the job is scheduled for execution in the specified partition; if the partition is not specified, partition 1 is assumed. Specifying a PARTITION value of A, B, C, or D indicates the job can be run in any of the specified partitions, depending upon which partition is available at the time the job is scheduled to be run. The following are the partition assignments for the different parameters of the PARTITION operand:

	Partition Assignment
PARTITION-1	1
PARTITION-2	2
PARTITION-3	3
PARTITION-A	1 or 2
PARTITION-B	1 or 3
PARTITION-C	2 or 3
PARTITION-D	1 or 2 or 3

CORE: This parameter specifies the amount of main storage required to execute the largest step in a job. It ensures an adequate partition size for the duration of that job. The CORE parameter does not assign main storage for a partition; a SET command must be used for this purpose. When this parameter is specified, the partition must have the requested amount of main storage available before the job can be executed. When this parameter is not specified, the system assumes the job is scheduled in a partition with adequate main storage and uses the present partition size as the minimum for the job. CORE can be any multiple of 2 from 8 to 238 for 5704-SC1, or from 8 to 488 for 5704-SC2. If it is not a multiple of 2, it is rounded up to the next multiple of 2. A number less than 8 is treated as 8.

SPOOL: This parameter indicates whether or not spooling is to be active for the execution of the job. SPOOL-YES specifies that a job can be run with spooling active. If the parameter is not given, SPOOL-YES is assumed.

SPOOL-NO ensures that the job is not loaded if spooling is active in that partition. It is the user's responsibility to ensure that I/O devices for the job are available and spooling has been stopped for the partition. By using the STOP OCC command, spooling can be stopped for the partition, and the spooled reader, punch, or printer can be stopped. SPOOL-NO should be used when a job requires dedicated use of one or more devices being used by spooling.

QCOPY (5704-SC2 only): This parameter specifies whether or not the spool file copy program may copy this job from the reader queue. QCOPY-NO specifies that the job cannot be copied. QCOPY-YES specifies that the job may be copied. If the parameter is not specified, QCOPY-YES is assumed.

Considerations

When keyword parameters are not specified on this statement, comments cannot be given following the JOB statement identifier.

Consider the following examples:

```
//JOBA JOB CORE-8 comment
```

is valid.

```
//JOBB JOB comment
```

is not valid.

PRINTER

Functions

- Defines the system print device.
- Provides related information for a spooling environment.

Placement

The PRINTER statement must precede the RUN statement.

Format

// PRINTER parameters

Contents

The parameters are as follows:

- DEVICE-$\left\{\begin{matrix}\underline{1403}\\ 3284\end{matrix}\right\}$
- LINES-nnn
- FORMSNO-nnn
- COPIES-nn
- DEFER-$\left\{\begin{matrix}\underline{\text{YES}}\\ \text{NO}\end{matrix}\right\}$
- ALIGN-$\left\{\begin{matrix}\text{YES}\\ \underline{\text{NO}}\end{matrix}\right\}$
- PRIORITY-$\left\{\begin{matrix}0\\ 1\\ 2\\ 3\\ 4\\ 5\end{matrix}\right\}$
- QCOPY-$\left\{\begin{matrix}\underline{\text{YES}}\\ \text{NO}\end{matrix}\right\}$
- CLOSE-$\left\{\begin{matrix}\underline{\text{YES}}\\ \text{NO}\end{matrix}\right\}$

DEVICE: This optional parameter identifies the system print device. If it is omitted, 1403 is assumed. The 3284 printer cannot be spooled.

LINES: This optional parameter alters the number of print lines per page (forms length). The specified value can range from 12 to 112. The specified forms length remains in effect for that partition until another PRINTER statement with a LINES parameter is entered or until the next IPL. This parameter overrides the forms length specified during system generation. The forms length specified in a user program overrides the value specified in the LINES parameter. The forms length in a user program is in effect for the duration of that job step only (at the end of the step, the forms length is restored to the previous value).

FORMSNO: This optional parameter is used to inform the operator when to change forms in the printer and what forms to insert. Up to 3 characters can be used in any combination; commas, apostrophes, or blanks cannot be used. When this parameter is used, the print writer issues a message if the forms type for the current prınt step is different from the forms type of the previously printed step. The response taken to this message informs the print writer if separator pages should be printed between jobs and job steps. The parameter is effective only for the step in which the PRINTER statement was included.

COPIES: This optional parameter allows the user to obtain from 1 to 99 copies of a job step's spooled printed output. If the parameter is not specified, only one copy is printed. When multiple copies are requested, the print writer produces the number of requested copies before continuing to the next step on the print queue. This parameter is effective only for the step in which the PRINTER statement was included. The COPIES parameter is ignored when spooling is not active.

DEFER: This optional parameter can be used to defer the spooled printed output of a step until the step has completed execution. DEFER-NO allows the output of a step to begin printing before the job step has completed execution. DEFER-YES prevents printing until the step has completed execution. The DEFER parameter applies only to the job step in which the PRINTER statement was included. It is ignored when print spooling is not active. If the parameter is not specified, DEFER-YES is assumed.

ALIGN: This optional parameter is used to assist the operator with forms alignment. When ALIGN-YES is specified, the printer stops after printing the first line, and a message is displayed. The response to the message indicates that (1) the forms are aligned and printing should continue, or (2) the first line should be reprinted for realignment. If more than one copy of output is requested and ALIGN-YES is specified, the printer halts for forms alignment for each copy. If ALIGN-NO is specified, the printer does not stop. The ALIGN parameter applies only to the job step in which the PRINTER statement was included. It is ignored when print spooling is not active. If the parameter is not specified, ALIGN-NO is assumed.

PRIORITY: A priority can be assigned to a job step to indicate its order of printing from the print queue. Priority 0 is the same as priority 1 except that 0 indicates that the job step is to remain on the print queue until released by a RELEASE command. Priority 5 is the highest priority. Within a given priority, job steps are scheduled for printing on a first-in, first-out basis. The PRIORITY parameter applies only to the job step in which the PRINTER statement was included. PRIORITY is ignored when print spooling is not active. If the parameter is not specified, the priority of the job at the time of execution is used.

QCOPY (5704-SC2 only): This optional parameter can be used (QCOPY-NO) to prevent the spool file copy program ($QCOPY) from copying a job step that is on the print queue. QCOPY-YES allows $QCOPY to copy a job step. If the parameter is not specified, QCOPY-YES is assumed.

The QCOPY parameter applies only to the job step in which the PRINTER statement was included.

CLOSE (5704-SC2 only): This optional parameter can be used (CLOSE-NO) to group multiple print steps of a job as one job step under one stepname. If CLOSE-NO is specified, no further print steps of that job are closed at end of step, but they are closed at end of job. If CLOSE-YES is specified, a print step is closed at end of step. If the parameter is not specified, CLOSE-YES is assumed.

The specified CLOSE parameter remains in effect for that partition until another PRINTER statement with a CLOSE parameter is entered, or until the next end of job. The CLOSE parameter is ignored when print spooling is not active.

If a previous print step specified CLOSE-NO, a PRINTER statement with an ALIGN, COPIES, DEFER, FORMSNO, PRIORITY, or QCOPY parameter is ignored and causes an information message.

Considerations

When a PRINTER statement is encountered and printer output for the job step is being spooled, the effect of the COPIES, DEFER, ALIGN, CLOSE, and FORMSNO parameters is delayed until the print writer is ready to print the output.

If logging was assigned to the 1403, all OCL for the job step is included with the data. If multiple copies are requested (COPIES parameter), the OCL is included with each copy. Forms alignment is performed on the first OCL statement for that step. For these reasons, it may be desirable to suppress logging for those steps requiring multiple copies or forms alignment.

