



Horst.

PUBLICATIONS UPDATE

Operating System/3 (OS/3)

Spooling and Job Accounting Concepts and Facilities

UP-9975-A

This Library Memo announces the release and availability of Update A to "SPERRY® Operating System/3 (OS/3) Spooling and Job Accounting Concepts and Facilities", UP-9975.

This manual describes the spooling and job accounting software component of SPERRY Operating System/3 (OS/3) for System 80 users.

This update documents the following new spooling and job accounting features for the 10.0 release:

- The BEGIN SPL command now calls the RP output writer when a spool file destined for a workstation auxiliary printer is released from the held state. There is also a new parameter that lets you specify whether an output writer should be called.
- A new message included with the DELETE SPL command questions whether you want to delete the subfile queue specified in the command keyin.
- The DISPLAY SPL, STATUS command now displays the percentage of spool file space that is still available.
- The BEGIN SPL, DELETE SPL, HOLD SPL, PR, PU, PD, and RP commands now let you use the binary job number as a spool file command modifier.
- The DISPLAY SPL and DISPLAY ACT commands now display the binary job number.

All other changes are corrections or expanded descriptions applicable to features present in spooling and job accounting prior to the 10.0 release.

Copies of Update A are now available. You can order the update only or the complete manual with the update through your local Sperry representative. To receive only the update, order UP-9975-A. To receive the complete manual, order UP-9975.

LIBRARY MEMO ONLY

LIBRARY MEMO AND ATTACHMENTS

THIS SHEET IS

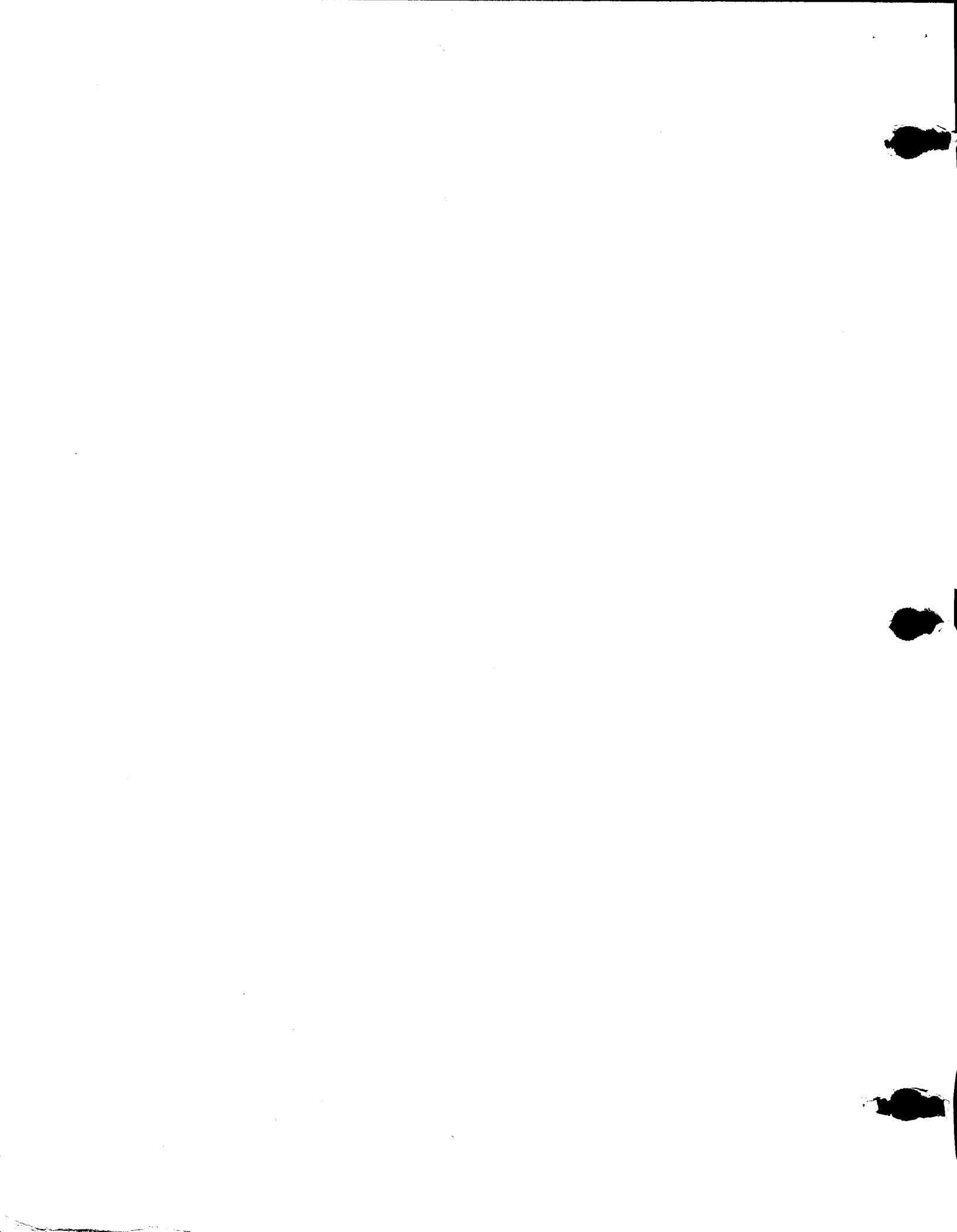
Mailing Lists
BZ, CZ, and MZ

Mailing Lists B00, B02, MBW,
28U, and 29U
(Package A to UP-9975,
36 pages plus Memo)

Library Memo for
UP-9975-A

RELEASE DATE:

May 1986



UNISYS

OS/3

**Spooling and
Job Accounting**

**Operating
Guide**

Relative to Release
Level 11.0

Priced Item

August 1987

Printed in U S America
UP-9975



UNISYS

OS/3

**Spooling and
Job Accounting**

**Operating
Guide**

Copyright© 1987 Unisys Corporation
All Rights Reserved
Unisys is a trademark of Unisys Corporation.
Previous Title: OS/3 Spooling and Job Accounting
Concepts and Facilities

Relative to Release
Level 11.0

Priced Item

August 1987

Printed in U S America
UP-9975

NO WARRANTIES OF ANY NATURE ARE EXTENDED BY THE DOCUMENT. Any product and related material disclosed herein are only furnished pursuant and subject to the terms and conditions of a duly executed Program Product License or Agreement to purchase or lease equipment. The only warranties made by Unisys, if any, with respect to the products described in this document are set forth in such License or Agreement. Unisys cannot accept any financial or other responsibility that may be the result of your use of the information in this document or software material, including direct, indirect, special or consequential damages.

You should be very careful to ensure that the use of this information and/or software material complies with the laws, rules, and regulations of the jurisdictions with respect to which it is used.

The information contained herein is subject to change without notice. Revisions may be issued to advise of such changes and/or additions.

FASTRAND, ✦SPERRY, SPERRY✦UNIVAC, SPERRY, SPERRY UNIVAC, UNISCOPE, UNISERVO, UNIS, UNIVAC, and ✦ are registered trademarks of Unisys Corporation. ESCORT, PAGEWRITER, PIXIE, PC/IT, PC/HT, PC/microIT, SPERRYLINK, and USERNET are additional trademarks of Unisys Corporation. MAPPER is a registered trademark and service mark of Unisys Corporation. CUSTOMCARE is a service mark of Unisys Corporation.



Preface

This manual is one of a series designed to instruct and guide the system administrator in the use of the SPERRY Operating System/3 (OS/3). This manual specifically describes the spooling system, its effective and automatic use, and how you can control it. Our intended audience is the system administrator with a basic knowledge of data processing, but with little spooling experience, and the programmer and operator whose experience is on nonspooling systems.

This manual:

- Presents basic spooling concepts
- Describes input and output spooling functions, including remote spooling capabilities and general spooling controls
- Explains how you initialize or recover the spool file
- Describes how you produce a job accounting report
- Summarizes the OS/3 spooling capabilities and controls

As you read through the manual, you'll notice our description of each spooling facility includes its associated job control statement and console or workstation command. These basic statements and commands are summarized in Table 9-1 to provide you with a quick reference to the various ways you control spooling within your system. You can easily locate the specific format and parameters for each console command in the operator's spooling reference (Appendix C). It's intended that you remove this appendix from this manual and insert it in your operations handbook so that it is handy for your operator.

Appendix D provides you with a spooling summary that may be used in a similar manner to locate specific information for all spooling control options. This summary includes the spooling console command formats and parameters in brief reference form, as well as similar spooling information for your system generation options, job control statements, and workstation commands.

In addition to the operator's spooling reference and the spooling summary, the appendixes contain the output writer formats for a tape, disk, and format label diskette.

Other current OS/3 publications referenced in this manual and useful to the spooling user are:

Operations handbook operator reference, UP-8859

Describes the commands and operating procedure for the console/workstation operator

Sort/merge user guide/programmer reference, UP-8819

Describes the independent routine used to sort records and merge files

ICAM utilities user guide, UP-8552

Describes the set of utilities provided by ICAM, including the facility to submit batch jobs from a remote terminal

Distributed data processing concepts and facilities, UP-8811

Describes distributed data processing, its ICAM networks, and the commands and procedures for using it

General editor (EDT) user guide/programmer reference, UP-9976

Describes the functions and use of the general editor and its command language

Interactive services commands and facilities user guide/programmer reference, UP-9972

Describes the interactive services commands

Contents

PAGE STATUS SUMMARY

PREFACE

CONTENTS

1. BASIC SPOOLING AND JOB ACCOUNTING CONCEPTS

| | | |
|------|-------------------------|-----|
| 1.1. | INTRODUCTION | 1-1 |
| 1.2. | WHAT IS SPOOLING? | 1-1 |
| 1.3. | WHY USE SPOOLING? | 1-2 |
| 1.4. | WHAT IS JOB ACCOUNTING? | 1-5 |

2. SPOOLING ELEMENTS

| | | |
|----------|---|-----|
| 2.1. | BASIC ELEMENTS OF SPOOLING | 2-1 |
| 2.1.1. | The Input Reader | 2-3 |
| 2.1.2. | The Spooler and the Spool File (SYSPPOOL) | 2-4 |
| 2.1.3. | The Output Writer | 2-5 |
| 2.2. | ESTABLISHING YOUR SPOOLING ENVIRONMENT | 2-7 |
| 2.2.1. | Spooling Simplified | 2-7 |
| 2.2.1.1. | Before Your Job Is Run | 2-8 |
| 2.2.1.2. | After Your Job Run Begins | 2-8 |

3. INPUT SPOOLING FUNCTIONS

| | | |
|--------|--------------------------|-----|
| 3.1. | BASIC INPUT FUNCTIONS | 3-1 |
| 3.1.1. | Multiple Input Readers | 3-2 |
| 3.1.2. | Nonresident Symbiont | 3-2 |
| 3.1.3. | Input File Retention | 3-2 |
| 3.1.4. | Automatic Job Scheduling | 3-2 |

| | | |
|------------|---|------|
| 3.2. | INPUT FROM PUNCHED CARDS | 3-3 |
| 3.2.1. | Punched Cards Containing Different Column Totals | 3-3 |
| 3.2.2. | Multiple Input Files in a Card Deck | 3-3 |
| 3.3. | INPUT FROM TAPE | 3-3 |
| 3.3.1. | Multiple Input Files on Tape | 3-3 |
| 3.4. | INPUT FROM DISKETTE | 3-3 |
| | | |
| 4. | OUTPUT SPOOLING FUNCTIONS | |
| | | |
| 4.1. | BASIC OUTPUT FUNCTIONS | 4-1 |
| 4.1.1. | Multiple Output Writers | 4-2 |
| 4.1.2. | Nonresident Symbiont | 4-3 |
| 4.1.3. | Output File Retention | 4-3 |
| 4.1.4. | Nonburst and Burst Mode Operation | 4-3 |
| 4.1.4.1. | The Output Writer in Nonburst Mode | 4-4 |
| 4.1.4.2. | The Output Writer in Burst Mode | 4-6 |
| 4.1.4.3. | Using Nonburst and Burst Modes | 4-8 |
| 4.1.4.3.1. | Breakpointing Printer and Punch Files | 4-8 |
| 4.2. | OUTPUT TO PRINTER AND CARD PUNCH | 4-13 |
| 4.2.1. | Holding Files from Output Writer Processing | 4-13 |
| 4.2.2. | Headers to Separate Files | 4-13 |
| 4.2.3. | Printing the Job Log | 4-14 |
| 4.2.4. | Test Lines | 4-15 |
| 4.2.5. | Printing the Console Log | 4-15 |
| 4.2.6. | Printing the Workstation Log | 4-16 |
| 4.2.7. | Multiple Copies of a File | 4-17 |
| 4.2.8. | Device Swapping | 4-17 |
| 4.3. | OUTPUT TO WORKSTATION AUXILIARY PRINTER | 4-17 |
| 4.4. | OUTPUT TO DISKETTE: DATA SET LABEL MODE | 4-21 |
| 4.5. | OUTPUT TO TAPE, DISK, AND FORMAT LABEL DISKETTE: REDIRECTED OUTPUT | 4-21 |
| 4.6. | ACCUMULATING AND TRANSFERRING LOGS | 4-22 |
| 4.6.1. | Controlling How Logs Are Processed in Your System | 4-23 |
| 4.6.2. | Transferring the Accumulated Job and Workstation Log for Future Processing | 4-28 |
| 4.6.3. | Transferring the Accumulated Console Log for Future Processing | 4-29 |
| | | |
| 5. | GENERAL SPOOLING CONTROL | |
| | | |
| 5.1. | OVERVIEW | 5-1 |
| 5.2. | DISPLAYING THE STATUS OF SPOOLED FILES | 5-1 |
| 5.3. | HOLDING SPOOLED FILES | 5-2 |
| 5.4. | RELEASING SPOOLED FILES | 5-3 |
| 5.5. | DELETING SPOOLED FILES | 5-4 |

6. REMOTE SPOOLING

- | | | |
|------|---|-----|
| 6.1. | OVERVIEW | 6-1 |
| 6.2. | REMOTE SPOOLING USING THE REMOTE BATCH PROCESSOR | 6-2 |
| 6.3. | REMOTE SPOOLING USING DISTRIBUTED DATA PROCESSING | 6-5 |

7. SPOOLING INITIALIZATION AND FILE RECOVERY

- | | | |
|------|------------|-----|
| 7.1. | OVERVIEW | 7-1 |
| 7.2. | COLD START | 7-2 |
| 7.3. | WARM START | 7-2 |

8. PRODUCING JOB ACCOUNTING REPORTS

- | | | |
|------------|--|------|
| 8.1. | SYSTEM LOG ACCUMULATION PROGRAM (SL\$LOG) | 8-1 |
| 8.1.1. | SL\$LOG Output Files | 8-1 |
| 8.1.2. | SL\$LOG Job and Workstation Log Output | 8-2 |
| 8.1.2.1. | Accounting Record Formats | 8-2 |
| 8.1.2.1.1. | Accounting Data (Bytes 0-130) | 8-2 |
| 8.1.2.1.2. | Accounting Data Sorting Keys (Bytes 133-255) | 8-12 |
| 8.1.2.1.3. | Interactive Services Sorting Keys (Bytes 133-255) | 8-13 |
| 8.1.2.2. | Log Record Format | 8-14 |
| 8.1.2.3. | Workstation Log Record Format | 8-14 |
| 8.1.3. | SL\$LOG Console Log Output | 8-15 |
| 8.1.4. | Executing SL\$LOG to Transfer Job and Workstation Log or Console Log Records to Disk | 8-16 |
| 8.1.5. | Executing SL\$LOG to Transfer Job and Workstation Log or Console Log Records to Tape | 8-17 |
| 8.2. | JOB LOG REPORT PROGRAM | 8-18 |
| 8.2.1. | Accounting Report Types | 8-19 |
| 8.2.2. | Executing JOBLOG | 8-22 |
| 8.2.3. | Sample Job Control Streams | 8-23 |

9. ESTABLISHING AND CONTROLLING YOUR SPOOLING AND JOB ACCOUNTING SYSTEM

- | | | |
|------|---|-----|
| 9.1. | GENERAL | 9-1 |
| 9.2. | SPOOLING AND JOB ACCOUNTING CONTROL LEVELS | 9-1 |
| 9.3. | ESTABLISHING OPTIONS DURING SYSTEM GENERATION | 9-2 |

APPENDIXES**A. FORMAT OF OUTPUT WRITER TAPE****B. FORMAT OF OUTPUT WRITER DISK AND FORMAT LABEL DISKETTE****C. OPERATOR'S SPOOLING REFERENCE**

| | | |
|----------|---|-------|
| C.1. | OVERVIEW | C-3 |
| C.2. | GENERAL SPOOLING COMMANDS AND MESSAGES | C-4 |
| C.2.1. | Displaying the Status of Spool Subfiles (DISPLAY ACT/SPL/CNSLG) | C-6 |
| C.2.2. | Displaying the Spooling Mode of Operation (DISPLAY SPL,STATUS) | C-11 |
| C.2.3. | Deleting Closed Spool Subfiles (DELETE) | C-12 |
| C.2.4. | Placing Spool Subfiles on Hold (HOLD) | C-13 |
| C.2.5. | Releasing Spool Subfiles (BEGIN) | C-15 |
| C.2.6. | Breakpointing Active Spool Subfiles (BRKPT) | C-17 |
| C.2.6A. | Changing the Device Type and/or the Number of Copies for Spooled Files (CH SPL) | C-19 |
| C.2.7. | Changing the System Spooling Criteria (SET SPL,BURST/NBURST) | C-20b |
| C.2.8. | Changing the System Accumulation of Job Log and Workstation Log Files (SET SPL,DUMP/ENDDUMP) | C-20c |
| C.2.9. | Changing the Spool File Printout Format (SET SPL,HEADER/NOHDR) | C-20c |
| C.2.10. | Changing the Job Log and Workstation Log File Print Specifications (SET SPL,NOACT/NOLOG/NOPRINT/PRINT) | C-21 |
| C.2.11. | Changing the Console Log and Workstation Log File Specifications (SET SPL,CNSLG) | C-22 |
| C.2.12. | Suppress or Display Forms Change Message (SET SPL,TEST/NOTEST) | C-24 |
| C.3. | INPUT READER COMMAND | C-24 |
| C.4. | OUTPUT WRITER COMMANDS AND MESSAGES | C-26 |
| C.4.1. | Loading an Output Writer (PR/PU/PD) | C-27 |
| C.4.2. | Loading an Output Writer for Nonauxiliary Destined Print Files Created on Behalf of a Workstation User (PR,UID) | C-28 |
| C.4.3. | Loading an Output Writer for Output Destined for a Workstation Auxiliary Printer (RP,UID) | C-29 |
| C.4.4. | Directing the Operation of an Output Writer (Messages) | C-32 |
| C.5. | SPOOL FILE PROCESSING AIDS | C-34 |
| C.5.1. | Saving the Accumulated Job and Workstation Log in SYSLOG (SL\$LOG Program) | C-34 |
| C.5.1.1. | Using Tape for the Job and Workstation Log SYSLOG File (RUN DUMPLOGT) | C-37 |
| C.5.1.2. | Using Disk for the Job and Workstation Log SYSLOG File (RUN DUMPLOG) | C-38 |
| C.5.1.3. | Adding Job and Workstation Log Subfiles to the SYSLOG File on Disk (RUN DUMPLOG) | C-38 |
| C.5.1.4. | Reinitializing the Job and Workstation Log SYSLOG File on Disk (RUN DUMPLOG) | C-39 |
| C.5.2. | Producing Job Accounting Reports (JOBLOG Program) | C-40 |
| C.5.2.1. | Running JOBLOG with Tape Input (RUN JBLOGT) | C-40 |
| C.5.2.2. | Running JOBLOG with Disk Input (RUN JBLOG) | C-41 |
| C.5.3. | Saving the Accumulated Console Log in SYSLOG (SL\$LOG Program) | C-43 |
| C.5.3.1. | Using Tape for the Console Log SYSLOG File (RUN DUMPLOGT) | C-43 |

| | | |
|----------|--|------|
| C.5.3.2. | Using Disk for the Console Log SYSLOG File (RUN DUMPLOG) | C-44 |
| C.5.3.3. | Adding Console Log Subfiles to the SYSLOG File on Disk (RUN DUMPLOG) | C-45 |
| C.5.3.4. | Reinitializing the Console Log SYSLOG File on Disk (RUN DUMPLOG) | C-45 |