If printer forms must be aligned during the printing of a step's output (ALIGN-NO was not used), two methods can be used:

1. If the step's printed output is to be restarted from the beginning, the following steps can be taken:
   a. Enter a RESTART command to stop the print writer.
   b. Align the forms.
   c. Respond to the message PRT RESTART PENDING.

2. If it is not necessary to restart, the following steps can be taken:
   a. Enter a STOP PRT command to stop the print writer.
   b. Align the forms.
   c. Enter a START PRT command to resume printing.

The system print device is used to print system output and can be changed in a spooled job stream. However, if the new system print device is not the 1403, printed system output is not placed on the print queue; rather, it goes directly to the system print device. To resume spooling of system printed output, the system print device must be restored to the 1403.

PUNCH

During system generation, a system punch device is assigned to each partition. If the device is changed by a PUNCH statement, the new punch device remains in effect until changed by another PUNCH statement or until another IPL occurs.

Functions

- Changes the system punch device for the partition in which the statement was entered.
- Provides information related to a spooling environment.

Placement

The PUNCH statement must precede the RUN statement.

Format

// PUNCH parameters

Contents

The parameters are as follows:

- DEVICE-device
- CARDNO-nnn
- DEFER-$\left\{\begin{array}{c}\text{YES}\\ \hline \text{NO}\end{array}\right\}$
- COPIES-nn
- PRIORITY-$\left\{\begin{array}{c}0\\1\\2\\3\\4\\5\end{array}\right\}$
- QCOPY-$\left\{\begin{array}{c}\text{YES}\\ \hline \text{NO}\end{array}\right\}$

DEVICE: This optional parameter identifies the system punch device, which can be any of the following:

- MFCU1 – Primary hopper of the 5424
- MFCU2 – Secondary hopper of the 5424
- MFCM1 – Primary hopper of the 2560
- MFCM2 – Secondary hopper of the 2560
- 1442 – 1442 Card Read Punch
- 3741 – Directly attached 3741 Data Station or Programmable Work Station. The 3741 is not supported as a spool punch device.

If the DEVICE parameter is not used, the device remains unchanged.

CARDNO: This optional parameter is used to inform the operator when to change card types in the punch and what cards to insert. Up to 3 characters can be used in any combination; commas, apostrophes, or blanks cannot be used. When this parameter is used, the punch writer issues a message (CHANGE CARD TO TYPE nnn) if the card type for the current punch step is different than the card type of the previously punched step. The parameter is effective only for the step in which the PUNCH statement is included.

DEFER: This optional parameter can be used to defer the spooled punched output of a step until the step has completed execution. DEFER-NO allows the output of the step to begin punching before the job step has completed execution. DEFER-YES prevents punching until the step has completed execution. The DEFER parameter applies only to the job step in which the PUNCH statement was included. It is ignored when punch spooling is not active. If the parameter is not specified, DEFER-YES is assumed.

COPIES: This optional parameter allows the user to obtain from 1 to 99 copies of a job step's spooled punched output. If the parameter is not specified, only one copy is punched. When multiple copies are requested, the user must respond to the END OF PCH STEP message before the punch writer starts punching the next copy. Once started, the punch writer produces the number of requested copies before continuing to the next step on the punch queue. This parameter is effective only for the step in which the PUNCH statement is included. The COPIES parameter is ignored when punch spooling is not active.

PRIORITY: A priority can be assigned to a job step to indicate its order of punching from the punch queue. Priority 0 is the same as priority 1 except that 0 indicates that the job step is to remain on the punch queue until released by a RELEASE command. Priority 5 is the highest priority. Within a given priority, job steps are scheduled for punching on a first-in, first-out basis. The PRIORITY parameter applies only to the job step in which the PUNCH statement was included. PRIORITY is ignored when punch spooling is not active. If the parameter is not specified, the priority of the job at the time of execution is used.

QCOPY (5704-SC2 only): This optional parameter can be used (QCOPY-NO) to prevent the spool file copy program ($QCOPY) from copying a job step that is on the punch queue. QCOPY-YES allows $QCOPY to copy a job step. If the parameter is not specified, QCOPY-YES is assumed.

The QCOPY parameter applies only to the job step in which the PUNCH statement was included.

Considerations

When punching, the system does not verify that cards to be punched are blank.

When punching spooled output on a 1442 or MFCM, spool may insert blank cards between decks to cause all cards of the deck just punched to be fed into the correct stacker. On the MFCU, the last card is stacked without the extra blank cards.

The system punch device is used to punch system output and can be changed in a spooled job stream; however, if the new system punch device is not the spooled punch device, punched system output is not placed on the punch queue but goes directly to the new system punch device. To resume spooling of system punched output, the system punch device must be restored to the device designated as the spool punch device during system generation.

When a PUNCH statement is encountered as the step is being executed, only the DEVICE parameter takes effect. The effect of the COPIES, DEFER, and CARDNO parameters is delayed until the punch writer is ready to punch the output.

Consider the following example:

```
// PUNCH DEVICE-MFCU2,CARDNO-50
```

If the previous punch step had a different card type prior to punching the current step, the system displays CHANGE CARD TO TYPE 50, informing the operator to change card type to 50.

READER

During system generation, a system input device is assigned to each partition. If the device is changed by a READER statement, the new input device remains in effect until changed by another READER statement or until another IPL occurs.

Function

Changes the system input device for the partition in which the statement is entered.

Placement

The READER statement must precede a LOAD or CALL statement. The READER statement must not be placed between the LOAD and RUN or the CALL and RUN statements.

If the READER statement is used in a procedure, the system input device is changed when the READER statement is processed, but OCL statements are not read from the new system input device until the procedure is completely executed.

The system input device is used to read source statements, control statements, and OCL statements. Changing the system input device affects the placement of these statements.

Format

```
// READER code
```

Contents

code: The selected code identifies the system input device; it can be one of the following:

• CONSOLE	– 3277 Display Station (CRT/Keyboard)
• MFCU1	– Primary hopper of the 5424
• MFCU2	– Secondary hopper of the 5424
• MFCM1	– Primary hopper of the 2560
• MFCM2	– Secondary hopper of the 2560
• 1442	– 1442 Card Read Punch
• 2501	– 2501 Card Reader
• 3741	– Directly attached 3741 Data Station or Programmable Work Station

Considerations

The system input device can be changed in a spooled job stream; however, if the new system input device is different from the spooled input device, jobs are not scheduled for execution from the reader queue. To resume scheduling of jobs from the reader queue, the system input device must be restored to the device designated as the spooled reader during system generation.

Consider the following example, which illustrates how an optional step can be executed.

Assume that the following cards are in MFCU1, which is the spooled input device:

```
//JOB1 JOB
// CALL  PAYROL,R1
// RUN
// READER  MFCU2
// CALL  ANALYS, R1
// RUN
/.
/.
```

And assume that MFCU2 contains these cards:

```
// LOAD  SPECAL,R1
// RUN
// READER MFCU1
```

When the // READER MFCU2 statement is encountered in the input job stream, the system input device is changed to MFCU2. The system reads the next OCL statement from MFCU2. If it is not necessary to execute the program named SPECAL, then only the READER statement is needed in MFCU2.

/.

The /. statement is a job stream delimiter.

Functions

- Separates jobs, causes end of job, and prevents the reading of more OCL for the job.
- Indicates the end of the spooled input job stream if spooling is active and if two consecutive /. statements are used.
- Changes the partition from job mode to step mode, if spooling is not active, by causing end of job in the partition in which the statement is entered.

Placement

The placement of this statement is determined by how it is used in the job stream. The /. statement is not allowed in a procedure. Because a job cannot be executed as an inquiry program, the JOB and /. statements are ignored if a rollin is pending for that partition.

Format

/.

Content

None (comments may begin in position 3).

Considerations

A /. statement can be used to delimit jobs in a job stream. When a job being put into the reader queue is not terminated by a /. statement, the reader (input spooling routine) generates a /. statement as the last statement for the job in the queue; the /. statement is generated when the next JOB statement is encountered. This generated statement may contain extraneous characters following the /. that appear when the system history area is displayed or printed, or when the statement is logged on the system logging device.

Some programs (for example, $COPY) do not recognize a /. statement with comments or extraneous characters as end of file unless the statement is supplied by the system reader. Such programs should not attempt to read from the spool reader unless it is also the system reader.

The following examples show how the /. statement can be used.

1. Used as a delimiter between jobs:

```
//JOB1 JOB
//STEPA LOAD PROGA,R1
  ≀
// RUN
Data
/*
/&
//STEPB LOAD PROGB,R1
// RUN
Data
/*
/&
```

/. This indicates the end of the JOB1 and prevents the reading of more OCL for JOB1. If any job steps in JOB1 had been canceled, the /. indicates the end of the job. If this statement were not in the job stream, the following invalid JOB statement and the following LOAD and RUN statements would have been read as part of the OCL for JOB1:

```
//JOB2 KOB
//STEPA LOAD PROGC,R1
// RUN
```

2. Used to indicate the end ct spooled data.

```
//JOB1 JOB
//STEPA LOAD PROGA,R1
   ≀
// RUN
Data
/*
/&
//STEPB LOAD PROGB,R1
   ≀
// RUN
Data
/*
/.
//JOB2 JOB
//STEPA LOAD PROGC,R1
// RUN
Data
/*
/&
//STEPB LOAD PROGD,R1
   ≀
// RUN
Data
/*
/. ⎫
/. ⎭   Indicates end of spooled input.
```

3. Used to end job mode and return to step mode (nonspooled systems only).

```
//JOB1 JOB SPOOL-NO,PARTITION-1,CORE-12
//STEPA LOAD PROGA,R1
   ≀
// RUN
Data
/*
/&
//STEPB LOAD PROGB,R1
   ≀
// RUN
Data
/*
/.      Ends job mode.  Begin step mode.
// LOAD PROGC,R1
// RUN
```

The upper part of Figure 7-1 shows the processing of a System/3 Model 15 job stream in a partition without the spooling facilities. The first job, which needs a great deal of printing time, slows down throughput. The lower part shows the processing times of the same job stream with spooling. Here, the total time is divided into processing time, queue time (the time the output of the job is in the print queue, ready for printing), and printing time. (Input spooling and multiprogramming are not considered in this example.)

Although the time elapsed between the reading of a job and the completion of its output increases when using spooling, overall system performance is usually improved. This can be seen from the difference between points 2 and 3 when all five jobs have finished processing. In addition, the processing unit is available for processing because, between points 1 and 2, the only activity is the printing of the spooled print data which requires little processing unit time.

This is only a theoretical example of spooling performance in a single batch environment. The actual increase in throughput achieved depends, for example, on the I/O orientation of each program, the sequence of the particular jobs, and the speed and number of unit-record devices.

PERFORMANCE FACTORS

Performance of a system varies widely whether or not spooling is used. In some cases, there may be little or no advantage in spooling. In other cases, spooling provides substantial benefits to overall application throughput. Therefore, it is important to evaluate each application separately. It is very difficult, if not impossible, to predict the performance of a spooling system.

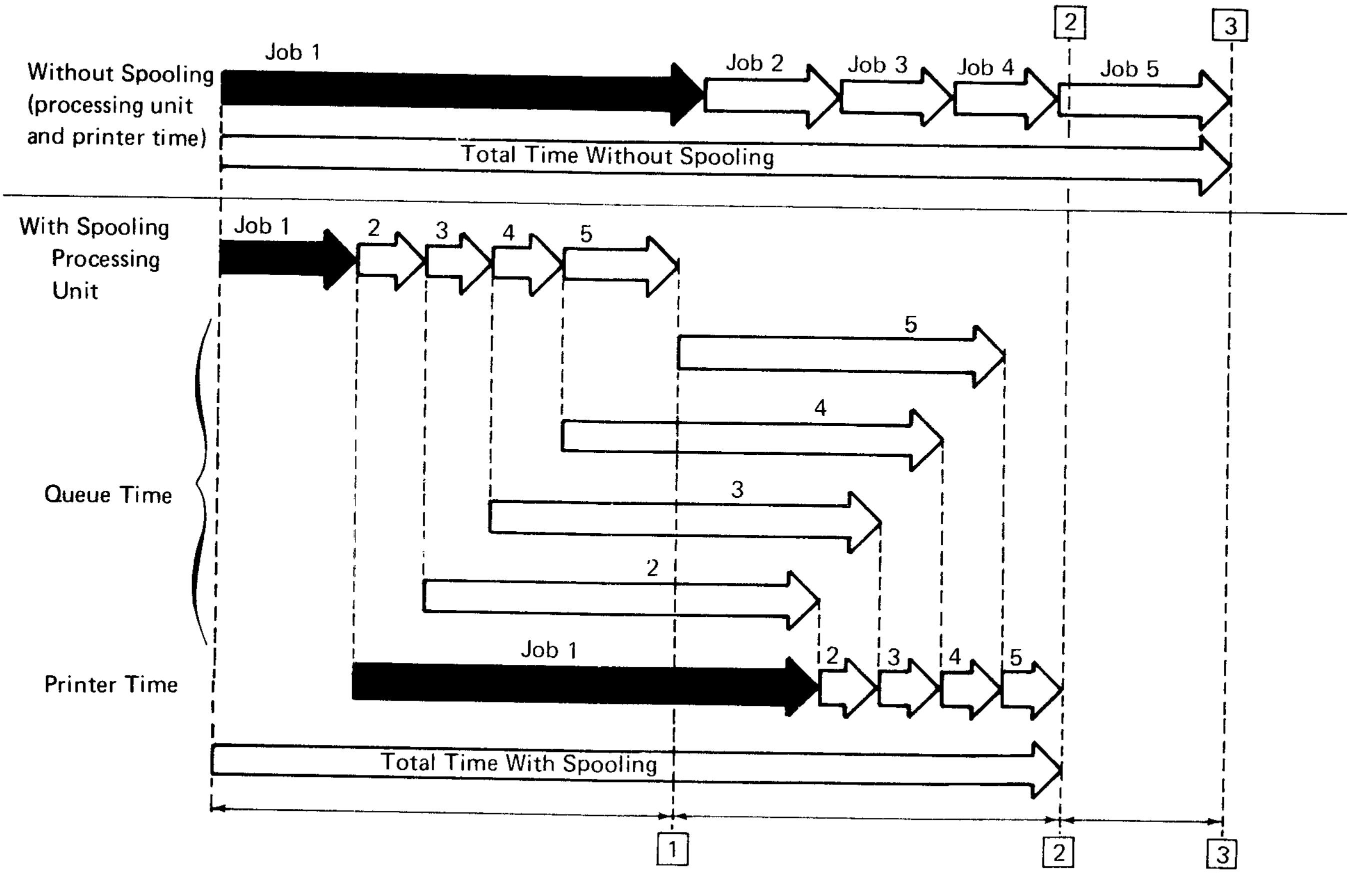
There are many factors that affect performance; when measuring the performance of a system, these factors should be considered. Consideration of these items can result in improved performance. (Specific spooling techniques are not included.)

System Considerations

- Speeds of I/O devices, for example:
  - The Model 10 has both a regular and a high-speed 5444 disk drive available; the Model 15 has only the high-speed 5444.
  - Printer performance is affected by the print chain/train that is used.
- Number of 5445/3340/3344 drives used; device contention can affect spool performance.