D. SPOOLING SUMMARY

| | | |
|--------|--|-------|
| D.1. | GENERATING THE SPOOLING SYSTEM | D-1 |
| D.1.1. | SUPGEN Spooling Parameters | D-2 |
| D.1.2. | I/OGEN Spooling Parameters for Virtual Devices | D-8 |
| D.1.3. | I/OGEN Spooling Parameters for Workstation Auxiliary Printers | D-9 |
| D.2. | CONTROLLING THE SPOOLING SYSTEM | D-11 |
| D.2.1. | Job Control Spooling Statements | D-11 |
| D.2.2. | Spooling Macroinstruction | D-22 |
| D.2.3. | Workstation User Spooling Commands | D-23 |
| D.2.4. | Operator General Spooling Commands | D-26a |
| D.2.5. | Operator Input Reader Command | D-34 |
| D.2.6. | Operator Output Writer Commands | D-34 |
| D.3. | PROCESSING THE SPOOL LOG FILE | D-39 |

INDEX

USER COMMENT SHEET

FIGURES

| | | |
|-------|--|------|
| 1-1. | System Operation without Spooling | 1-3 |
| 1-2. | System Operation with Spooling | 1-4 |
| 2-1. | Basic Spooling Element Relationships | 2-2 |
| 2-2. | Reading Input from Cards and Diskette | 2-3 |
| 2-3. | Spool File Input and Output during Program Execution | 2-5 |
| 2-4. | Writing Direct and Redirected Output | 2-6 |
| 2-5. | Using Main Storage Space to Write to the Spool File | 2-8 |
| 2-6. | Removing and Replacing Blanks in Output Records | 2-10 |
| 2-7. | Writing Output on a First-Terminated, First-Out Basis | 2-11 |
| 2-8. | Sample Output Listing | 2-13 |
| 4-1. | Writing Output Data in Nonburst Mode | 4-5 |
| 4-2. | Writing Output Data in Burst Mode | 4-7 |
| 4-3. | File Status at Time of Breakpoint | 4-9 |
| 4-4. | Using Breakpoint to Queue Files from an Incomplete Job Step | 4-10 |
| 4-5. | Writing Breakpointed Files from the Spool File | 4-11 |
| 4-6. | Conditions of File Availability to the Output Writer | 4-12 |
| 4-7. | Sample Job Log | 4-14 |
| 4-8. | Using the Spool LOG File to Collect Logs for Printing and Accumulate Logs for Future Use | 4-22 |
| 4-9. | Spooling Conditions for Collecting and Printing the Job Log | 4-23 |
| 4-10. | Spooling Conditions for Collecting and Printing the Console Log | 4-24 |
| 4-11. | Spooling Conditions for Collecting and Printing the Workstation Log | 4-25 |
| 4-12. | Spooling Conditions for Accumulating the Job Log and Workstation Log | 4-26 |
| 4-13. | Spooling Conditions for Accumulating the Console Log | 4-28 |

| | | |
|------|---|------|
| 6-1. | Spooling Element Relationships with Remote Batch Devices Included | 6-2 |
| 6-2. | Spooling Element Relationships with Distributed Data Processing Devices Included | 6-5 |
| 8-1. | Format of the Job Accounting Data (Bytes 0-130 of the Job Log's Accounting Records) | 8-3 |
| 8-2. | Typical Job Accounting Report | 8-20 |
| 8-3. | Typical Account Summary Listing | 8-20 |
| 8-4. | Typical Interactive Services Usage Summary | 8-21 |
| C-1. | Typical Job Log Subfile Printout | C-36 |

TABLES

| | | |
|------|---|------|
| 9-1. | Summary of OS/3 Spooling and Job Accounting Capabilities and Controls | 9-3 |
| C-1. | Spool File Directories | C-5 |
| C-2. | Spool File Command Modifiers | C-5 |
| C-3. | Output Writer Function Codes | C-30 |

1. Basic Spooling and Job Accounting Concepts

1.1. INTRODUCTION

Spooling and job accounting is a software component of SPERRY Operating System/3 (OS/3) that greatly enhances a system's performance and your ability to account for its use. With spooling, input and output data is handled more efficiently, increasing job throughput. Additionally, spooling maintains a log file for each job processed in the system. This log file is then available for processing by either user job accounting programs or those supplied by Sperry. We will discuss how these logs are accumulated and used to produce job accounting reports later on in this manual. First, let's take a close look at spooling, the elements that make up the spooling system, and how they work for you.

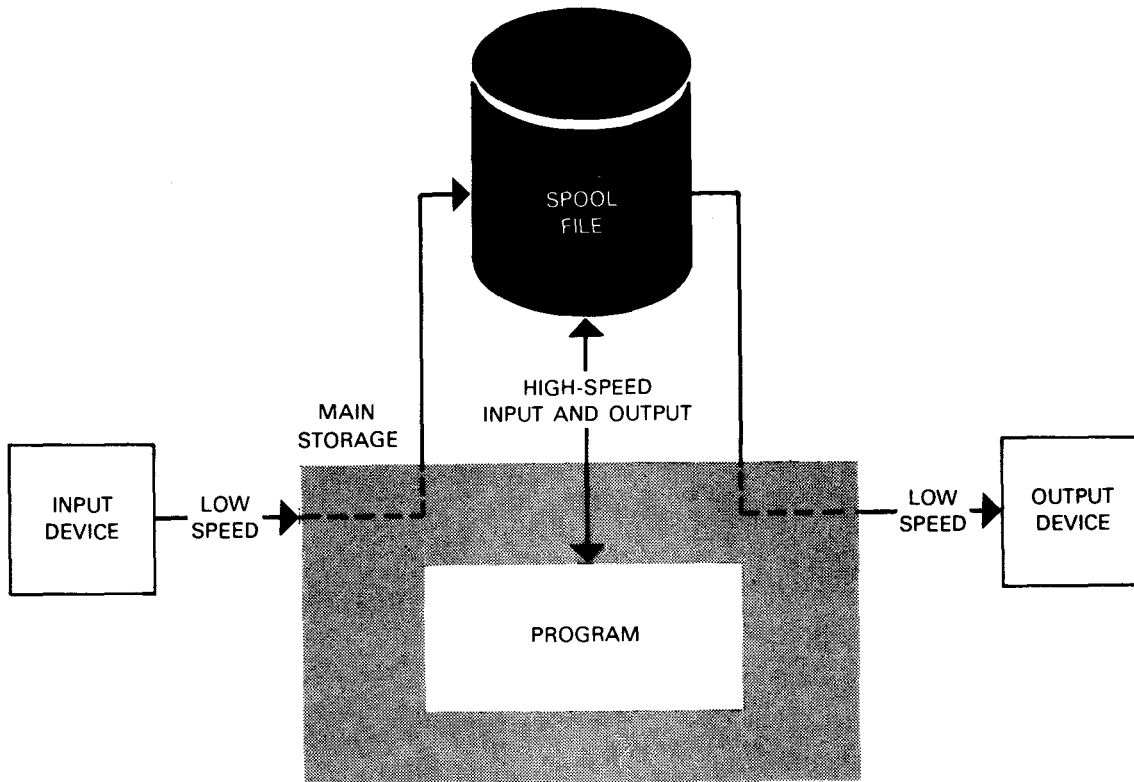
1.2. WHAT IS SPOOLING?

Spooling (simultaneous peripheral operations online) is a technique for increasing job throughput by using a disk to simulate other devices for faster processing. Normally, the central processor is able to handle several programs under execution in main storage at a very high speed. Without spooling, however, certain conditions exist that either prohibit a program from executing or inhibit the processor's throughput efficiency during an executing program's data transfers. First, for a program to begin execution, all of the devices it requires must be available and *dedicated* to that program. Second, once the devices are dedicated and program execution begins, low-speed I/O devices effectively "tie up" the processor whenever a program sends or receives data. These relatively low-speed I/O devices are any diskette drive, tape drive, printer, card reader, and card punch configured in your system.

Spooling overcomes these conditions by:

1. simulating any number of I/O devices required by a program; and
2. compensating for the differences between operating speeds of the central processor and these devices by using a *spool file* (named SYSPool) on disk to hold input and output data. Because disk drives are high-speed I/O devices, the disk data transfer speed most closely matches the speed of the processor.

The central processor, through spooling, accesses this disk spool file at high speed to receive, or *read*, input and to send, or *write*, output for a program. The low-speed devices interface with the spool file independently from the program.



The processor does not have to wait for an input device, such as a card reader, to input the data required by the program. The input data, previously read in from the input device, is available from the high-speed disk spool file. Likewise, the processor does not have to wait until an output device, such as a printer, is available to print the program's output. The output data is written at high speed directly to the spool file, where it is held until the output device is available. In this manner, the spool file is used to *buffer* input and output operations between the high-speed central processor and the relatively low-speed I/O devices. The program is not device dependent, nor is it forced to execute at low speed using those devices.

1.3. WHY USE SPOOLING?

To further see the benefits of spooling, look at the inefficient nonspooling system shown in Figure 1-1.

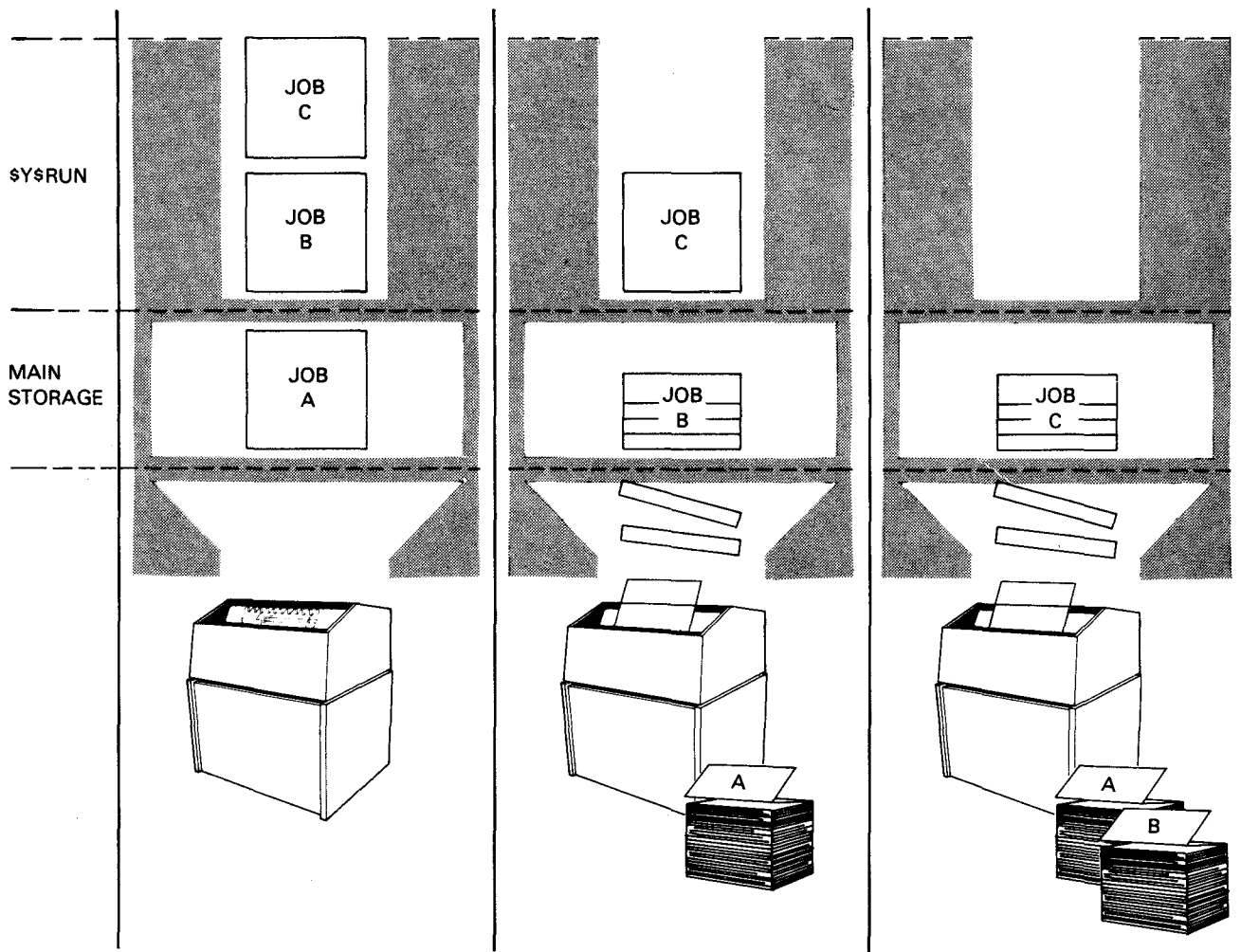


Figure 1-1. System Operation without Spooling

In Figure 1-1, there are three jobs (job A, job B, and job C) to be processed; however, only job A can begin execution using the single system printer. Job B and job C wait in the system run library file (\$YS\$RUN) because they also require output printing and the printer is dedicated to job A for its entire run (until it terminates).

Because job A hasn't started to produce its output, the printer is idle and, therefore, nonproductive. Even after its output is created, job A remains in main storage until its last output record is written over to the printer at the printer speed - not disk speed. Once job A terminates, main storage space and the printer are freed; job B can begin execution while job C continues to wait.

Now look at the same system operation when spooling is configured (Figure 1-2).

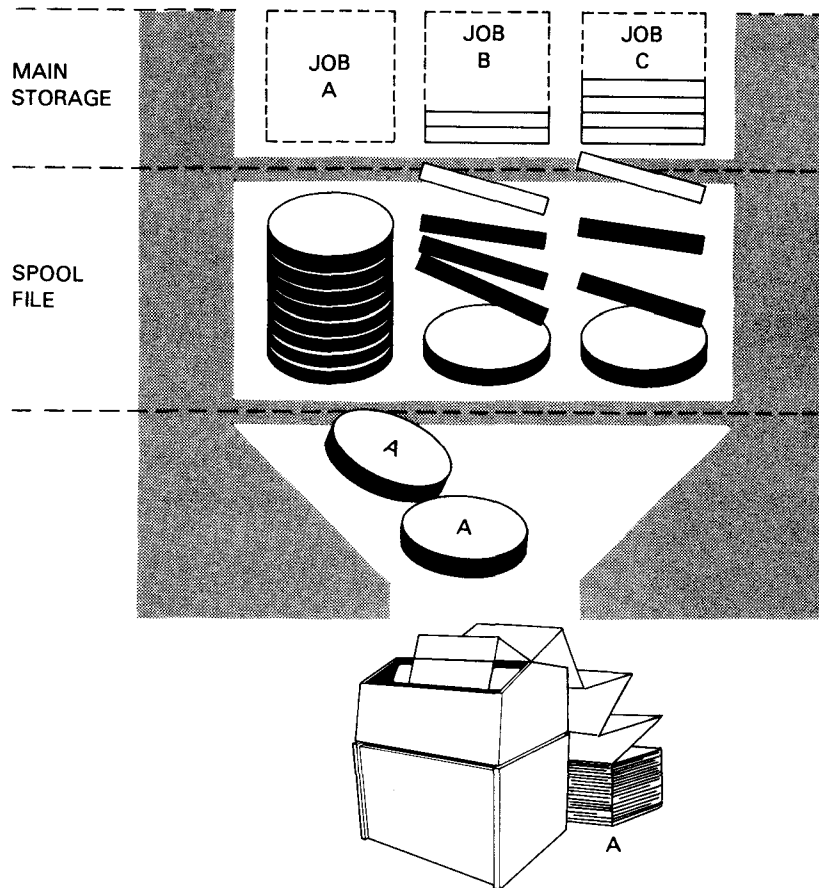


Figure 1-2. System Operation with Spooling

In Figure 1-2, all three jobs are permitted to execute and produce output for the printer - even though there is only one printer.

Why?

Because spooling intercepts each job's printer output and stores it in the spool file. The jobs write their output at high speed to the disk spool file, so each job terminates sooner to set main storage free. See how job A has terminated after writing its last output record? As far as job processing is concerned, job A has actually printed its output. Spooling simulates the printer and stores the output, in printer format, at a representational, or *virtual*, printer address in the spool file. Thus spooling creates a virtual device for every physical, or *real*, device required by each job to be processed. No job waits for a device.

As soon as a job terminates and the printer is available, spooling puts the printer to use. You see in Figure 1-2 that spooling has started to write job A's output to the printer (at the printer's low speed). At the same time, the output from the other jobs quickly accumulates in the spool file and, just as soon as the printer finishes with job A, spooling begins to write the job B output - provided that job B has terminated. Processing continues on through job C's output, continuously using the printer and making main storage space available as soon as possible.

This example of processing job output presents simplified and automatic spooling in a system. It also illustrates these three basic purposes of spooling:

1. Maximizing CPU time and main storage use

Your programs use space and time while reading from and writing to low-speed devices. With spooling, the programs read and write at high speed by using the disk spool file; then they terminate. This saves central processor time and frees main storage space. Spooling takes care of accumulating the low-speed input in the spool file, and of writing accumulated output to a low-speed device.

2. Maximizing the scheduling and running of jobs

With spooling, several jobs that are running concurrently can create output for one device, because the output is stored on the disk and later written to the device. Thus, you can run more jobs that require these devices than the actual number of devices you have in your system.

3. Maximizing the use of I/O devices

Your I/O devices are idled by steps in a job's execution that do not use a device. With spooling, the I/O devices work continuously because data is read and written on a continuous basis independent of a single program's execution.