Application Considerations

- Job mix in each partition:
  - Single batch vs multiple batch
  - Single or dual batch and communications
  - Single or dual batch and inquiry
  - Unit-record I/O vs disk/tape I/O
- Relationship of steps within a job; dependency of one step on a preceding step.
- Volume of data:
  - Number of print lines and number of pages
  - Number of cards read
  - Number of cards punched and/or printed
  - Large volumes of unit record I/O tend to favor spooling
- Sequence and priority of jobs:
  - Job priority affects the sequence in which it is executed.
  - The first job should have a small amount of card input, so execution can begin while the remainder of the job stream is being read to disk.
- I ocation of files:
  - The spool file should be on a dedicated 5445/3340/3344, or on a 5445/3340/3344 with nonactive (backup) files.
  - Use of automatic work file allocation can increase run time and prevent optimum file location.
  - Files should be located to minimize access arm movement.



The improvement in system performance achieved by output queueing when spooling is shown by the difference in time between points 2 and 3. (Printing can begin prior to the completion of the program.)

Figure 7-1. Processing Five Jobs with and without Spooling

User Program Considerations

- File organization and access methods used:
  - Direct files may provide greater performance than indexed files.
  - Processing of indexed files requires more seeks (index and data).

- Use of data management options:
  - Double buffering
  - In-core index
  - Block size
  - External buffers (5704-SC2 only)
  - Disk file sharing (5704-SC2 only)

- Complexity of processing:
  - Number of unit-record I/O operations per logical program cycle
  - Balance of unit-record I/O and disk/tape/processing unit operations
  - Processing oriented; unit-record oriented; or disk/tape oriented
  - Use of program overlays

- Partition size:
  - Affects compilation time
  - Affects sorting
  - Affects the number of memory resident overlays in storage at one time

Operational Considerations

- Spooling options:
  - Level of spooling support (input/print/punch); affects main storage available to the partition(s).
  - Partitions spooled (1, 2, or 3); affects processing unit contention.
  - DEFER-NO; overlaps printed or punched output with program processing.

- Logging device:
  - Use of 3277 in one partition may tie up another partition.
  - Suggest not using 3277 to enter OCL or data.
  - If logging messages to a spooled printer, the OCL is spooled along with any printed data. If there is no data, OCL is spooled anyway requiring disk space and time.

- Halt/Nohalt: The halt or nohalt mode and severity level affect operator involvement.

- Operator familiarity with the job stream:
  - Can reduce contention situations.
  - Operator should be familiar with messages and action to be taken.

Chapter 8. Problem Situations

The following problem situations may be encountered when you use spooling.

SPOOL FILE IS FULL

When space is no longer available within the spool area on disk, the error message SPOOL FILE IS FULL appears on the CRT. If the print or punch queues contain job steps that are ready to be printed or punched or if the reader queue contains jobs waiting for execution, this space can be freed. The print and/or punch writers can be started or the jobs on the reader queue can be canceled in order to relinquish the disk space being occupied by the queues. Entering a REUSE command can make some space available before a writer completes the output for a print or punch step. When the operator responds to the SPOOL FILE IS FULL message, spool attempts to obtain space. If space is available, the partition continues to run; otherwise, the diagnostic message is issued again.

When the SPOOL FILE IS FULL message appears and all the queues are empty, the job must be rerun. If the size of the spool area is less than 199 cylinders for 5445, 166 cylinders for 3340 main data area, or 186 cylinders for a 3344 main data area, do the IPL again and start spool with a larger size for the spool area.

If it is not possible to increase the size of the spool area, you can remove steps from the print queue or punch queue by starting the appropriate writer. Also, jobs on any of the queues can be canceled in order to make space available. Entering a REUSE command can make some space available before a writer completes the output for a print or punch step. For future runs, using less input data can make space available for other jobs on the reader queue.

PERMANENT DISK ERRORS

If a significant performance degradation occurs on frequently run jobs, have your customer engineer check the error logging area to determine if temporary disk errors have been occurring on the disk drives containing the spooling area and the data files. When permanent disk errors occur, the operator is notified with a message. However, when temporary errors occur, the system can perform up to 256 retries before continuing. If temporary errors occur frequently, the time required to perform retries and to log these errors can cause a performance degradation. If changing data file locations and using a disk pack with no alternate tracks for the spooling area do not relieve the problem, contact your IBM representative.

If disk errors occur within the spool file, the reader or writer that was accessing the file at the time of the error is terminated when the message is responded to. For example, if the system supports input, print and punch spooling, and if a disk error occurs while the punch writer is accessing the spool file, the punch writer is terminated The reader and print writer may not be affected.

DEVICE FAILURES

The failure of a unit record device normally causes system operation to stop. However, if spooling is supported, applications can be executed even though the unit record device is not available. If the spool reader is unavailable, all the jobs that were placed in the reader queue can be executed. If the printer or punch is unavailable, the output for these devices will be saved in the print and punch queues until the device becomes available and the print or punch writer is initiated.

LOST JOBS OR JOB STEPS

The following conditions can cause jobs or steps to be lost:

- The operator IPLs the system during processing.
- A programming error causes a processor check, as indicated by the red processor check light.

A display of system status indicates that disk space has been lost if (1) all queues are empty, and (2) the total number of track groups does not equal the number of available track groups.

Entering a START SPOOL,NEW command at IPL is the only way to recover the lost disk space.

Chapter 9. Second 1403 Printer

If more printing capacity is needed for a System/3 Model 15D, a second 1403 printer can be attached and a second spool print writer can be selected during system generation. (An RPQ is required to attach a second 1403 printer.) The second print writer functions identically to the first print writer; anything in the print queue can be directed to the second printer.

No changes to any OCL statements or programs are required in order for spooling to print output on the second printer. However, no support is provided to use the second printer directly without spooling it.

The system generation autostart and autowrite options apply to both print writers. If autostart, for example, is selected for the printer during system generation, autostart applies to both print writers when spool is started after an IPL.

The only chain image that can be used by the second printer is the one specified for the system printer during system generation. For example, if the printed output from job A requires a special print chain or train, job A must be printed on the first printer because the chain image of the second printer cannot be changed. For more information on the chain image, see *Changing the Print Chain/Train* in Chapter 2.

OPERATOR CONTROL COMMANDS

The START, STOP, and RESTART operator control commands each have two additional operands available when the second printer is selected during system generation.

Operand PR1 (W1) has the same effect as PRT (W); either can be used to start, stop, or restart the first print writer.

Operand PR2 (W2) applies only to the second print writer.

Consider the following examples:

1. START PR2,CHK
   or
   S W2,CHK

   This command causes the second print writer to print only those job steps with a forms type of CHK.

2. STOP PR2,STEP
   or
   P W2,ES

   This command causes the second print writer to terminate at the end of the current step or copy. When the PR2 WRITER TERMINATED message is issued, the operator can start the second writer with a different forms type operand.

3. RESTART PR2,PAGE
   or
   T W2,PG

   This command causes the second print writer to restart printing at the beginning of the page on which the writer was terminated. If, for example, the second print writer was terminated on page 78 because of an I/O error for which a cancel option was taken, this command would cause the writer to restart printing at the beginning of page 78.

OPERATIONAL CONSIDERATIONS

Three methods can be used to control which printer will print which print steps:

- When both writers are started without a forms type operand, they print the spooled job steps on a first-in, first-out basis. (This is the method in effect when autostart is used for the print writers at IPL.) This method is effective i: most job steps have the same forms type.

- When one print writer is started with a forms type operand and the other without, the writer started without a forms type operand can print any forms type except the one specified for the other print writer. If approximately half of the total printing is on one forms type, this method can restrict all of the forms changes to one printer.

- When both writers are started with a forms type operand, each prints only the jobs steps with its specified forms type. Both writers can be started with the same forms type operand.

If a job step has multiple copies, all copies will be printed by one printer, even though the other printer is idle.

PRINT QUEUE DISPLAY

In the print queue display for a system with two print writers, an A appears before the priority number if the job step is currently being printed by the first print writer (PR1); a B if by the second print writer (PR2).

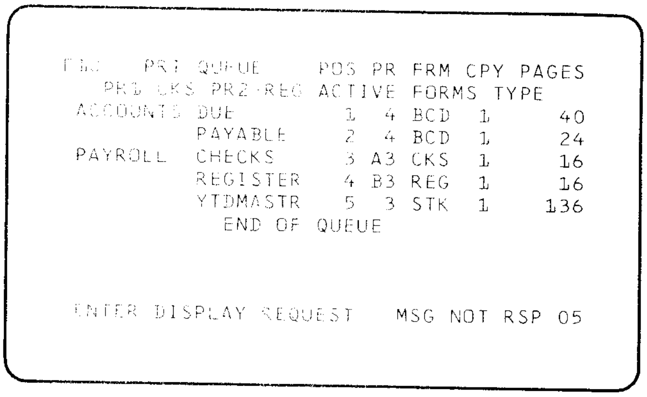
If one of the print writers was started using the forms type operand, the active forms type is displayed in one of the following messages:

PR1-xxx ACTIVE FORMS TYPE
PR2-xxx ACTIVE FORMS TYPE

If both writers were started using a forms type, the following message is displayed:

PR1-xxx PR2-xxx ACTIVE FORMS TYPE

The following is an example of a print queue display when both writers were started with a forms type operand:



```
[illegible]     PRT QUEUE      POS PR FRM CPY PAGES
   PR1-CKS PR2-REG ACTIVE FORMS TYPE
  ACCOUNTS DUE           1  4 BCD  1      40
           PAYABLE       2  4 BCD  1      24
  PAYROLL  CHECKS        3 A3 CKS  1      16
           REGISTER      4 B3 REG  1      16
           YTDMASTR      5  3 STK  1     136
             END OF QUEUE



  ENTER DISPLAY REQUEST    MSG NOT RSP 05
```

MAIN STORAGE REQUIREMENTS

The main storage requirement for the second print writer is 2.02K. For more information on main storage requirements, see Appendix B.

Appendix A. Format of the Queues

The spool area (named $SPOOL in the VTOC) contains up to three queues (a reader queue, print queue, and punch queue) depending on the options selected during system generation. The queues are not fixed in size, and can contain a variable number of entries. Any queue is not considered full until all available disk space within the spool area has been used.

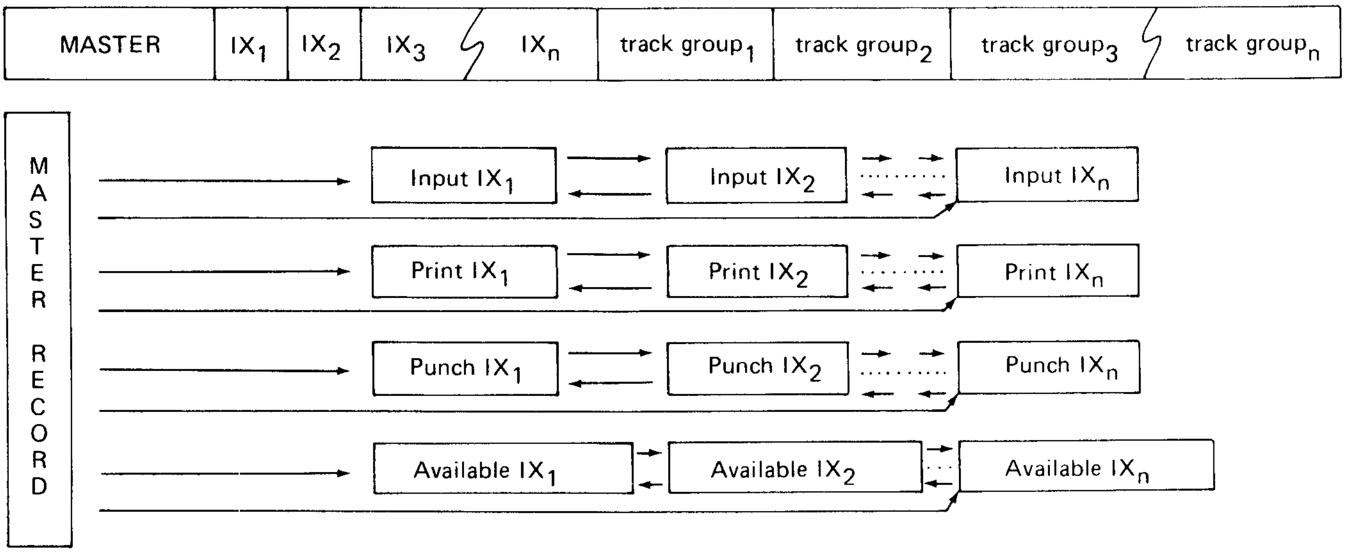
$SPOOL is allocated a permanent file from the highest available cylinder (5704-SC1), or from the lowest available cylinder (5704-SC2). For example, a 50-cylinder spool file on a newly initialized 5445 disk will be in cylinders 151-200.

The basic structure of the spool area is similar to a partitioned data set. The area is divided into track groups of 1, 2, 4, 5, or 10 tracks. An index within the area contains an entry for each track group. The first 256 bytes in the spool area is a master record that contains the pointers to all the indexes within any one queue or all the available indexes.

Each index contains the beginning and end disk addresses of a unique track group within the spool area. The number of spooled input, print, or punch records can vary within a track group. Spool records on disk can vary in length and are written in the following format:

K	Spooled record

Where K is 1 to 4 bytes of control information including the length of the record.



Appendix B. Main Storage Requirements

Figure B-1 will assist you in estimating the main storage requirements for various levels of spooling support. Additional requirements depend on the options you select during system generation. (For information on SCP programs, program products, and data management routines, see *IBM System/3 Model 15 System Generation Reference Manual,* GC21-7616.) Some of the options are:

- Support for the following devices:
  - 3284 Printer
  - 3410/3411 Magnetic Tape Subsystem
  - 3340 Direct Access Storage Facility
  - 3741 Data Station or Programmable Work Station
  - Second 1403 Printer

- Support for the following processing unit features:
  - Interval Timer
  - Binary Synchronous Communications Adapter (BSCA)
  - Binary Synchronous Communications Controller (BSCC)
  - Multiple Line Terminal Adapter (MLTA)
  - Serial I/O Channel (SIOC)
  - Unit Record Restart

- Support for the following functions:
  - I/O Storage Protection
  - Communications Control Program (CCP)
  - Number of partitions supported by spooling
  - Spooled reader device
  - Spooled punch device
  - Memory resident overlays

		LEVEL OF SPOOLING SUPPORT																			
		Print	Print/Punch			Input/Print					Input/Print/Punch										
Printer	1403	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Punch Device	MFCU		X								X	X	X	X							
	MFCM			X											X	X	X				
	1442				X													X	X	X	X
Input Device	MFCU					X					X							X			
	MFCM						X								X						
	1442							X				X							X		
	2501								X				X			X				X	
	3741									X				X			X				X
One Partition		7.11	11.01	11.79	10.41	10.91	10.95	10.41	10.44	10.90	12.81	12.81	12.84	12.81	13.63	13.62	13.59	12.71	12.21	12.24	12.71
Two Partitions		8.43	13.46	14.25	12.86	13.36	13.40	12.85	12.88	13.35	16.39	16.38	16.41	16.38	17.21	17.20	17.16	16.29	15.78	15.81	16.28
Three Partitions		9.75	15.91	16.70	15.31	15.80	15.84	15.30	15.33	15.79	19.96	19.95	19.99	19.95	20.79	20.77	20.74	19.86	19.35	19.38	19.85

Figure B-1. Estimated Main Storage Requirements of Spooling Support

Appendix C. Size of Spool Area

The following formulas can be used to estimate the amount of 5445/3340/3344 disk space (in cylinders) required for the the spool area. (102,400 is the number of bytes per cylinder on a 5445 disk, and 245,760 is the number of bytes per cylinder on a 3340/3344 disk.)