1.4. WHAT IS JOB ACCOUNTING?

As part of the spooling operation, job accounting information is generated for each job run on your system. This information can be useful for billing purposes and provides an overview of system usage. Some of the information given to you is the amount of main storage allocated and the devices used by each job, the job date and elapsed time, and the clock time. More information on job accounting is presented in Section 8.



2. Spooling Elements

2.1. BASIC ELEMENTS OF SPOOLING

For you to understand spooling concepts, you'll need to know the basic spooling elements and their functions, the relationship between these elements, and the way you establish spooling in your system.

The spooling system comprises these four basic elements:

1. The *input reader*

Reads input to be spooled from a card reader, tape drive, or data set label diskette

2. The *spooler*

Provides the interface between all I/O operations and the spool file

3. The *spool file*

Stores all spooled input and output

4. The *output writer*

Writes spooled output to a printer, card punch, or data set label diskette; and writes print or punch output to a tape, disk, or format label diskette for temporary storage (redirected output).

Figure 2-1 shows you a simplified relationship between these elements during spooling.

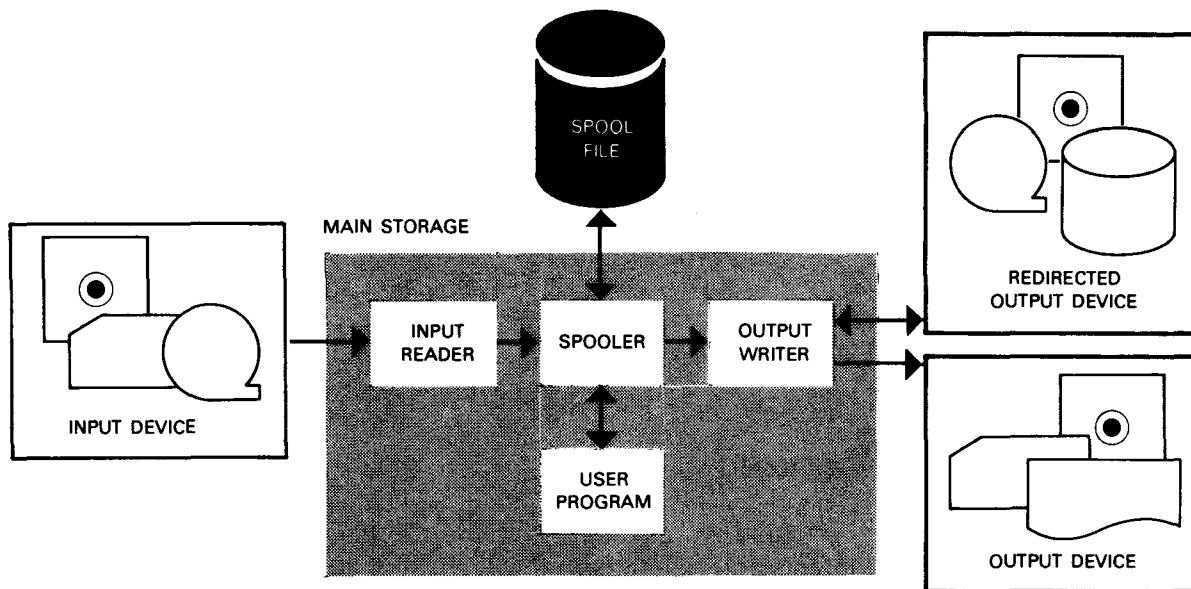


Figure 2-1. Basic Spooling Element Relationships

Instead of reading input from punched cards or a diskette directly to a program in main storage, the input is spooled into the spool file before the program that requires it executes. First, the input reader reads the tape, punched cards, or diskette input data to be spooled. Then, the spooler takes over to write the input into the spool file, where it is held until the program begins execution and requests it. Whenever the program calls for input data, the spooler transfers the input from the spool file to the program.

When the program generates output designated for an output device, the spooler intercepts the output and writes it to the spool file. Once the designated output device is available, the output writer calls on the spooler to obtain the output data from the spool file. The output data is then passed to the output writer to be processed to the device designated by the program.

Figure 2-1 shows that input data may be spooled from a card reader, tape drive, or diskette, and output data may be spooled to a printer, card punch, or diskette. Or, print or punch output may be redirected to a tape, disk, or diskette for temporary storage. This option to redirect and temporarily store output can be used to make space available in the spool file, for example, or to print the nonessential output "after hours", thus allowing more urgent jobs to use the printer during prime time. Later, when you want to print or punch this redirected output, the output writer handles the operation without having to use the spool file.

The following paragraphs discuss the basic spooling elements in further detail.

2.1.1. The Input Reader

The input reader reads records and passes them to the spooler, which in turn writes the input data to the spool file. An input reader works with only one tape drive, card reader, or data set label diskette at a time. However, any number of input readers – one input reader for each tape drive, card reader, and diskette drive in your system – can be functioning concurrently. Thus, multiple read operations from different devices are performed concurrently in a spooling environment. Figure 2-2 illustrates how two input data files are read.

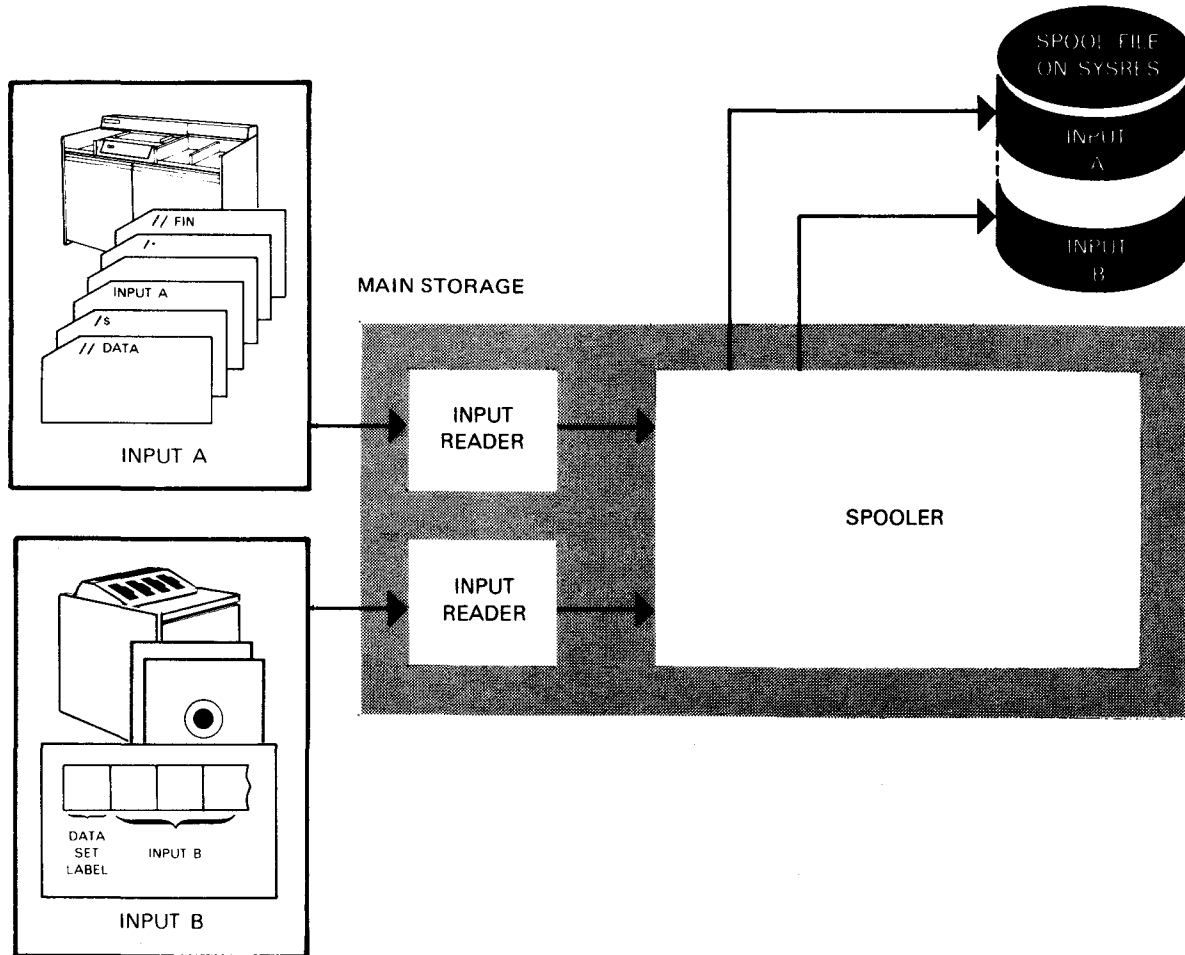


Figure 2-2. Reading Input from Cards and Diskette

The input reader is activated, or *loaded*, into the system by the IN console command entered by your operator. Input files from different devices are processed concurrently by the input readers that are loaded and assigned, one to each device. Any number of tape, punched card, or diskette files may be stored in the spool file at one time, however. This means that while the number of input readers functioning depends on the number of real input devices configured, the number of input files read from a device and stored in the spool file is unlimited. By creating virtual device files to represent the real device originally used for the input, spooling simulates whatever number of devices your executing programs require.

The input data is identified in one of two ways, depending on whether it is on punched cards or diskette (Figure 2-2):

1. A punched card or tape input file is identified by a // DATA job control statement that precedes the first data record.
2. A diskette input file is identified by its data set label, which your operator includes in the IN command. The // DATA and // FIN job control statements are not used to identify diskette input files.

These input identifiers are used later by the spooler to find the virtual input file in the spool file when the program calls for its tape, punched card, or diskette input.

In Figure 2-2, the input reader, loaded via the operator IN command, reads the input A data file. As input A is being processed, the operator enters the IN command with a data set label to load another input reader for the diskette input. This second input reader reads the input B data file. Record by record, the input data files A and B are processed by their respective input reader and stored in the spool file by the spooler.

2.1.2. The Spooler and the Spool File (SYSPPOOL)

We have said that the spooler writes input to the spool file; however, this is only one of the spooler functions. The spooler is the hub of the spooling system. It is the exclusive interface with the spool file, and the spool file contains all the input data needed and output data generated by your programs. The spooler is the only spooling element that always resides in main storage; the input reader and output writer are resident only when active to conserve space.

All input and output goes through the spooler. The spooler intercepts all I/O program requests to selected devices, then accesses the spool file for the input or output requested. The spooler locates input or output data by using a pointer to the virtual device file. Figure 2-3 shows how the spooler accesses the spool file during program execution.

In addition to handling all input to and output from the spool file, the spooler takes care of all error handling, page spacing, and synchronization of the data files. Also, the spooler handles any number of input or output files, including multiple data files per job. The spooler uses spooling control space and buffers reserved in main storage as it reads and writes input/output data, performs various duties associated with the data, and collects records pertaining to the data. Although the spooler is always resident, the spooling control space and buffers are reserved only as needed.

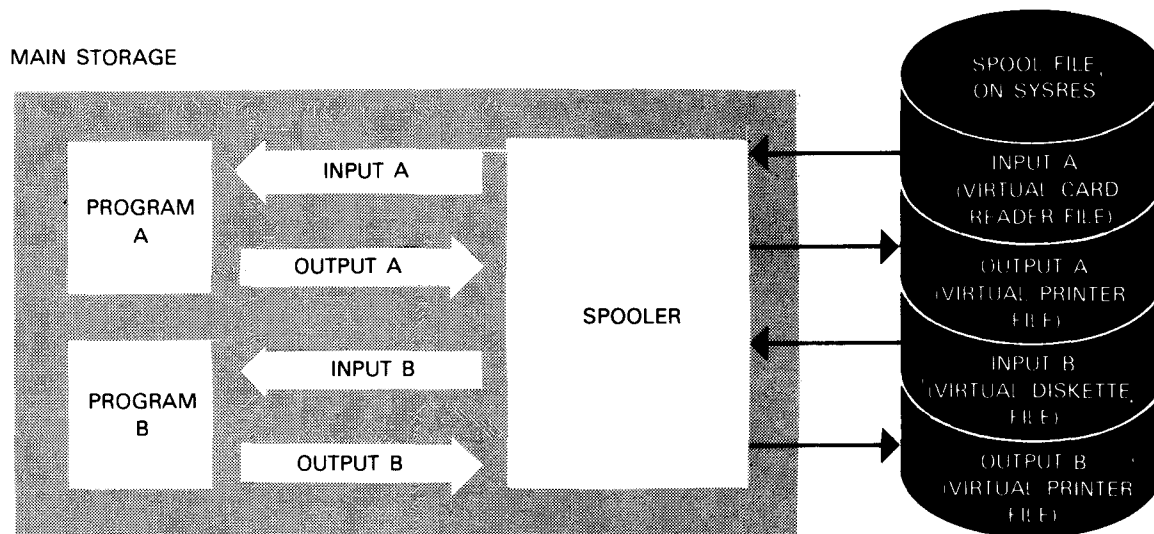


Figure 2-3. Spool File Input and Output during Program Execution

We've already shown (Figure 2-2) that the input A and input B data files currently reside in the spool file after being processed by the input reader and the spooler. In Figure 2-3, program A begins execution and requests input A from the card reader. The spooler intercepts the request, locates the input A virtual card file, and supplies the input A records from the spool file to program A as it requires them. When program B begins to execute, the spooler intercepts program B's request for input B on diskette and obtains the input B records from the input B virtual diskette file.

All I/O request intercepts and data record transfers are handled in this manner so that no program is waiting.

Although there is only one real printer, both programs are allowed to generate printer output (output A and output B), because the spooler writes it to virtual printer files. (As far as the programs are concerned, they process input from and output to the real device.) Because the spool file and the whole spooling operation are transparent to job processing, your jobs do not require special job control or record reformatting.

2.1.3. The Output Writer

Your program's output resides in a virtual device file in the spool file until the real output device specified by the program is available. When that device is free, the spooler reads the output data and the output writer writes it to the output device, or, if requested, to the redirected output device. Although you have many ways of controlling the output writer and of tailoring your output, the output writer functions automatically. It does not require an operator command to load it. Figure 2-4 illustrates how output data is processed directly, or is redirected, to an output device.

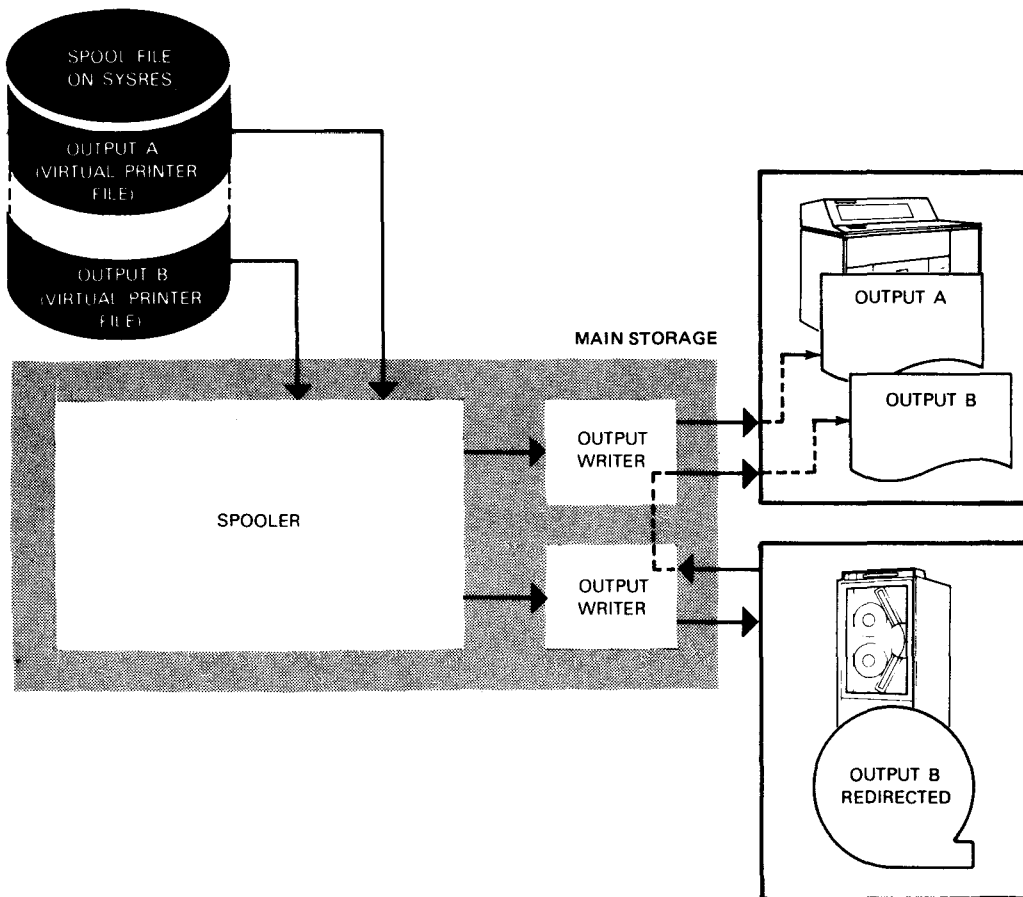


Figure 2-4. Writing Direct and Redirected Output

Like the input reader, a given output writer handles only one device at a time, but there can be an active output writer for every output device in your system. Also, like input files, any number of output files may be stored in the spool file at one time. The output writer writes output directly to a printer, card punch, or a diskette drive, as well as redirecting output to a tape, disk, or diskette drive. When you write output data directly from the spool file to a diskette, it may be used as input to the input reader or your user program. Output data is formatted in *data set label mode* on the diskette when you process it directly for this purpose.

You redirect output data to a diskette so that it may be later printed or punched without having to be read into the spool file first. Because the data redirected to a diskette is formatted in *format label mode*, it may not be used as input to the input reader. Therefore, only the output writer processes redirected data when it is reintroduced, and only for the purpose of printing or punching it. We'll discuss writing output data to a diskette in further detail in Section 4.

As previously stated in our discussion, your output data files currently reside in the spool file, waiting to be printed. (See Figure 2-4.) When the printer becomes available, the spooler reads the output A data file records and the output writer writes these records directly to the printer.

Although output B is designated for the printer also, the operator requests that it be redirected to tape, because the tape drive is available and the output B printer listing isn't needed right away. Therefore, the spooler obtains output B from its virtual printer file in the spool file, and the output writer writes it to the available tape device. Note that if output B wasn't redirected to tape, it would still be occupying space in the spool file while waiting for the printer to finish printing output A. Later on, when the printer is free, the operator enters a command to print the redirected output B.

To further illustrate how the spooling elements are used, let's take a look at how spooling is established in your system and how a job might then be run using spooling.

2.2. ESTABLISHING YOUR SPOOLING ENVIRONMENT

Spooling is established in your system at one of four levels:

1. Output data is spooled and spooling is to local devices.
2. Input and output data is spooled and spooling is to local devices.
3. Input and output data is spooled and spooling is to local and remote batch devices (remote batch processing is included). This level may be further tailored so that spooling is restricted to output only.
4. Input and output data is spooled and spooling includes distributed data processing (automatically includes remote batch spooling). This level may be further tailored so that spooling is restricted to output only.

Once you decide which spooling level best suits your needs, you tailor and install spooling in your operating system during system generation (SYSGEN) time.

2.2.1. Spooling Simplified

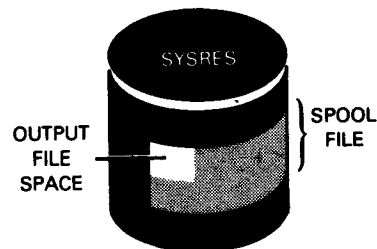
Most of the spooling parameters are entered so that their most popular or usable option is automatically used during spooling operations. An option automatically used for a parameter is called a default. For some parameters, you decide on the particular default you want to use. Once you generate your system, the spooling environment is established and all of your jobs run with spooling, automatically using the parameter defaults, and without programmer or operator control. This is not to say that once spooling is generated into your system, its options cannot be changed or controlled, however. Spooling is very flexible, as we've said. Job control statements and console and workstation commands may be used to select different parameter options from some default options entered during SYSGEN; thus, those spooling functions can be controlled. We'll talk about these options later in Section 9, where all of the ways to control spooling are described.

Let's examine just how one of your jobs would run if:

- you generate your system with an output spooling level for local devices; and
- you use all of the SYSGEN parameters with their automatic default options.

2.2.1.1. Before Your Job Is Run

When your operator performs the *initialization* procedures to load and bring up the system, the SYSGEN parameter default options are included. Some of the spooling options establish the space requirements necessary for the spooling system to operate. Space for your spool file is reserved on your SYSRES disk volume. In addition, the amount of space initially required for each of your output files is established, to be assigned to each file when it opens. If a particular output file requires more, the space allotment automatically increases for that file. The spooling system also requires main storage control and buffer space; however, for efficiency, this space is reserved only as required by the input reader, output writer, or the spooler.



All internal cross-references necessary for spooling to operate are established for each spooling element. These cross-references, or pointers, along with the reserved space, enable the spooling system to process each program's output file or files in your job.

2.2.1.2. After Your Job Run Begins

As soon as you start to run your job, main storage space is reserved for the job. The spooler uses this space to hold, or buffer, data and to control spooling operations. As the job's program, or job step, creates output data records, they are temporarily stored in the buffer space until the spooler writes them to the spool file. Figure 2-5 shows the use of main storage to write to the spool file.

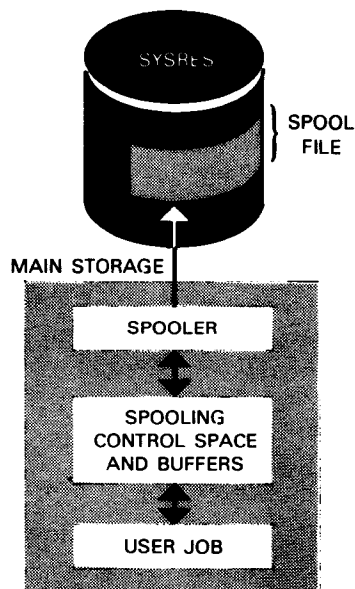
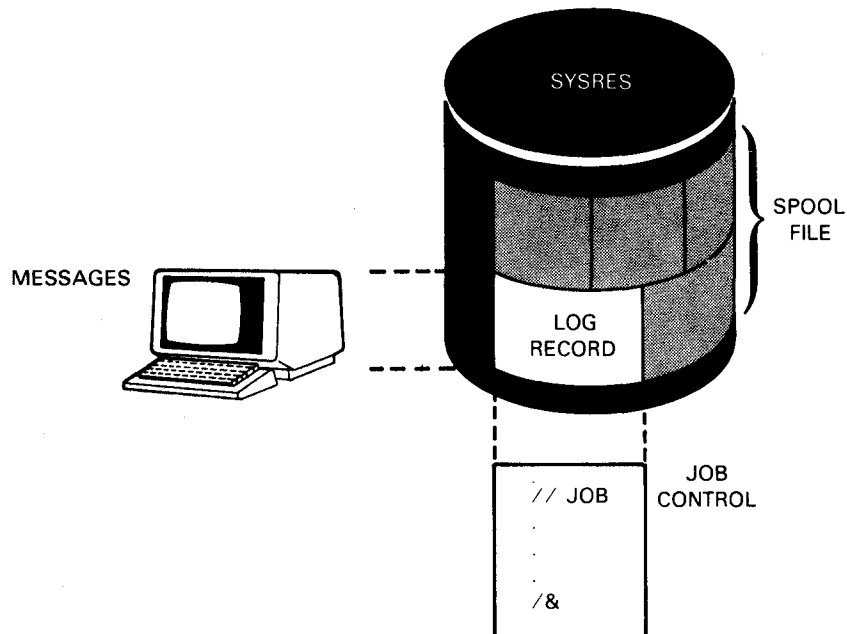


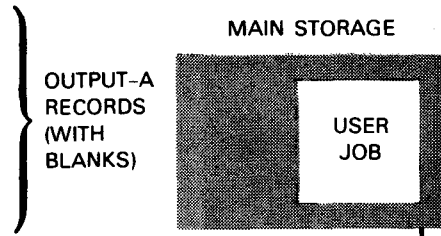
Figure 2-5. Using Main Storage Space to Write to the Spool File

Informational *log records* pertaining to the job are collected in the spool file and printed when the job terminates. These log records contain the job control used and the outgoing messages displayed on the console or workstation screen during job execution.

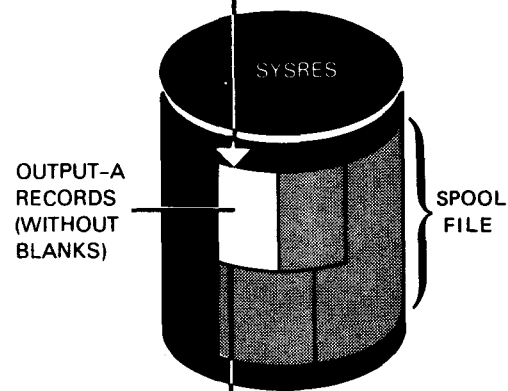
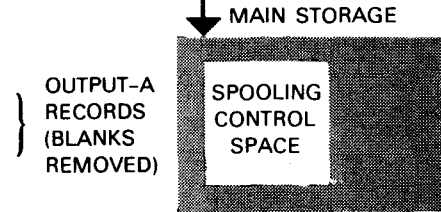


So far, we have described how space is reserved for the spool file on your SYSRES disk and in main storage for the spooler to process the output data from your job step (program). As the job generates output files, these files take up some of this space at different times during program execution with spooling. Your jobs produce output files with spacing characters, or blanks, to assure that the output is properly spaced and aligned for readability. Since these blanks are needed only when printing or punching the output, it would be inefficient to allow them to take up reserved space in the spool file or main storage while the output file waits to be processed. The spooler eliminates wasted space by deleting groups of blanks in each output record as it is generated. The spooler recreates these blanks before the record is written to the printer or punch. Figure 2-6 shows how these blanks are removed and replaced in output records.

| EMPLOYEE NAME | NUMBER | TELEPHONE EXT. |
|-------------------|--------|----------------|
| ROBBINS, SUSAN R. | 100698 | 2311 |
| STEWART, ADAM P. | 101234 | 2682 |
| TOLIN, BARRY K. | 098656 | 2556 |



| EMPLOYEE NAME | NUMBER | TELEPHONE EXT. |
|-------------------|--------|----------------|
| ROBBINS, SUSAN R. | 100698 | 2311 |
| STEWART, ADAM P. | 101234 | 2682 |
| TOLIN, BARRY K. | 098656 | 2556 |



| EMPLOYEE NAME | NUMBER | TELEPHONE EXT. |
|-------------------|--------|----------------|
| ROBBINS, SUSAN R. | 100698 | 2311 |
| STEWART, ADAM P. | 101234 | 2682 |
| TOLIN, BARRY K. | 098656 | 2556 |

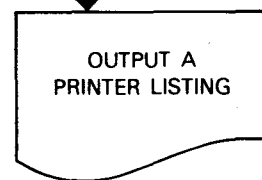
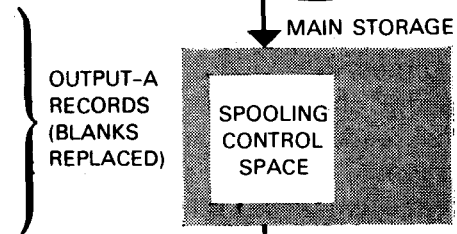
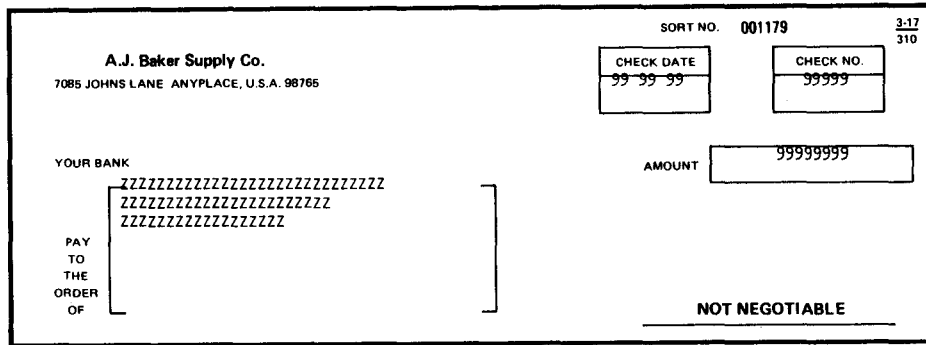


Figure 2-6. Removing and Replacing Blanks in Output Records

If the job output is to be printed or punched onto a special form (for example, print a company check), your operator can check for proper alignment by printing the first page of the form as a sample. For security reasons, the sample contains Z's and 9's rather than actual alphabetic and numeric characters.

After test printing, a misaligned check form might look like this:



As a job generates output data, spooling processes each data record to the spool file, and subsequently to the intended output device, in the order the job step (program) creates them. Since more than one job is usually being run, the spooler processes other job output records as they are created. The job that terminates first consequently has all of its output data residing in the spool file first - available to be written out to a device.

Thus, the order in which a job terminates determines the order in which its output data is written to an output device. Figure 2-7 illustrates this concept of writing output, where all three jobs require the same output device.

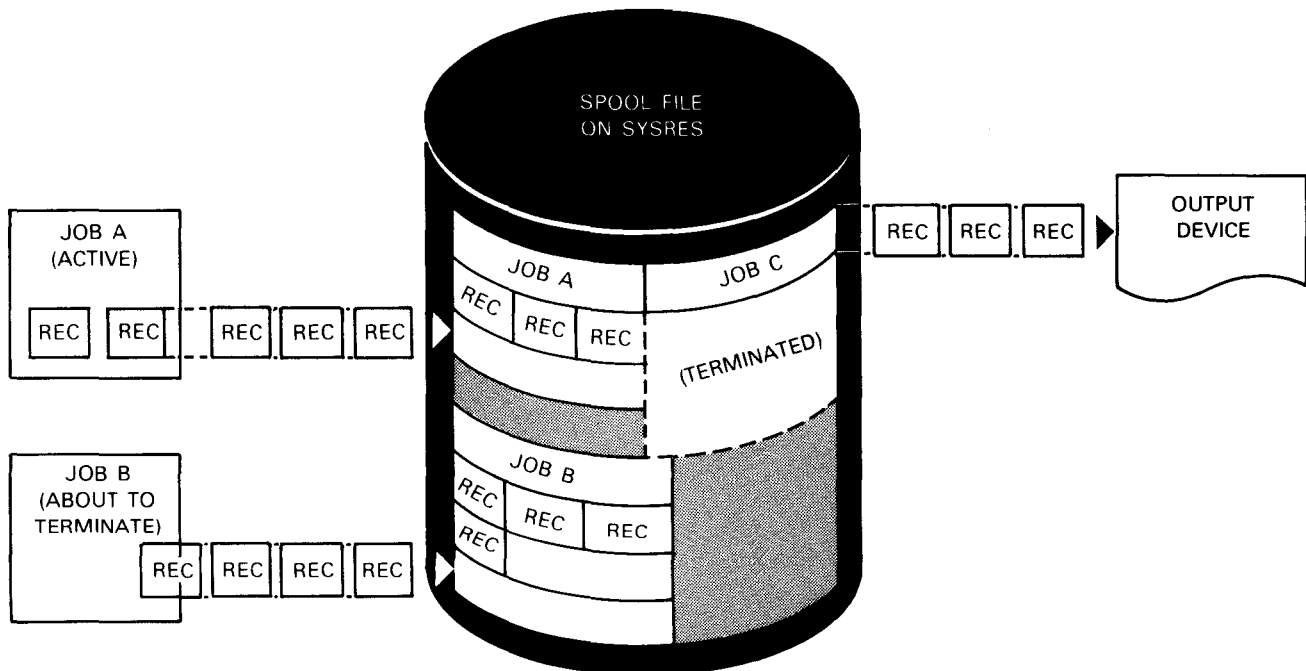


Figure 2-7. Writing Output on a First-Terminated, First-Out Basis

Note that job A is still active and generating output records. The first record created resides in the job A space in the spool file; however, the last output record has not yet been created.

Job B is also active and generating output records. And, like job A, the first record created by job B resides in the spool file and output records continue to be added into the job B space. Job B creates its last output record, however, and is about to terminate.

Job C produced all of its output and terminated. All job C output records reside in the spool file space reserved for them. Since the output device is available, these records are written to it. After job C output is processed and the device is available, the output from job B is written.

In this manner, the contents of one job's output data files are processed consecutively and intact before output from another job, when both jobs require the same device. The integrity of each job's output is always preserved. (Note that if two or more jobs are terminated and waiting in the spool file for the same output device to become available, the job that was run first would be processed first by the output writer.)

Before the printing of your job's output file, a file separator, or header, is printed. Included in the header are the job name and the current date and time, printed in a manner that clearly separates one job from another. The informational log records that have been collected for the job are printed next and then are deleted from the spool file. Once the header pages and log records are printed, the output data produced by your job is printed. Figure 2-8 shows this output format.



3. Input Spooling Functions

3.1. BASIC INPUT FUNCTIONS

When input data is spooled, you are automatically provided several functions. You select or tailor some of these functions during SYSGEN; others are transparent to you. Once your system is generated, input spooling provides additional capabilities for your operator and programmer to control how input is to be spooled. To select a function, your operator enters the input reader command (IN) at the console and includes parameter options. Or, your programmer includes similar parameters in the // DATA job control statement for the same purpose.

As a function of input spooling, you are provided:

- multiple input readers – one reader for each active input device;
- better use of main storage space, because the input reader is nonresident;
- capability of retaining an input file after it is processed;
- option to automatically call a job to process an input file as soon as it is completely written to the spool file;
- capability of using 80-, 51-, 66-, or 96-column punched cards as input to the input reader;
- capability of having multiple input files in your punched card deck;
- capability of using tape as input to the input reader;
- capability of having multiple input files on a single tape volume; and
- capability of using multiple diskettes for one input file.

The following paragraphs describe the input spooling functions. Note that input is handled from local and remote locations, depending on how your system is configured. This section describes the local input spooling functions only; see Section 6 for remote spooling capabilities.

3.1.1. Multiple Input Readers

Multiple copies of the input reader are loaded when you read input from multiple card reader, tape, and diskette devices – one input reader to service one device. Your operator loads the input reader by entering the IN command.

3.1.2. Nonresident Symbiont

The input reader is a nonresident routine, or symbiont; that is, the IN operator command loads it into main storage only when needed, rather than having it reside there all the time.

3.1.3. Input File Retention

Normally, your input data file is deleted from the spool file after a job processes it. Your operator and programmer have the option of retaining the processed input file, however. A retained input file remains in the spool file, available for additional processing, until you delete it (5.5).

When the input file is on tape or is a card deck, the programmer uses a parameter option in the // DATA job control statement to retain the file. When the input file is on a diskette, your operator includes a similar parameter in the IN command.

3.1.4. Automatic Job Scheduling

When input data is written to the spool file (via the IN operator command), you have the option of automatically calling a job to process it immediately, or you can process the input later. To process the data immediately, your programmer includes a // RUN job control statement in the job control stream for the input data on punched cards or diskette. The // RUN statement causes the input reader to schedule the job to process the data. This job is scheduled, however, only when the input file is successfully closed in the spool file. Your programmer can include the statement anywhere in the card or diskette file. To make best use of system resources, however, place the statement immediately following the input data the job requires (for example, to avoid making the run processor (called by // RUN) wait while your operator replaces an input data card jammed in the card reader).

When input remains in the spool file for later processing, your operator enters the RUN command at the appropriate time to schedule the job to process it. The RUN command is used by your operator (at the console) or your programmer (at a workstation) for this purpose.

3.2. INPUT FROM PUNCHED CARDS

When input is from punched cards, both your operator and your programmer are provided with input options. The operator provides options via the IN command and the programmer via the // DATA job control statement. Cards punched in image mode (that is, punched with a binary format), may not be used as spooling input.

3.2.1. Punched Cards Containing Different Column Totals

The input reader reads input from 51-, 66-, and 96-column punched cards, in addition to 80-column punched cards. The 80-column punched card input is automatically processed unless your operator includes the parameter option specifying 51- or 66-column punched cards in the IN command, or your system configuration supports 96-column punched cards.

3.2.2. Multiple Input Files in a Card Deck

Multiple input files in a card deck are read by using the // DATA job control statement that precedes and identifies each input data file. Your programmer can include any number of files, separated by this statement, in a card deck. The final punched card of input data must be a // FIN job control statement.

3.3. INPUT FROM TAPE

When the input is from tape, the tape must be unlabeled and unblocked and contain no block numbers. Your tape input is limited to a single volume. Multiple tape volumes are not permitted. The programmer provides input options via the // DATA job control statement.

3.3.1. Multiple Input Files on Tape

Multiple input files on tape are read by using the // DATA job control statement that precedes and identifies each input data file. Your programmer can include any number of files, separated by this statement, on a tape volume.

3.4. INPUT FROM DISKETTE

When input is on a diskette, the IN operator command contains the input options. The input reader permits your input file to be multivolume, that is, residing on more than one diskette. When a file is multivolume, the input reader creates an input file in the spool file for each diskette volume and links the files together by their data set label, volume serial number, and record size. The operator's parameter option to retain a diskette input file after your job processes it may not be used with multivolume spooled input.



4. Output Spooling Functions

4.1. BASIC OUTPUT FUNCTIONS

The output writer automatically provides you with many functions when you spool your output data. It is loaded automatically whenever the specified device is available for the output data residing in the spool file. However, your operator can manually load the output writer, via a console command, and can control output processing as well. You select some functions or choose a function option at SYSGEN; others are transparent to you. Once your system is generated, output spooling provides additional capabilities to control output processing and the use of system resources.