1. Spool requirement:

   Disk space = 1.1 (A + B + C) +2 cylinders

   where:

   A = Disk space to contain spool input records
   B = Disk space to contain spool print records
   C = Disk space to contain spool punch records

2. Spool input requirement:

   $$\underset{\text{(input disk space)}}{A} = \frac{N \times L}{S}$$

   where:

   A = Figure (rounded up to nearest integer) to be used in #1
   N = Maximum number of input records to be contained on disk at any one point in time
   L = Length of the input record (80 or 96) (use average length if input is from a 3741 except if put on the reader queue by $QCOPY)
   S = 102,400 for a 5445 or 245,760 for a 3340/3344

3. Spool print requirements:

   $$\underset{\text{(print disk space)}}{B} = \frac{N \times L}{S}$$

   where:

   B = Figure (rounded up to nearest integer) to be used in #1
   N = Maximum number of print records to be contained on disk at any one point in time
   L = Average length of print records
   S = 102,400 for a 5445 or 245,760 for a 3340/3344

4. Spool punch requirements:

   $$\underset{\text{(punch disk space)}}{C} = \frac{(P \times LP) + (I \times LI)}{S}$$

   where:

   C = Figure (rounded up to nearest integer) to be used in #1
   P = Maximum number of punch records to be contained on disk at any one point in time
   LP= Average length of the punch records
   I = Maximum number of punch records with printed data to be contained on disk at any one point in time
   LI = Average length of the printed data on those cards
   S = 102,400 for a 5445 or 245,760 for a 3340/3344

Example:

This example illustrates the use of the four formulas on a system with a 5445 disk. An installation wants the spool area to contain 4000 input records, 2 hours of printing, and 2000 punch records. The installation includes an MFCU and a 1403 Model 2 Printer (600 lpm) The MFCU is the spooled input and punch device. It has been estimated that the average print record length is 100 bytes, the average punch record is 80 bytes, and that half of the punched output contains printing with an average length of 75 bytes.

$$\underset{\text{(input)}}{A} = \frac{\text{4000 records x 96 bytes}}{\text{102,400 bytes/cylinder}} = \text{3.75, or 4 cylinders}$$

The next calculation converts the hours of print time to number of records.

2 hrs x 60 min/hr x 600 lines/min = 72,000 lines or records.

$$\underset{\text{(print)}}{B} = \frac{\text{72,000 records x 100 bytes}}{\text{102,400 bytes/cylinder}} = \text{70.3, or 71 cylinders}$$

$$\underset{\text{(punch)}}{C} = \frac{\text{(2000 x 80) +(1000 x 75)}}{\text{102,400 bytes/cylinder)}} = \text{2.29, or 3 cylinders}$$

Required disk space = 1.1 (4 + 71 + 3) + 2 = 87.80, or 88 cylinders.

Appendix D. Size of Track Groups

The spool area on disk is used for the reader, punch, and print queues. Space is allocated as required by the spooling routines; each time space is needed for a queue, another track group is assigned. The number of track groups within a given number of cylinders depends on the size of each track group. The size of each track group is determined by the user (1, 2, 4, 5, or 10 tracks).

Figure D-1 shows the number of track groups that will be produced in the spool file based on the size of the spool file and the track group size.

A small track group size reduces the amount of wasted space in the spool file. However, some performance may be sacrificed if the spooling routines must obtain space more frequently. Therefore, user application requirements and characteristics determine which track group size to use. In general, the following guide may be used.

If a large number of jobs is expected to be queued and disk space is small, the track group size should be small.

If a small number of jobs is expected to be queued, the track group size should be as large as possible.

If a mix is expected, a compromise between these two suggestions must be made.

Entries in the spool area begin on track group boundaries:

- Each job on the reader queue
- The printed output for each step (including the OCL for that step if the logging device was spooled during program execution)
- The punched output for each step

	5445				
ALLOCATED	TRACK GROUP SIZE				
CYLINDERS	1	2	4	5	10
10	195	98	49	39	19
20	390	197	99	79	39
30	585	296	149	119	59
40	780	395	198	159	79
50	975	493	248	190	99
60	1170	592	298	238	119
70	1365	691	347	278	139
80	1560	790	397	318	159
90	1755	888	447	358	179
100	1951	987	496	398	199
110	2146	1086	546	437	219
120	2341	1185	596	477	239
130	2536	1283	645	517	259
140	2731	1382	695	557	279
150	2926	1481	745	597	299
160	3121	1580	795	636	319
170	3317	1678	844	676	339
180	3512	1777	894	716	359
190	3707	1876	944	756	379

	3340/3344				
ALLOCATED	TRACK GROUP SIZE				
CYLINDERS	1	2	4	5	10
10	197	98	49	39	19
20	395	198	99	79	39
30	593	298	149	119	59
40	791	397	199	159	79
50	989	497	249	199	99
60	1187	596	299	239	119
70	1385	696	349	279	139
80	1583	795	398	319	159
90	1781	895	448	359	179
100	1979	994	498	399	199
110	2177	1094	548	439	219
120	2375	1193	598	478	239
130	2573	1293	648	518	259
140	2771	1392	698	558	279
150	2969	1492	748	598	299
160	3166	1591	797	638	319
170	3364	1691	847	678	339
180	3562	1790	897	718	359

Figure D-1. Number of Track Groups for Model 15 Spooling

Appendix E. Spool Messages

Message	Reason/Meaning
SPUTAD	The spool device is being used
SPUTAF	Align forms to line 1
SPUTAG	Align forms to line 1 of second printer (5704-SC2 only)
SPUTAP	A print restart is pending
SPUTBF	An error was found in the spool file queues (5704-SC2 only)
SPUTCA	Cannot accept CHANGE copies command (5704-SC2 only)
SPUTCA	Cannot accept HOLD command (5704-SC2 only)
SPUTCA	Cannot accept RESTART command
SPUTCC	Change card type
SPUTCF	Change form type
SPUTCH	Spool file disk error
SPUTCS	Cannot spool this job
SPUTDE	Spool file disk error
SPUTDF	The spool file contains data
SPUTEP	End of punch step
SPUTFA	Spool file attributes at spool start (5704-SC2 only)
SPUTFF	The spool file is full
SPUTFZ	Form size change required
SPUTIE	Disk error on spool file index
SPUTIF	An invalid spool file was found
SPUTJE	The JOB statement is in error
SPUTJQ	Reader queue job started
SPUTNA	The requested queue is not supported
SPUTNF	Jobname or stepname was not found
SPUTNI	No input jobs are available
SPUTNP	NPRO the punch device
SPUTNR	Program input device is the spool reader
SPUTNS	Not enough disk space
SPUTPR	A punch restart is pending
SPUTPT	The punch writer is terminated
SPUTPW	The print writer is terminated (on page nnnnn)
SPUTQE	The print/punch queue is empty/held
SPUTRE	The reader queue is empty/held
SPUTRH	The print restart is halted
SPUTRP	Ready the punch device
SPUTRQ	Job/step/queue was affected by a command
SPUTRQ	nnnn track groups available
SPUTRT	The reader is terminated
SPUTSA	Spool is active
SPUTSF	The stacker is full
SPUTSH	Spool is halted
SPUTTM	Start date and time, and end time
SPUTWA	The reader/writer is active

For a detailed description of these messages, see *IBM System/3 Model 15 System Messages,* GC21-5076.

Appendix F. Operator Control Commands (OCC) Summary

INFORMATION ABOUT SYNTAX ILLUSTRATIONS

- Uppercase letters, numbers, and commas must be entered as shown. Parentheses ( ) indicate an abbreviation for the item above. Do not enter brackets, braces, or parentheses.

- Lowercase items represent variables.

- Items or groups of items within brackets [ ] are optional —enter one or none.

- Items in braces { } are alternatives—enter one.

- Optional items that are underlined indicate the default value. If none of the items are entered, the default value is used.

- All operands are positional. Operands must be entered in the order shown. If any operand (except the last) is omitted, a comma must be entered in its place.

- There must be at least one blank between the command and the operands. The first blank after the start of the operands terminates the operands. There must be only blanks after the last operand.

OCC LIST

Operator control commands (OCC) allow the operator to communicate with the sytem. All OCC are entered from the CRT/keyboard. By using OCC, the operator can communicate with the system before, during, or after job execution. The following is a complete list of OCC. For a complete description of OCC, see the *IBM System/3 Model 15 Operator's Guide,* GC21-5075.

Command	Operand	Function
CANCEL (CN)	SPOOL (SP)	Cancels spooling support (at IPL only).
	RDRQ, { jobname } (RQ) { characters** } PRTQ, { jobname } [,stepname] (WQ) { characters** } [,characters**] PCHQ, { jobname } [,stepname] (PQ) { characters** } [,characters**]	Cancels a job or a job step, or multiple jobs or job steps. The characters** operand is for 5704-SC2 only.
	*RDRQ (*RQ) *PRTQ (*WQ) *PCHQ (*PQ)	Cancels an entire queue.
	P1 [,2 ,3] P2 [,2 ,3] P3 [,2 ,3]	Cancels program execution in a partition. The 2 or 3 following the P1, P2, or P3 indicates the option of the cancel. P3 operand is for 5704-SC2 only.

Command	Operand	Function

CHANGE
(G)
	CORE,size [RDRQ], {jobname }
	[(RQ)] {characters**}
		Changes the main storage requirements, number of copies, card type, forms type, partition assignment, or priority of a job or job step or multiple jobs or job steps.

	CPY,[number], {PRTQ }
	{(WQ) }, {jobname } [,stepname]
	{PCHQ } {characters**} [,characters**]
	{(PQ) }
		The characters** operand is for 5704-SC2 only.

	CRD,card type, [PCHQ], {jobname } [,stepname] [(PQ)] {characters**} [,characters**]	
	FRM,forms type, [PRTQ], {jobname } [,stepname] [(WQ)] {characters**} [,characters**]	
	PTN, [1 2 3 A B C D], [RDRQ], {jobname } [(RQ)] {characters**}	
	PTY, {1 (Y) 2 3 4 5}, {RDRQ, {jobname } (RQ) {characters**} PRTQ, {jobname } [,stepname] (WQ) {characters**} [,characters**] PCHQ, {jobname } [,stepname] (PQ) {characters**} [,characters**]}	
DATE (DT)	{mm/dd/yy mmddyy dd/mm/yy ddmmyy}	Sets the system partition dates.

Command	Operand	Function
DISPLAY (D)	STATUS [,P1 ,P2 ,P3 ,N]	Displays the system status, partition status, and area names. Pn and N operands are for 5704-SC2 only.
	HISTORY [,task] [,task] . . . (H)	Displays the history either of the entire system or of one to seven tasks. Task operands[1] are for 5704-SC2 only.
	SVAID (SV)	Displays service aids options.
	CORE	Displays main storage.
	RDRQ (RQ)	Displays the contents of the spooling queues.
	PRTQ (WQ)	
	PCHQ (PQ)	
DUMP (K)	P1 [,2 ,3] P2 [,2 ,3] P3 [,2 ,3] SYSTEM [,DISK] (S)	Causes a dump of any partition or all of main storage to a printer; or dumps main storage, SWA, saved transient area, part of SHA, configuration records, PTF log sectors, microcode levels, and error recording data to $SYSDUMP file. P3 and DISK operands are for 5704-SC2 only.

[1] Possible task operands are P1, P2, P3, S, SP, OCC, CCP, CCS, CCU, C44, C55, C66, C77, C88, C99, CCC, CDD, CEE, CGG, CHH, CMM, CPP, CTT, CUU, CVV, CWW, CXX, CYY, CZZ.

Command	Operand	Function
HALT (HT)	[P1 P2 P3 SHA ,CCP ,tracks]	Causes the partition to halt with an EJ or ES message after the current job step is complete. Causes the system to halt or $HACCP to be automatically invoked under CCP before unprinted system history area entries are overlaid. CCP and tracks operands are for 5704-SC2 only.
HIPTY (HP)	high-priority	Limits the priority of jobs placed on the active spool file reader queue by $QCOPY under CCP (5704-SC2 only).
HOLD (H)	{ RDRQ (RQ) [,jobname ,characters**] PRTQ (WQ) [,jobname ,characters** [,stepname ,characters**]] PCHQ (PQ) [,jobname ,characters** [,stepname ,characters**]] }	Prevents a job, a job step, multiple jobs or job steps, or an entire queue from being scheduled for input or output. The characters** operand is for 5704-SC2 only.
IDELETE (I)		Causes the system to automatically delete I-type (informational) messages to which there was no response (when there should have been a response.)

Command	Operand	Function
KEEP (KP)	RDRQ, { jobname } (RQ) { characters** } PRTQ, { jobname } [,stepname] (WQ) { characters** } [,characters**] PCHQ, { jobname } [,stepname] (PQ) { characters** } [,characters**]	Keeps single or multiple printed, punched, or executed jobs and/or job steps on the queue (5704-SC2 only).
NOHALT (NHT)	[<u>P1</u> P2 P3 SHA]	Changes the processing mode of a partition from halt to nohalt or allows system history area entries to be overlaid without operator notification.
NOIDELETE (NI)	–	Causes the system to cease automatic deletion of I-type messages that received no response.
PTY (Y)	{ SPOOL (SP) P1 } { , } ... P2 } { = } P3 }	Changes the priority for the system tasks: spooling (reader, print writer, punch writer), partition 1, partition 2, and partition 3.
READER (RDR)	[<u>P1</u> P2] ,device P3]	Changes the system input device for a partition.
RELEASE (R)	RDRQ [,jobname] (RQ) [,characters**] PRTQ [,jobname [,stepname]] (WQ) [,characters** [,characters**]] PCHQ [,jobname [,stepname]] (PQ) [,characters** [,characters**]]	Releases a job or job step, multiple jobs or job steps, or an entire queue from the hold state. The characters** operand is for 5704-SC2 only.
RESTART (T)	[<u>PRT</u> [,PAGE] (W) (PG) [,number] PR1 [,PAGE] (W1) (PG) [,number] PR2 [,PAGE] (W2) (PG) [,number] PCH (P)]	Restarts a job step's printed or punched output. PR1 and PR2 operands are for 5704-SC2 only.

REUSE
(RU)
	{ PRTQ ,jobname[,stepname] }
	(WQ)
	{ PCHQ ,jobname[,stepname] }
	(PQ)
		Frees space in the spool file from the currently printing or punching job step (5704-SC2 only).