To select or tailor a function or to override an output spooling option included at SYSGEN, your operator enters spooling commands and messages at the console and includes parameter options. The operator can control the spooling environment throughout your system (via SET SPL commands), control selected groups of spooled output files (via the HOLD and BEGIN commands, for example), and control the particular processing requirement for a specific spooled file (via the RP, PR, PU, and PD output writer commands). Your programmer has similar capabilities to control a specific job's files through the use of job control statements. Or, the programmer uses the spooling commands available to a workstation user to control file processing for jobs associated with the programmer's identification. Your programmer can also control the particular processing environment for a specific spooled file from a workstation (via the RP, PR, and PU output writer commands).

As a function of output spooling, you are provided:

- multiple output writers – one writer for each active output device;
- better use of main storage space, because the output writer is nonresident;
- capability of retaining an output file after it is processed to a device;
- capability of selecting how files are processed via the output writer mode of operation (burst or nonburst);

- option to write an incomplete printer or punch file by temporarily interrupting the job creating it (breakpoint);
- capability of holding files from output writer processing;
- option to print informational header lines to separate files;
- option to print all or selected parts of the job log;
- capability of testing forms alignment on a printer;
- capability of printing the console logs;
- capability of printing the workstation logs;
- option to produce up to 255 copies of an output file;
- capability of directing output to another similar device;
- capability of writing data set label diskette output files to use as input later;
- capability for redirecting printer or punch output to a tape, disk, or format label diskette;
- capability of accumulating and transferring job logs, workstation logs, and console logs for future processing;
- capability of printing output on a workstation auxiliary printer that is locally or remotely connected to your system; and
- capability of printing output continuously or one page at a time to local workstations equipped with an 0791 correspondence quality printer (CQP) and the cut sheet feeder.

Spooled output data is written to local and remote locations, depending on how your system is configured. This section describes local output spooling functions only; see Section 6 for remote batch processing and DDP spooling capabilities.

4.1.1. Multiple Output Writers

Multiple copies of the output writer may be operating together – one output writer to service one device. Multiple output writers are loaded automatically, or are manually loaded by the operator via the output writer command (RP/PR/PU/PD). They can also be loaded manually by your programmer from a workstation via the output writer command (RP/PR/PU).

4.1.2. Nonresident Symbiont

The output writer is a nonresident symbiont (routine); that is, it is loaded into main storage only when needed, rather than having it reside there all the time. The output writer is loaded automatically, or your operator can load it manually with the RP/PR/PU/PD command entry at the console. Your programmer also can load it manually from a workstation with the RP/PR/PU command.

4.1.3. Output File Retention

Normally, your output data file is deleted from the spool file after the output writer writes it to a device. Your operator and programmer have the option of retaining the written output file, however. When you retain the file, first it is processed and then it is placed in a "hold" state (that is, retained and unavailable for further output writer processing). Once the file is in this state, you can either release it for further processing (5.4) or delete it from the spool file (5.5). This contrasts with retained input files, which are available for further processing without being released; they are not in a hold state.

Your programmer uses a parameter option in the // SPOOL jproc to retain the output file in the spool file, or your operator includes a similar parameter option with the RP/PR/PU/PD command entry at the console. Your programmer can also include a similar parameter with the RP/PR/PU command from a workstation to retain the file.

4.1.4. Nonburst and Burst Mode Operation

So far, we have discussed output spooling functions that are very similar to those for input spooling. Output spooling, however, provides many additional capabilities to select and direct how your output is processed and written, and how your system resources are used. The output writer's mode of operation is one of the most important of these functions. The output writer operates in either *nonburst* or *burst* mode as it writes output files to a device. These files are available for processing under different conditions or at different times, depending on which mode is in effect. At SYSGEN, you initially establish the mode of operation, which is normally set to nonburst. Your operator changes the mode to burst whenever necessary and reestablishes nonburst mode when appropriate. The following paragraphs describe nonburst mode and burst mode operation and the advantages of each.

4.1.4.1. The Output Writer in Nonburst Mode

Nonburst mode specifies that the output writer cannot write an output data file to a device until after the job that created the file has terminated and the job's header and log are written. All of a job's output files are processed in the order in which they were created and as a continuous entity. Also, all like output from a job is produced on one device (for example, all of a job's printer files are printed on the same printer).

We looked at SYSGEN-established nonburst mode processing in Figure 2-7, which shows that output from the first terminated job is written from the spool file first, then the second terminated job's output is written, and so on. You establish nonburst mode for the entire spooling system at SYSGEN, or your operator reassigns nonburst by entering the SET SPL command with the nonburst parameter at the console. Although nonburst mode processing ensures the integrity of each job's output, it does not allow you to use parameter options to tailor the output, per se. Your operator does have the option, however, of entering the output writer command (RP/PR/PU/PD) with the nonburst parameter to manually load or direct an output writer to nonburst operation for a particular device. Your programmer also has the same option via the RP/PR/PU command from a workstation. This causes the output writer currently writing a job's print files in burst mode to complete the file in progress, then process any remaining print files in nonburst mode when the job terminates. Another output writer processing data to a different device in either mode is not affected.

Figure 4-1 shows how the output from a terminated job is written to an output device in nonburst mode. Note that the job contains two job steps, and that the second job step created two output files. The job step's data files are written to the spool file, where they wait to be placed into the *spooling queue* when their job step completes. The output writer is permitted to write only files that are in the queue (*queued files*), and the *completion of their job step* is what queues the files. In addition, the output writer must wait until the job terminates before processing the files in nonburst mode.

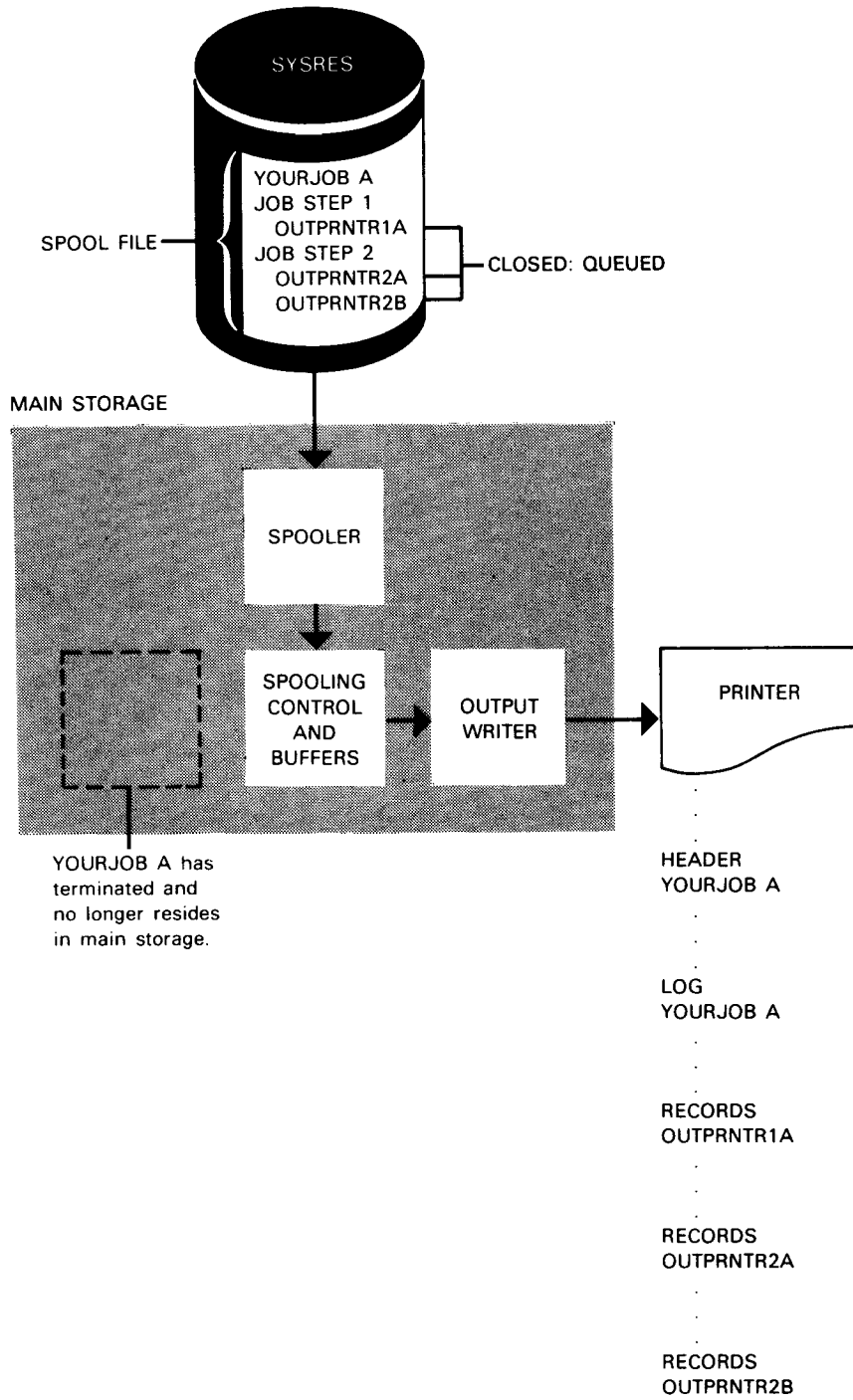


Figure 4-1. Writing Output Data in Nonburst Mode

4.1.4.2. The Output Writer in Burst Mode

Burst mode operation specifies that output files are available for processing before their jobs have terminated. You use burst mode, for example, to obtain a particular output file in the quickest way possible or to make space in the spool file available for other files. The output writer writes an output data file to a device as soon as the job step that created the file is completed and the file is queued, but before the job has terminated.

Burst mode (without tailoring parameters) directs that all queued output files from completed job steps are written out on a first-in, first-out basis, regardless of what job created them. This contrasts with nonburst mode, which directs that queued output files are written according to their job's termination order (first-terminated, first-out) and further, in the order in which their job created them. You include parameter options at SYSGEN, and your operator includes them in a console entry, to tailor how you want the output data in the spool file to be processed. Spooling enables you to determine whether nonburst or burst mode is currently established in your system. Your operator uses the DISPLAY spooling command to display the current mode on the console screen and, if burst mode, the tailoring parameter established with it.

When using burst mode, you usually establish the mode for your entire spooling system. Under these conditions, all output writers are loaded automatically with the completion of any job step, to process the completed (closed) queued files created by the step. A header is written first, followed by the output data file. When the output writer processes more than one data file, it writes a header before each file to separate and identify them. When a completed job step creates two (or more) data files designated for the same type of output device, the first file is written, followed by a header and the second file. However, when you have more than one of that device type available, your operator can load another output writer to process the second file to the second device. In this case, both files are written simultaneously to their respective device. The log is written separately, with a header, when its job terminates, regardless of whether the output files for the job are all written or in what order.

Figure 4-2 shows how the output from an active job is processed to an output device when the entire spooling system is set to burst mode. The figure also shows you how the remaining output is written after the second job step is completed, its output is queued, and the job is terminated.

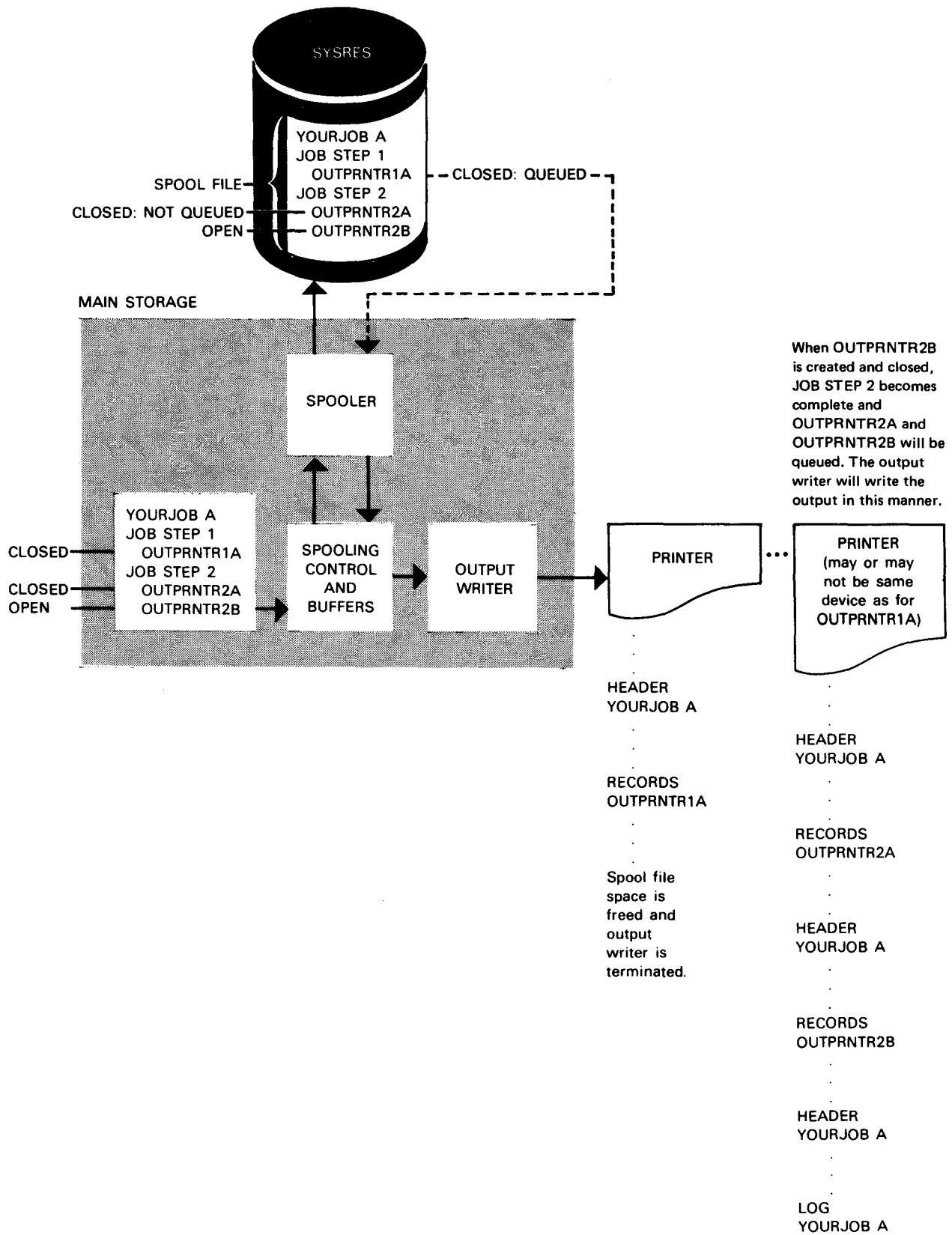


Figure 4-2. Writing Output Data in Burst Mode

Your operator enters the SET SPL command with the burst parameter to establish burst mode operation for the entire system. All output writers to be loaded automatically in the future operate in burst mode. Previously loaded output writers, however, continue to function in the mode established for them, until their assigned tasks are completed. One tailoring parameter option may be included with the SET SPL command to direct output file processing. In this manner, for example, your operator can specify that only those output records created by a particular job, or with the same file name, or designated for the same device type are to be processed.

Your operator enters the output writer command (RP/PR/PU/PD) with the burst parameter to load or direct an output writer in burst mode to satisfy a particular processing requirement. Your programmer can perform this same function by entering the command (RP/PR/PU) from the workstation. For example, an output writer can be loaded to begin writing output designated for a specified device type. Or, an output writer currently operating in nonburst mode can be directed to change to burst mode. Other output writers currently loaded or to be loaded in the future are not affected. When an output writer is manually loaded or directed to burst mode, the operator may include up to three parameter options to tailor the output.

4.1.4.3. Using Nonburst and Burst Modes

As we look at Figures 4-1 and 4-2, some advantages of operating in nonburst and burst modes can be seen. When you want all of a job's output data files processed consecutively as an entity, you use nonburst mode. Burst mode operation, however, permits you to write queued output data files from any completed job step to the output device, without waiting for the job to terminate. The flexibility these two operating modes provides is further enhanced when special processing conditions occur.

Under certain conditions, a warning message is displayed at the console indicating that the spool file space is nearly depleted. This may occur, for example, if you are running a very large printing job and, as the records are created and stored in the spool file, the output writer in nonburst mode cannot begin because the job isn't terminated. Your operator can make some of the spool file space available by loading an output writer in burst mode to process printer files; however, you must have queued printer files in the spool file, and they should be fairly large files to effectively make space available. Spooling provides a method, called *breakpointing*, of making printer or punch files available for processing whether or not they are complete.

4.1.4.3.1. Breakpointing Printer and Punch Files

So far, we know that when a job terminates, the files created by the job step are closed. Records residing in the spooling buffer space that pertain to the closed files are written by the spooler to the appropriate file in the spool file. The job step's files are then placed in queue and are available for processing.

Breakpointing allows you to queue an *incomplete* job step's printer or punch files in the spool file, thereby making them available to be written in burst mode. (Files created for a diskette may not be breakpointed.) The breakpoint temporarily stops the job step from creating records for those files. The buffer records are closed and written to the spool file; then the breakpointed files are placed in the spooling queue.

You can breakpoint files either immediately or at the end of a printer page. When you breakpoint immediately, any records in the spooling buffer space that pertain to the breakpointed files are immediately written to the spool file. When you breakpoint at the end of a page, the job step is permitted to create additional records until there are enough in the buffer space to fill a printer page, at which point they are placed in the spool file.

Figure 4-3 shows you the file status of a large printer job at the time a breakpoint is entered.