Command	Operand	Function
ROLLOUT (RO)	–	Interrupts a rollout-evoking program to allow another program to be loaded and executed (5704-SC1 only).
SET	[P1 P2 P3 SH] ,size	Overrides the partition and file share area size specified during system generation.
SIMULATE (SM)	{ON OFF}	Enables or disables 5444 simulation on 3340 drive 2 (5704-SC1 only).
START (S)	SPOOL [,P1 (SP) ,P2 ,P3 [,NEW] [,unit] [,size] [,track-group-size]]	Restarts spooling in the designated partition. Starts spooling at IPL.
	RDR [,ROLL] (R)	Starts the spooled reader, print writer(s), or punch writer.
	PRT[,forms type] (W)	
	PR1[,forms type] (W1)	PR1 and PR2 operands are for 5704-SC2 only.
	PR2[,forms type] (W2)	
	PCH (P)	
	P1 P2 P3	Restarts program execution in a partition (5704-SC2 only).

Command	Operand	Function
STOP (P)	SPOOL (SP) {,P1 ,P2 ,P3}	Terminates spooling in the designated partition.
	RDR (R)	Stops the spooled reader, print writer(s), or punch writer.
	PRT (W) [,STEP (ES) ,PAGE (PG)]	
	PR1 (W1) [,STEP (ES) ,PAGE (PG)]	PR1 and PR2 operands are for 5704-SC2 only.
	PR2 (W2) [,STEP (ES) ,PAGE (PG)]	
	PCH (P) [,STEP (ES)]	
	P1 P2 P3	Stops program execution in a partition (5704-SC2 only).
TIME (TM)	{hh/mm/ss hhmmss}	Sets time-of-day clock.
TLOG (TL)	{ON OFF}	Enables or disables the currently loaded transaction logging routine (5704-SC2 only).
TRACE (E)	{ON OFF} ,SYSTEM (S)	Enables or disables the system trace routine (which must be currently loaded).

This glossary contains terms used in this manual. If you do not find the term you are looking for, refer to *IBM System/3 Model 15 Introduction,* GC21-5094.

IBM is grateful to the American National Standards Institute (ANSI) for permission to reprint its definitions from the *American National Standard Vocabulary for Information Processing* (Copyright 1970 by American National Standards Institute, Incorporated), which was prepared by Subcommittee X3K5 on the Terminology and Glossary of the American National Standards Committee X3.

An asterisk (*) preceding a definition in the list below indicates an American National Standard definition.

activate: To make spooling support ready to operate; spool becomes inactive once a CANCEL SPOOL or STOP SPOOL command is issued.

application program: A program written for or by a user that applies to the user's own work.

batch processing: *(1) Pertaining to the technique of executing a set of computer programs such that each is completed before the next program of the set is started. (2) Pertaining to the sequential input of computer programs or data.

checkpoint: (1)* A place in a routine where a check, or a recording of data for restart purposes, is performed. (2) A point at which information about the status of a job and the system can be recorded so that the job step can be later restarted. (3) To record such information.

checkpoint/restart facility: A facility for restarting execution of a program at some point, other than at the beginning, after the program was terminated due to a program or system failure. A restart begins at a checkpoint of a job step, and uses checkpoint records to reinitialize the system.

cold start: Same as *initial program load.*

combined file: A file of cards upon which both input and output operations are performed. All cards in the file are read, but not all cards in the file are punched, interpreted and/or stacker selected.

configuration: The group of machines, devices, and programs that make up a data processing system.

console: That part of a computer used for communication between the operator and the computer.

contention: (1) A condition on a communication channel when two or more locations try to transmit at the same time. (2) Unregulated bidding for a resource by multiple users.

CRT: Cathode-ray tube. A display device on which images are produced on a cathode ray tube.

data: * A representation of facts, concepts, or instructions in a formalized manner suitable for communication, interpretation, or processing by humans or automatic means.

data file: A collection of related data records organized in a specific manner. For example a payroll file (one record for each employee, showing the rate of pay, deductions, etc) or an inventory file (one record for each inventory item, showing the cost, selling price, number of stock, etc).

execute: To carry out an instruction or group of instructions, as in a program.

initial program load: The initialization procedure that causes an operating system to commence operation. Abbreviated IPL.

input: * Pertaining to a device, process, or channel involved in the insertion of data or states, or to the data or states involved.

input/output: (1)* Pertaining to either input or output, or both. (2) A general term for the equipment used to communicate with a computer, commonly called I/O. (3) The data involved in such communication. (4) The media carrying the data for input/output.

input queue: A queue (waiting list) of job definitions in direct access storage arranged in order of assigned priority.

installation: A particular computing system, in terms of the work it does and the people who manage it, operate it, apply it to problems, service it, and use the results it produces.

job: (1)* A specified group of tasks prescribed as a unit of work for a computer. By extension, a job usually includes all necessary computer programs, linkages, files, and instructions to the operating system. (2) A collection of related problem programs, identified in the job stream by a JOB statement.

job step: A unit of work associated with one processing program or one cataloged procedure and related data. A job consists of one or more job steps.

job stream: The sequence of operation control statements and data submitted to an operating system on an input unit especially activated for this purpose by the operator.

main storage: The general purpose storage of a computer.

MFCM: Multi-Function Card Machine. Same as IBM 2560 Multi-Function Card Machine.

MFCU: Multi-Function Card Unit. Same as IBM 5424 Multi-Function Card Unit.

operating system: * Software which controls the execution of computer programs and which may provide scheduling, debugging, input/output control, accounting, compilation, storage assignment, data management, and related services.

operation control language: A programming language used to code operation control statements. Abbreviated OCL.

operation control statement: A statement in a job or job step that is used in identifying the job or describing its requirements to the operating system.

operator control command: A statement to the supervisor issued via the console device, that causes the supervisor to provide requested information, alter normal operations, initiate new operations, or terminate existing operations. Abbreviated OCC.

output: * Pertaining to a device, process, or channel involved in an output process, or to the data or states involved.

output queue: A queue of control information describing system output that specifies to an output writer the location and disposition of system output.

partition: A subdivision of main storage.

priority: A rank assigned to a job that determines its precedence in receiving system resources.

queue: (1) A waiting line or list formed by items in a system waiting for service; for example, jobs to be performed. (2) To arrange in, or form, a queue.

restart: The process of resuming a job at a checkpoint within the job step that caused abnormal termination. See checkpoint/restart facility.

rollout/rollin: An optional feature of the operating system that allows the temporary reassignment of a main storage partition from one program to another.

spooling: The reading and writing of input and output streams on disk concurrently with job execution, in a format convenient for later processing or output operations.

step: (1) * One operation in a computer routine. (2) See *job step.*

storage: * (1) Pertaining to a device into which data can be entered, in which it can be held, and from which it can be retrieved at a later time. (2) Loosely, any device that can store data. (3) Synonymous with memory.

system generation: The process of tailoring the system control programming to suit a user's requirements and including the desired program products.

system input device: A device specified as a source of input.

system history area: A space on a direct access storage device in which information relating to system activity is stored.

system output device: A device assigned to record output data.

throughput: The total volume of work performed by a computing system over a given period of time.

unit-record devices: Card readers, card punches, and printers.

Index

International Business Machines Corporation

General Systems Division
4111 Northside Parkway N.W.
P.O. Box 2150
Atlanta, Georgia 30301
(U.S.A. only)

General Business Group/International
44 South Broadway
White Plains, New York 10601
U.S.A.
(International)

IBM System/3 Model 15 User's Guide to Spooling (File No. S3-30) Printed in U.S.A. GC21-7632-5

GC21-7632-5