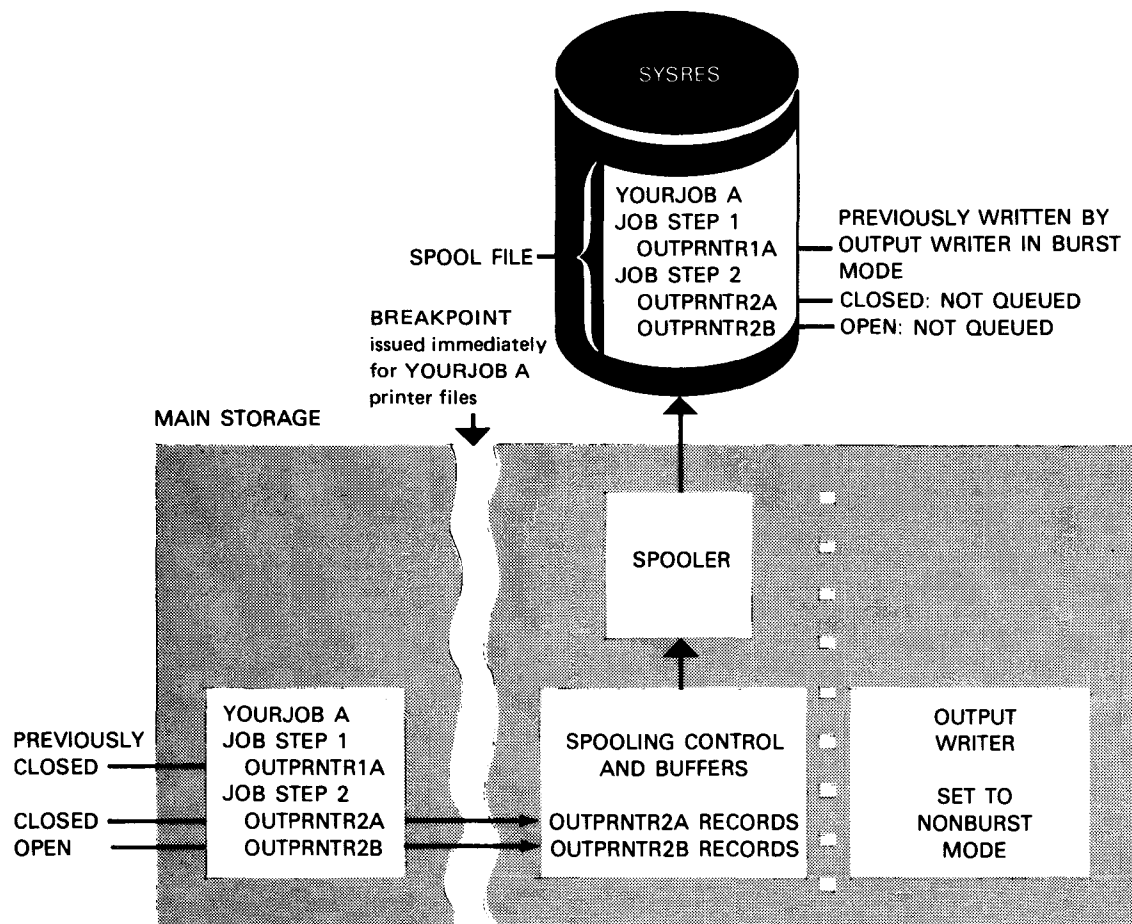


Figure 4-3. File Status at Time of Breakpoint

Before a breakpoint is issued, none of the job step 2 files in the figure may be processed because the output writer operation is set to nonburst mode. Your operator manually loads an output writer in burst mode to write the closed and queued printer file from the completed job step 1. However, let's assume in this example that your spool file space is nearly depleted and the job step 1 file is small and does not release sufficient space for the remainder of the job run. Consequently, your operator issues an immediate breakpoint against the job's printer files. As the breakpoint suspends the creation of records for the printer files, printer records in the buffer are written immediately to their respective printer file in the spool file, which is then queued (Figure 4-4).

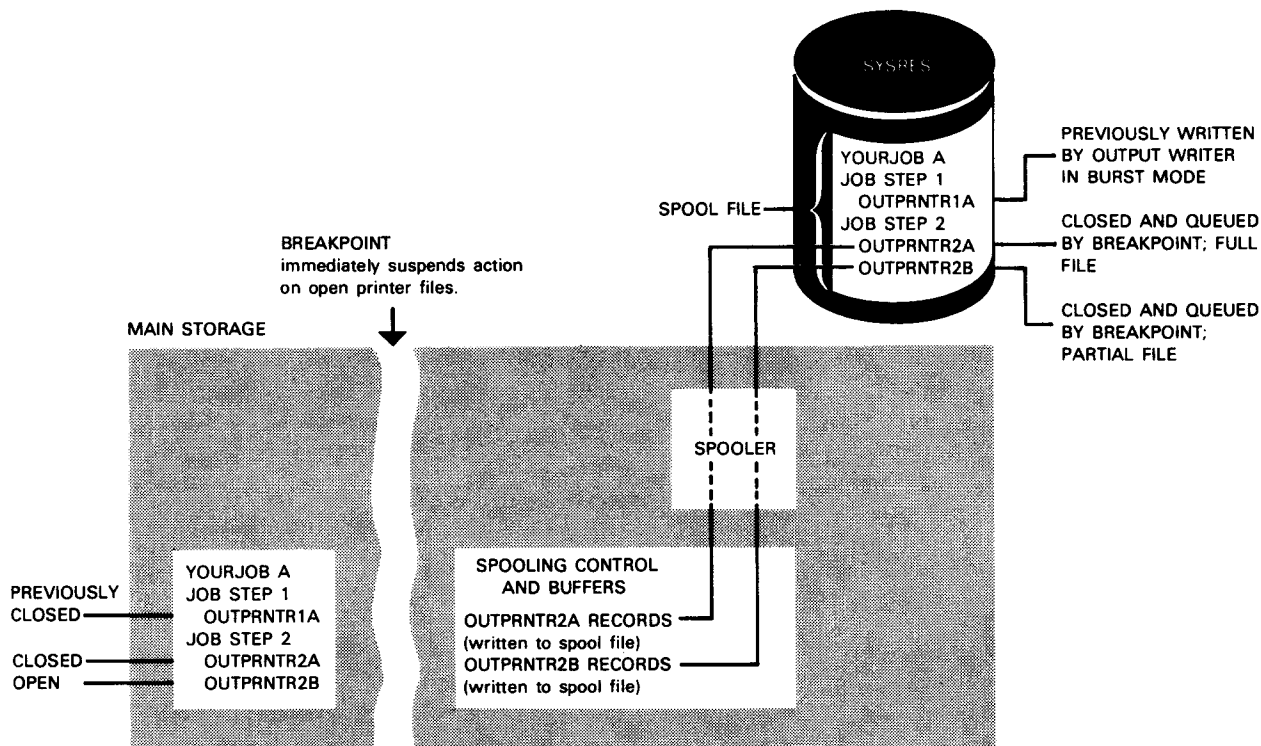


Figure 4-4. Using Breakpoint to Queue Files from an Incomplete Job Step

Once the printer files specified by the breakpoint are closed and queued, their job step resumes processing to create the remaining records. These remaining records are processed separately as a new file.

Figure 4-5 illustrates how the breakpointed printer files are available for processing while the remainder of the file is being created. In this example, the operator manually loads an output writer in burst mode to write the job step 2 files, because the spooling system is set to nonburst operation. If a breakpoint is required when the system is already set to burst mode, an output writer is loaded automatically after the breakpoint is issued.

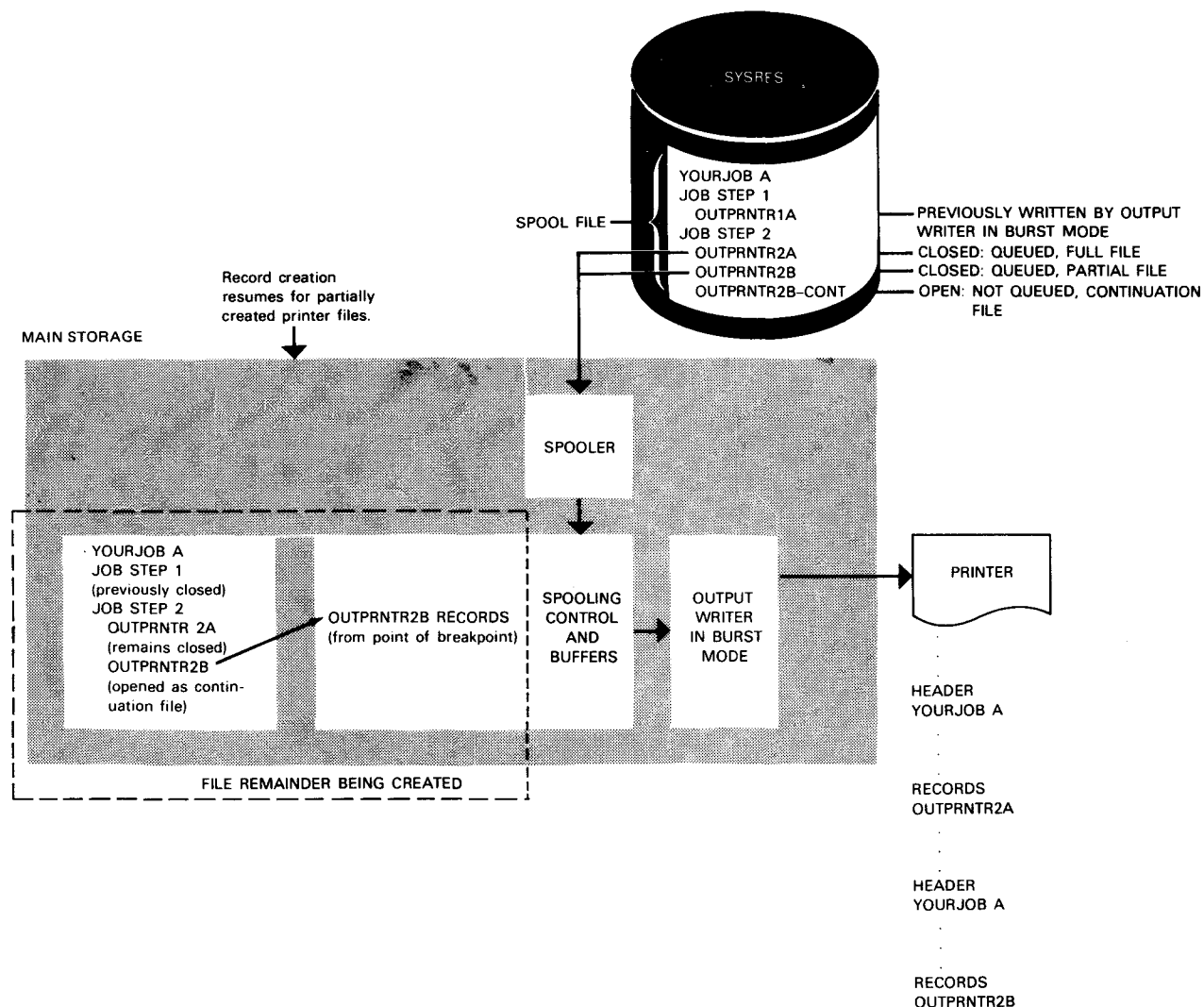


Figure 4-5. Writing Breakpointed Files from the Spool File

Your programmer uses the `// SPOOL jproc` to breakpoint a file after a specific number of pages are accumulated. Or, the programmer includes a breakpoint macroinstruction in the program to issue an immediate breakpoint. Otherwise, the breakpoint command (BRKPT) is entered by your operator (from the console) or programmer (from a workstation) to close and queue a file that is being created, either immediately or at the end of a page.

A breakpoint also may be used if your program malfunctions. Spooling automatically counts the number of output instructions your program issues when creating a file. If this number is exceeded, as may occur when a program is in a loop, a message is displayed to the initiator of the job. The message response options allow the initiator (from the console or a workstation) to breakpoint the file immediately or at the end of a page, to ignore the query, or to cancel the job. This enables the initiator to decide whether the file should be breakpointed and allowed to continue, or be cancelled.

Figure 4-6 shows the conditions under which files are available to an output writer. Two job steps and their associated output files are shown within a user job. The top of the figure shows the points at which the files are available in burst and in nonburst mode. The bottom shows file availability is altered because a breakpoint is issued.

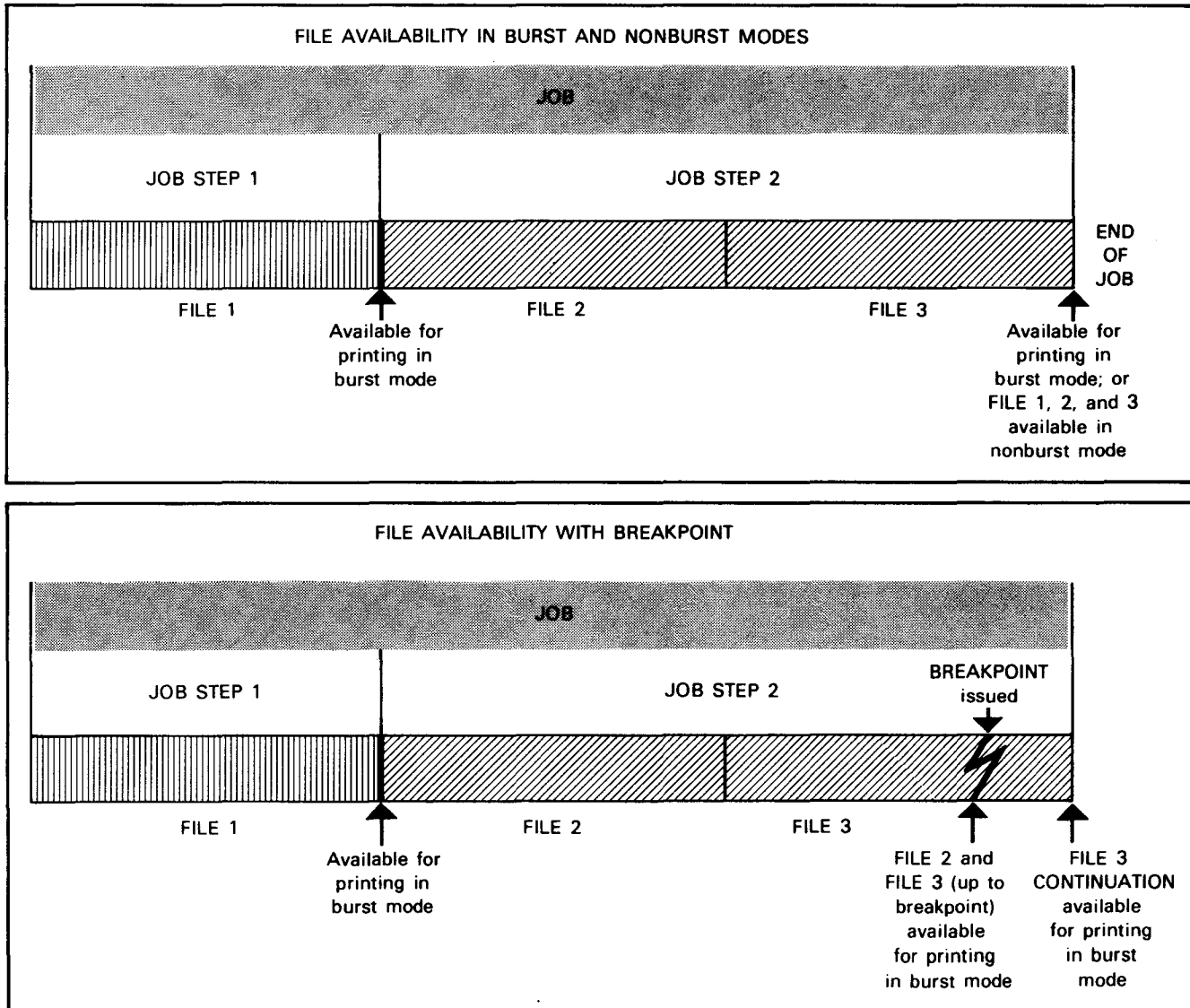


Figure 4-6. Conditions of File Availability to the Output Writer

4.2. OUTPUT TO PRINTER AND CARD PUNCH

When output is to be printed or punched, you and your programmer and operator are provided with output options.

4.2.1. Holding Files from Output Writer Processing

The operational mode you specify determines how your output files are processed in your system. You may, however, remove specific files from nonburst and burst mode processing by making them unavailable to the output writer.

Your programmer uses the `// SPOOL jproc` or enters the `HOLD` spooling command at a workstation to place queued printer or punch files in a hold state. The held files are not available for processing until your programmer (from a workstation) or your operator (from the console) releases them via the `BEGIN` spooling command.

The `HOLD` command is available to your operator at the console with the same capabilities as those provided at a workstation. In addition, the `HOLD` command console entry allows the operator to hold files currently being created, files already queued, or both. Also, your operator can manually direct a currently active output writer to hold the file it is processing until a `BEGIN` command releases the file.

This option to remove (hold) printer and punch files from nonburst and burst mode processing is part of the general options spooling provides to allow you to control all files in the spool file (input, output, and log). We discuss these general spooling control options separately. (See Section 5.) In the discussion, we describe your option to hold files from output writer processing in further detail, as well as your options to hold input files and local job's logs from program processing, and to release (`BEGIN`) held files.

4.2.2. Headers to Separate Files

Remember, in nonburst mode, one header is printed followed by the job's log and output files, whereas, in burst mode, a header is printed before each output file and before the job's log. A header is used to separate and identify all output. It contains the job name, account number, current date and time, and other information pertinent to the job. Figure 2-8 shows a typical job header.

At `SYSGEN` you may include the parameter to suppress headers for your output files. Your programmer suppresses printing a header for a particular output file or for a job by using the `// SPOOL jproc` and `// JOB` job control statement with the appropriate parameter options, respectively. Your operator overrides any header printing conditions previously established by entering the `SET SPL` command to establish or suppress printing headers throughout the spooling system.

4.2.3. Printing the Job Log

We have said that informational records pertaining to your job are collected in the spool file as the job runs. These log records contain the job control statements processed for the job, the messages displayed on the console or workstation screen in behalf of the job, and any dumps processed for the job. All messages regarding the program status are included in the log. In addition, informational records to be used for accounting purposes are collected, provided you include the job accounting option at SYSGEN. (We discuss how you produce job accounting reports in Section 8.)

From our discussions in this section, we know at what time a job's log is automatically printed, according to whether the output writer is set to burst or nonburst mode. The printed job log is normally used by the person who submitted the job for debugging and bookkeeping purposes. The type of each log record (L or A) is printed in column 122, as shown in the following sample job log (Figure 4-7). L-type, or LOG records, are those pertaining to job control and console messages, and A-type, or ACT records, are those used for accounting.

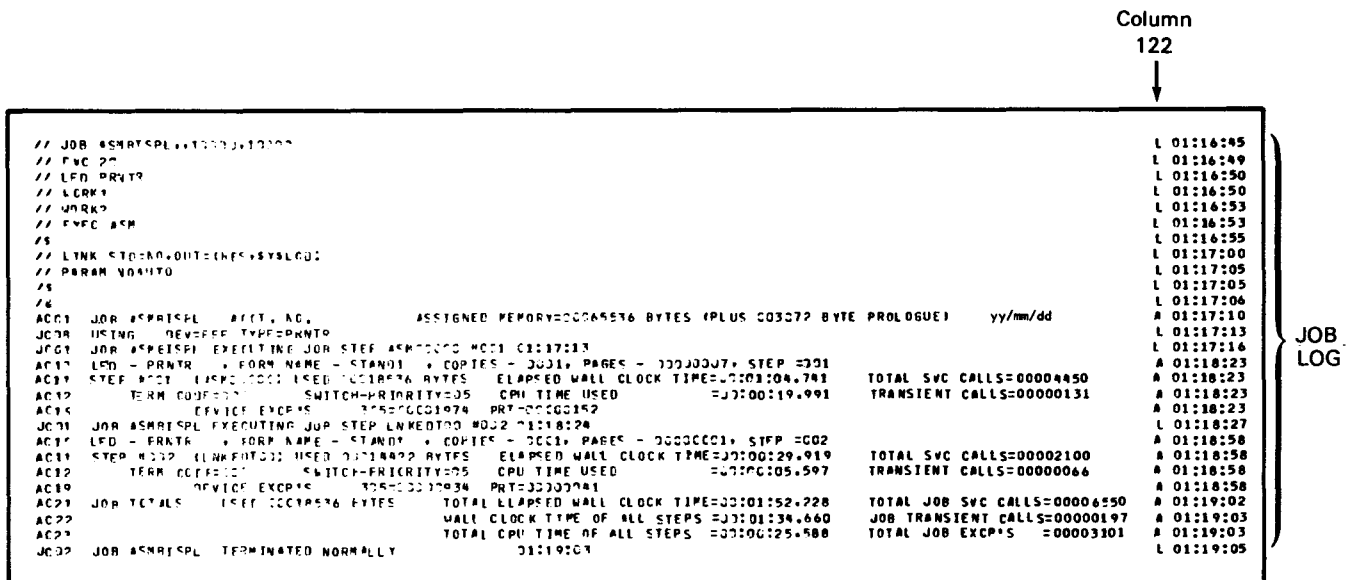


Figure 4-7. Sample Job Log

You have several options at SYSGEN time to tailor how you want the logs for your jobs to be printed. Similarly, parameter options in the // JOB statement and the SET SPL command permit your programmer and operator, respectively, to change the SYSGEN selection. Job log options allow you to print or to suppress printing of the log, or to print or not print only the LOG or only the ACT records from the log.

When a job creates output that must be printed on a specific printer, your programmer assigns that printer in the // DVC job control statement (for example, with a job control device assignment using //DVC 24 rather than // DVC 20 for any printer). To assign the same printer for the job's log, your programmer includes the // OPTION job control statement to specify the device (printer 24 in our example). Because a job's log is normally printed in nonburst mode for the first available printer, followed by the job's output, the // OPTION statement ensures that the log is not separated from the output; they are both printed on the required printer.

Job logs may also be accumulated and transferred to a tape or disk; however, this is a separate option and is described in 4.6.

4.2.4. Test Lines

When we explained how a job runs with spooling, we discussed the messages your operator received each time the output writer detects a change in form name. The operator is first requested to mount a specific form, then asked whether the alignment of the form should be tested. Forms alignment is tested by printing the first page of output with all numeric characters converted to 9's and all alphabetic characters converted to Z's.

The output writer does not query the operator for test lines pages if you suppress this feature during SYSGEN. Your programmer has the option of suppressing or displaying the test lines message for any job by including a parameter option with the // SPOOL jproc to override the SYSGEN selection. Your operator uses the SET SPL command to achieve the same purpose.

You should suppress the test line feature whenever page mode processing is selected for output directed to local workstations equipped with 0791 correspondence quality printers. (See 4.3.)

4.2.5. Printing the Console Log

The console log function provides a printed record of all communication during a working session between your system and your console without using a communications output printer. Those system messages entered from and directed to the console are collected in the spool file. At the end of a working session, your operator initiates an orderly system shutdown to disconnect all files, produce an accounting report of the job processing activity from the accumulated job and workstation log, and print the console log. Each console log record is identified by a C printed in column 122. (We discuss how you accumulate job logs and produce job accounting reports separately in 4.6 and Section 8, respectively.) The console log for that working session is deleted from the spool file after it is printed. If your operator doesn't print the console log when the session ends, log recovery must be specified during the next initial program load (IPL). Otherwise, the console log is lost.

You print the console log by breakpointing it. When your operator enters the BRKPT command for the console log, all records collected for the session, up to the breakpoint, are printed and then deleted from the spool file. Console log record collection resumes until another breakpoint. If you'd rather print the console log at some later time, you can direct the console log output to tape, disk, or diskette rather than to the printer. To do this, have your operator include the appropriate OUT=TAPE, DISK, or DISKETTE option when he enters the BRKPT command.

At SYSGEN time, you establish whether you want to collect the console log records and whether you want the console log printed when you breakpoint it. When you establish console log record collection, your operator overrides the SYSGEN selection by entering parameter options with the SET SPL command to turn the console log function on or off and to permit or suppress printing of the console log.

Console logs, like job logs, may be accumulated in the spool file until they are transferred to a tape or disk for further processing. This is a separate option, however, and is described in 4.6.

4.2.6. Printing the Workstation Log

The workstation log function is very similar to that for console logging. A workstation log provides a printed record of all communication between your system and workstation during a workstation session. All system commands and messages entered from the workstation and all system messages directed to the workstation are recorded in the log. It includes records generated by a workstation and by a terminal that functions as a workstation.

Record collection begins when your programmer logs on the workstation and continues until the programmer breakpoints the workstation log or logs off to end the session. When your programmer enters the BRKPT command for the workstation log or enters the LOGOFF workstation command, all records collected for the session are printed and then deleted from the spool file. For breakpointed logs, record collection resumes until another BRKPT or a LOGOFF is entered. The type of each record in the workstation log is printed in column 122 of the output. Workstation records are identified by W (R when a terminal is used as a workstation) and accounting records by A. If you'd rather print the workstation log at some later time, you can direct the log to tape, disk, or diskette rather than to the printer. To do this, have your programmer include the appropriate OUT=TAPE, DISK, or DISKETTE option when he enters the BRKPT command.

Workstation record collection is included in the SYSGEN parameter you specify to establish console log record collection. When you generate your system with workstation and console log record collection, your operator uses parameter options with the SET SPL command to override the SYSGEN selection and turn workstation logging on and off. When it is turned on, your programmer uses the LOGON command to permit or suppress printing the log. Your operator may override the printing condition specified by LOGON, however, by using the SET SPL command and appropriate parameters.

Workstation logs, like job logs and console logs, may be accumulated for future use. When you accumulate workstation logs, they are held with the job logs in the spool file until they are transferred to a tape or disk for further processing. We describe this separate option in 4.6.

4.2.7. Multiple Copies of a File

More than one copy of any output file may be produced by a given output writer. Your programmer uses the // SPOOL jproc and your operator uses the COPIES parameter option with the output writer command to print or punch up to 255 copies of a file.

4.2.8. Device Swapping

You may direct the output writer to swap devices, that is, to change the device it is currently using to print or punch its output. If, for example, a paper jam occurs in a printer, your operator can manually direct the output writer to use another printer or to write the remaining output to a different device. This allows the output writer to continue processing the output, rather than waiting for the original device to be made available.

When you direct output to another printer or to a card punch, and when you direct punch output to another card punch, you use the device swapping function of the output writer. When you write printer or punch files for temporary storage to a disk, format label diskette, or tape, you redirect the files for later printing or punching. We discuss redirected output to tape, disk, and diskette in 4.5.

4.3. OUTPUT TO WORKSTATION AUXILIARY PRINTER

If your system has locally or remotely connected workstations with auxiliary printers, you can print your print files at these auxiliary printers. These print files can be output files created for a job or from an interactive program such as the general editor (EDT). If you are running your own job, this output can be printed automatically at the end of a job or any time during the job by entering the RP command from the workstation. If you are running an interactive system program, you can print your output files at any time during your program session by entering either the interactive services PRINT command or the EDT @LIST command. (See the interactive services commands and facilities user guide for information on the RP command and PRINT command; the general editor (EDT) user guide/programmer reference for the @LIST command.)

In both cases, there are two requirements that must be met before any files can be printed on an auxiliary printer:

1. Your system must be generated to use locally or remotely connected workstations with auxiliary printers. To do this, you must include certain I/OGEN parameters when the system is generated. If you want to print files on locally connected auxiliary printers, you must include the WORKSTATION and AUXPRINTER parameters. If you want to print files on remotely connected auxiliary printers, you must include the REMWORKSTATION and REMPRINTER parameters.

2. You must specify that your print files are to be directed to a workstation auxiliary printer. How you specify this depends on whether you use an interactive system program, such as EDT, or run your own user job.

If you use an interactive system program, you enter the SCREEN WKSTN command to specify that your print files are to be directed a workstation auxiliary printer. Then, during your program session, you enter the interactive services PRINT command or the EDT @LIST command to print your files on the auxiliary printer. (See the interactive services commands and facilities user guide for information on the SCREEN WKSTN and PRINT commands; the general editor (EDT) user guide/programmer reference for the @LIST command.)

If you intend to run your own job, you must specify in the job control stream that your print files are to be directed to a workstation auxiliary printer. There are three job control statements that direct print files to an auxiliary printer. These are: // OPTION OUT, // OPTION LOG, and // ROUTE.

The // OPTION OUT job control statement directs all job output (print files and job logs) to a specified user. The format for auxiliary printer output is:

```
// OPTION OUT={ORIGINATOR }  
                {user-id }
```

ORIGINATOR specifies the user who initiated the job. If you specify ORIGINATOR, all job output (print file and job logs) from a job you initiated at your workstation will be directed to the auxiliary printer associated with your workstation.

You can also direct your print file output from a job initiated on your workstation to another user's workstation auxiliary printer. You do this by specifying the other user's user-id in the // OPTION OUT job control statement. It is important to note, however, that if you direct your output files to another user's auxiliary printer, that user must issue an RP command to initiate printing. Printing in this case is not automatic. (Keep in mind that if you are not the initiator of a job, but output is directed to your auxiliary printer, you must be aware of this. To find out if any output files have been directed to your user-id, use the DISPLAY SPL command. If any files have been directed to your user-id, you must then issue an RP command to start printing.)

The // OPTION LOG job control statement allows you to direct only the job log from your program to the auxiliary printer associated with your workstation. Its format for auxiliary printer output is:

```
// OPTION LOG=ORIGINATOR
```

You cannot use this statement to direct your job log to another user's auxiliary printer. It can be used only to direct your job log to the auxiliary printer associated with your workstation.

The // ROUTE job control statement allows you to direct specific files to an auxiliary printer. Its format is:

```
// ROUTE user-id
```

If you want to direct a specific file to an auxiliary printer, you include the // ROUTE job control statement in the device assignment set for that file. The user-id you specify can be your own under which you logged on and under which your program is running. It can also be another user's whose workstation has an auxiliary printer. However, the same restriction applies: If you specify another user's id, that user must issue an RP command to start printing. It is not automatic.

More than one job control statement can be used in a single job control stream to direct print files to different users' auxiliary printers. For instance, you can specify // OPTION OUT and then use // ROUTE to direct a specific print output file to another user-id as shown in the following example.

```
// JOB PAYROLL
.
.
.
// OPTION OUT=ORIGINATOR
.
.
.
// DVC 20
// ROUTE USER02
// LFD PRTFIL
.
.
.
// EXEC PROG1
/&
```

As you can see, all print output files are directed to the job's initiator except for the file named PRTFIL, which is directed to USER02. Because USER02 is not the job's initiator, USER02 must issue an RP command to begin printing.

You can also specify, in the job control stream, that the print output files you direct to an auxiliary printer are secured. When a file specified as secured is directed to a user, the user must be logged on if printing is to begin. You secure a file by using the // SPOOL jproc or the // SPL job control statement.

Printer output specified as destined for the workstation auxiliary printer can also be printed at the central site by entering the RP command with the UID=user-id parameter from the system console. If this is done, all auxiliary printer output created on behalf of the workstation user (identified by the UID parameter) will be printed on the central site printer rather than at the workstation auxiliary printer.

If you have an 0791 correspondence quality printer connected to your local workstation, you now have a choice of printing output in either of two modes: continuous or page. In continuous mode, the output prints continuously as in any normal print operation. In page mode, the output is printed one page at a time, with intervening forms mount messages at the end of each page requesting the operator to mount the next page. This mode is especially useful if you are concerned with optional word processing operations.

→ Mode selection is available via the SCREEN command, which is described in the interactive services commands and facilities user guide/programmer reference, UP-9972 (current version). Once selected, the mode remains in effect until you change it or until you re-IPL the system – in which case, the mode reverts to the continuous mode (default mode) of operation. The LOGOFF command has no effect on the print mode. You can make a print mode selection or change modes any time before the output writer begins printing the file. However, once printing begins, any change in printing mode does not occur until the next file is printed. If all your printing is in the same mode, you can select a mode immediately after you enter your LOGON command or you can include the mode selection as part of your execution profile. If you alternate between printing modes, you must select a mode before responding to the forms mount message.

The following guidelines should be considered before attempting to use the page mode feature:

1. The printer connected to your workstation must be a local 0791 CQP type. Otherwise, any attempt to initiate page mode printing is ignored.
2. Each spool subfile should have a unique form name to easily identify the file for printing.
3. Both the test lines message and the page separator should be suppressed through your job control specifications.
4. The output writer does not support page mode when the spooled output is created, either where page change occurs via a skip code other than a home code or if it occurs by spacing to the new page. The program you use to create the spool subfile therefore must end with a skip to top of form.
5. System 3 RPG programs and any ANSI '74 COBOL programs using the LINAGE clause will cause misalignment and therefore are not to be used when printing in page mode.

4.4. OUTPUT TO DISKETTE: DATA SET LABEL MODE

Output spooling enables you to process output files to diskette in data set label mode. You use diskette files written in this mode as input to the input reader and to user-written programs. A given file may be contained on up to seven diskette volumes.

Your programmer uses the // DVC job control statement to spool the output file to be created to diskette. After the file is closed and queued in the spool file, it is written automatically according to the current operation mode (nonburst/burst) or at the discretion of your operator.

Your operator manually loads an output writer to process the diskette output by entering the output writer console command. Or, the operator may choose not to mount the specified diskette volume until a more convenient time.

4.5. OUTPUT TO TAPE, DISK, AND FORMAT LABEL DISKETTE: REDIRECTED OUTPUT

Redirected output is written to a tape, disk, or format label diskette for temporary storage, to be printed or punched later. When you redirect an output file, you release the spool file space it occupied while you save the file. Once you reintroduce the saved file, it is written directly by the output writer to the output device; therefore, no spool file space is used. Note that your system must be configured with dynamic buffer management to redirect and reintroduce output to and from a disk.

Each output file you redirect to a disk or diskette volume is automatically indexed in a temporary file called \$Y\$SPOOL. The output writer creates the \$Y\$SPOOL index on the specified volume for the first output file you redirect. Then it creates a sequentially numbered file called, in this case, \$SPOOL01. If you add a second output file, the output is indexed in \$Y\$SPOOL and written to \$SPOOL02. These temporary \$SPOOL files are scratched when you reintroduce the output they contain. The output writer scratches \$Y\$SPOOL when you reintroduce the last remaining \$SPOOL file on the volume.

Your programmer redirects an output file by using a parameter option with the // SPOOL jproc. The output is automatically redirected to the device specified, provided the operator approves the redirection. Your operator has two methods of redirecting output by using the output writer console command. The operator manually loads a printer or punch output writer in burst mode and specifies a tape, disk, or diskette address; or the operator directs an output writer, currently writing to a printer or card punch, to write its output to a tape, disk, or diskette.

Output is reintroduced when your operator uses the output writer command to manually load an output writer to accept input from the redirected output device. Before this, the redirected files are available for processing by a user-written program. The formats for tape, disk, and diskette redirected output files are provided in Appendixes A and B.

4.6. ACCUMULATING AND TRANSFERRING LOGS

Output spooling provides the capability of accumulating and saving job logs, workstation logs, and console logs after they are collected and printed. The logs collect and accumulate in an area of the spool file called LOG, as Figure 4-8 illustrates. You transfer these accumulated logs from the spool LOG file to a common SYSLOG file on tape or disk to make them available for further processing.

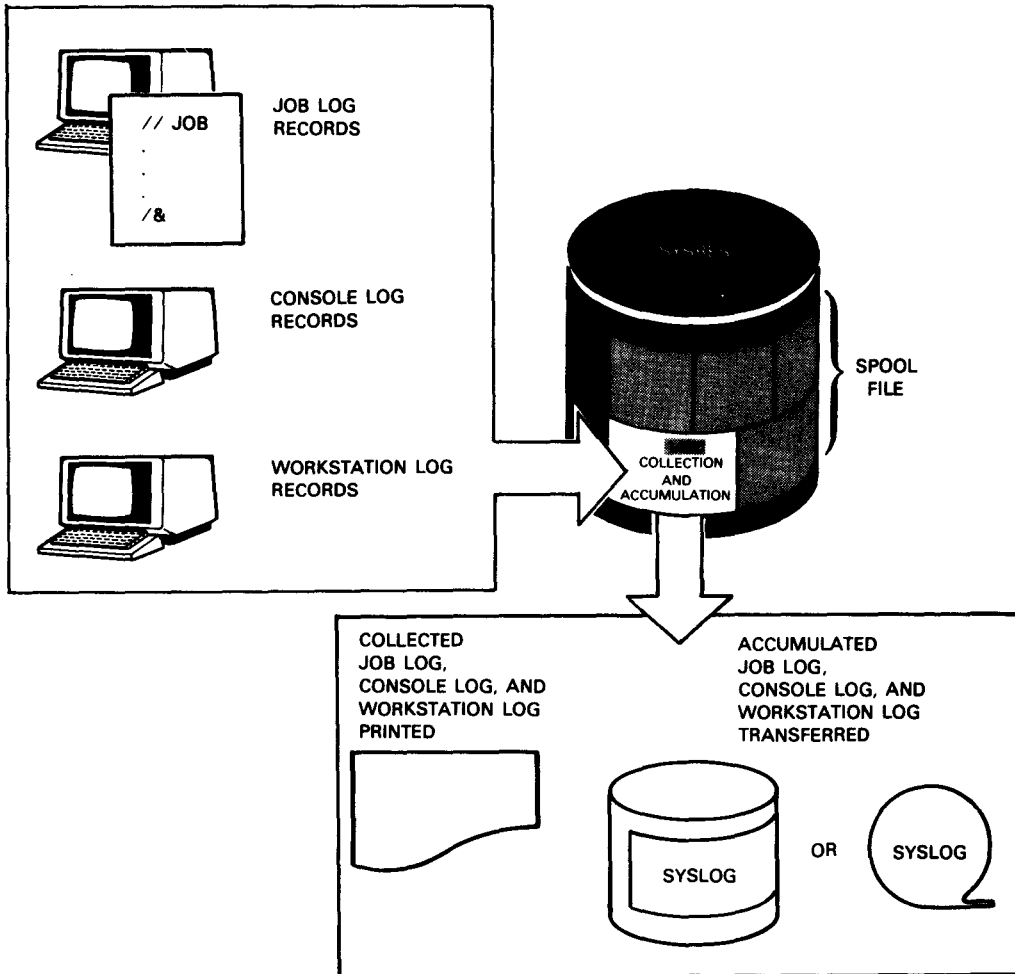


Figure 4-8. Using the Spool LOG File to Collect Logs for Printing and Accumulate Logs for Future Use

4.6.1. Controlling How Logs Are Processed in Your System

To accumulate logs, you establish certain SYSGEN spooling parameters and set other spooling conditions in your system during a working session. Earlier we described job, console, and workstation log printing. Now we're going to use some examples to take a close look at how you establish and control log collection and printing, how you accumulate and transfer logs, and how the spool LOG file is used to accomplish this. Whenever we talk about workstation log W records, the discussion also includes the workstation log R records you generate when using a terminal as a workstation.

Our first example of controlling spool LOG file input and output is presented in Figure 4-9, where we see the job log within the spool LOG file.

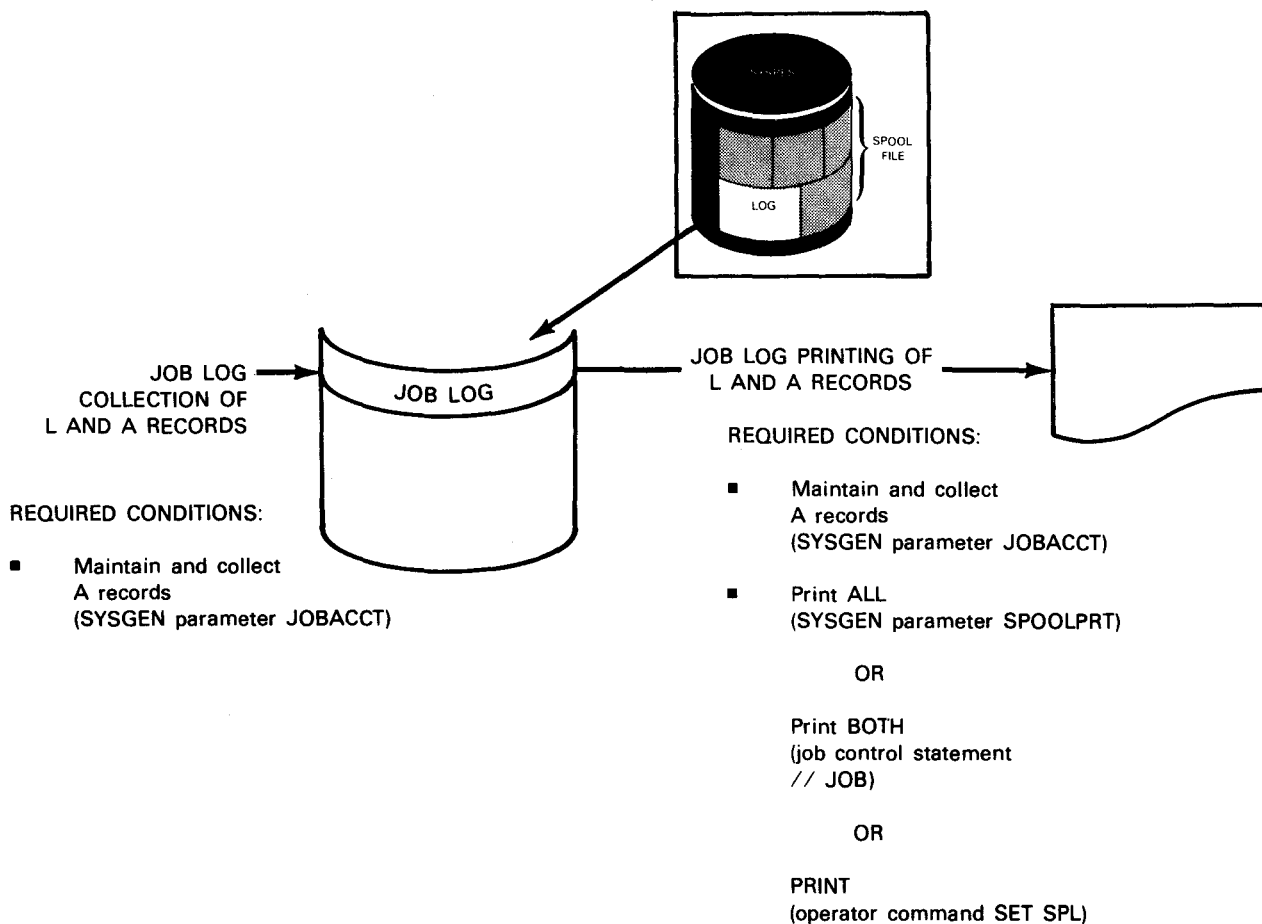


Figure 4-9. Spooling Conditions for Collecting and Printing the Job Log

Figure 4-9 shows job log records collecting in the job log so they can be printed when the job terminates. In this example, both L and A records are being collected. Spooling automatically collects L records, so the only required condition we had to establish was including the SYSGEN parameter to collect A records.

After we establish L and A record collection in the example, we have three ways to print the records. We can establish job log printing at SYSGEN, include a job control statement to print that job's log (overrides SYSGEN), or enter the operator command to permit job and workstation log printing (overrides SYSGEN and job control). These conditions print the L and A records shown in our example. However, you can establish other conditions to tailor job log printing as we described in 4.2.3.

The spool LOG file pictured in Figure 4-10 shows the console log with the job log.

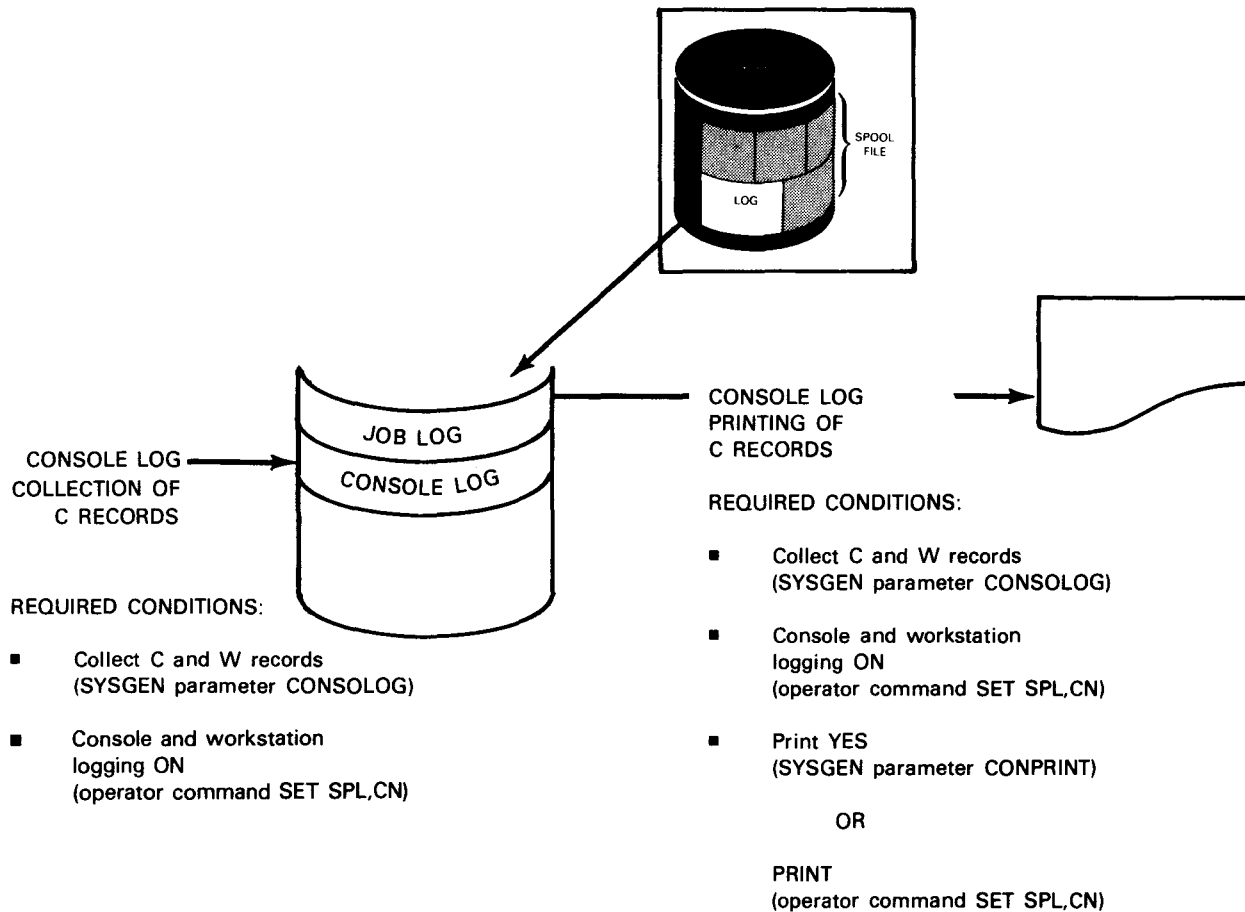


Figure 4-10. Spooling Conditions for Collecting and Printing the Console Log

The console log C records collect in the console log until they are breakpointed and printed. To collect the records in this example, we have established C and W record collecting during SYSGEN and made certain that console and workstation logging is turned on in the system. With these conditions, we include either the SYSGEN parameter for printing C records or the operator command to override SYSGEN and permit C record printing. Our example's spooling system is now set to print the breakpointed console log.

In Figure 4-11, we have added the workstation log to the console log and job log in the spool LOG file.

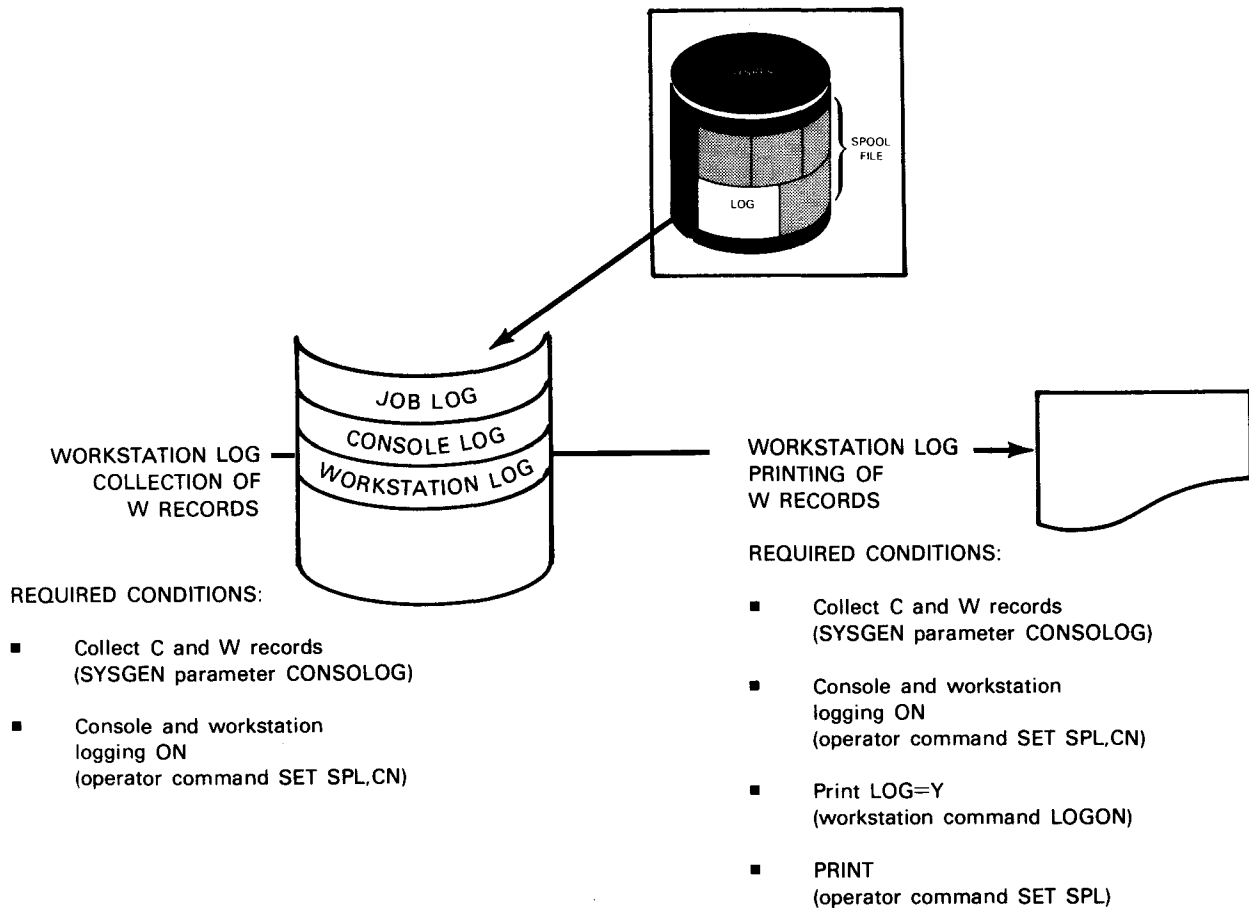


Figure 4-11. Spooling Conditions for Collecting and Printing the Workstation Log

The workstation log W records collect in the workstation log so they can be printed when the session ends or the log is breakpointed. The conditions we established to collect the C records in Figure 4-10 also establish the W record collection illustrated here. That is, we have included the SYSGEN parameter to collect C and W records and made sure that console and workstation logging is turned on. (For workstation logs, remember that an R replaces the W to identify records generated by a terminal that functions as a workstation.)

Once conditions are established for collecting W records, printing is established by the workstation user's LOGON command and the operator command to permit job and workstation log printing. In our example, the workstation log will be printed when a LOGOFF or BRKPT LOG command is entered from the workstation.

These last three examples have presented the spooling conditions you establish to collect and print job, console, and workstation logs. In the next two examples, we discuss how you use the spool LOG file to accumulate these logs for future use. Notice that the conditions required for collecting the logs are also required for accumulating them, while log printing is a separate option that doesn't affect log accumulation.

Let's look at Figure 4-12, where we see job and workstation logs grouped together as the accumulated job and workstation log.

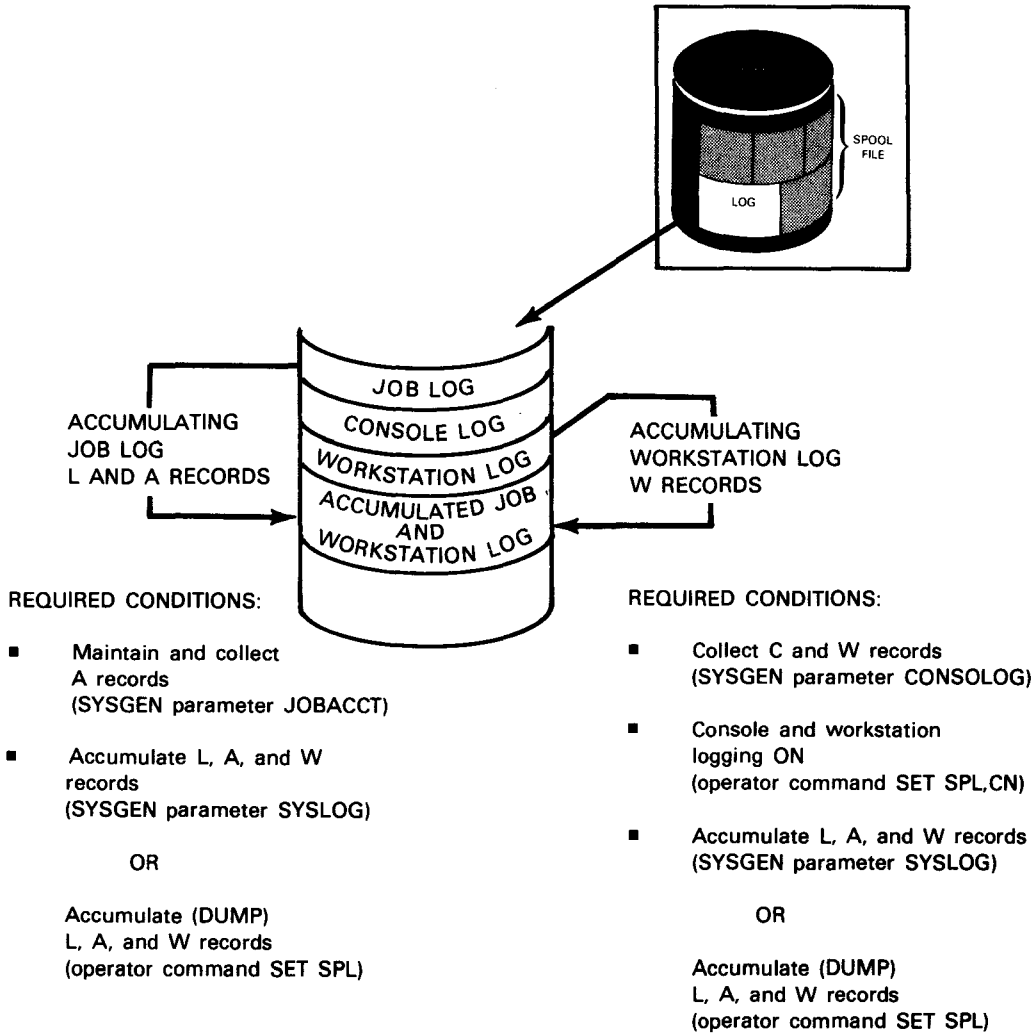


Figure 4-12. Spooling Conditions for Accumulating the Job Log and Workstation Log

Job logs and workstation logs are accumulated as a single log – the accumulated job and workstation log – in the spool LOG file; however, the conditions required to accumulate them are different. For the job log L and A records, the SYSGEN parameter to collect A records has been established (remember that the L records are collected automatically when you include spooling in your system). In addition, we either establish the SYSGEN parameter to accumulate L, A, and W records or enter the operator command to override SYSGEN and accumulate them.

For the workstation log records, we have established C and W record collecting during SYSGEN and made certain that console and workstation logging is turned on in the system. Once these conditions are established, your workstation logs are accumulated because the remaining required conditions are present in the example. That is, the conditions already established to accumulate job logs also accumulate the workstation logs. The job log and workstation log records accumulate in LOG until we transfer them to the SYSLOG file on tape or disk.

When you create the accumulated job and workstation log, you have the additional option of creating a separate identifier for a specific job's log in the accumulated log. You use this option when your system is configured with the general editor (EDT). Your programmer uses the // OPTION job control statement to create the identifier so that the job's log can be located upon request. By creating a log identifier with a job you initiate, you are able to check the accumulated log at a later time to see whether the job's log is there. If it is, you know that the job has run and terminated. The capability to see a job's log in the accumulated file is available by using the general editor.

Figure 4-13 illustrates grouping the console logs as the accumulated console log in the spool LOG file.

In our previous examples, we have established during SYSGEN the required condition for C and W record collection and made certain that console and workstation logging is turned on in the system. To accumulate the console log C records, we also include the appropriate SYSGEN parameter or enter the operator command to override SYSGEN and accumulate them. Either way, the console log records accumulate for us until we transfer them to SYSLOG.

A special system service program (SL\$LOG), using a job control stream (DUMPLOG or DUMPLOGT) supplied by Sperry, allows you to transfer (dump) the accumulated job and workstation log, and the accumulated console log, into the SYSLOG file. SL\$LOG cannot transfer both of your accumulated logs together as part of the same job. Usually, you first execute SL\$LOG to dump the accumulated job and workstation log so the job accounting records can be used as accounting report input. Next, you produce the accounting report by using the job accounting program supplied by Sperry. (We discuss how you produce this report in Section 8.) Then, you execute SL\$LOG to dump the accumulated console log and execute your own program to process it. After you transfer your accumulated logs, SL\$LOG deletes them from the spool LOG file.

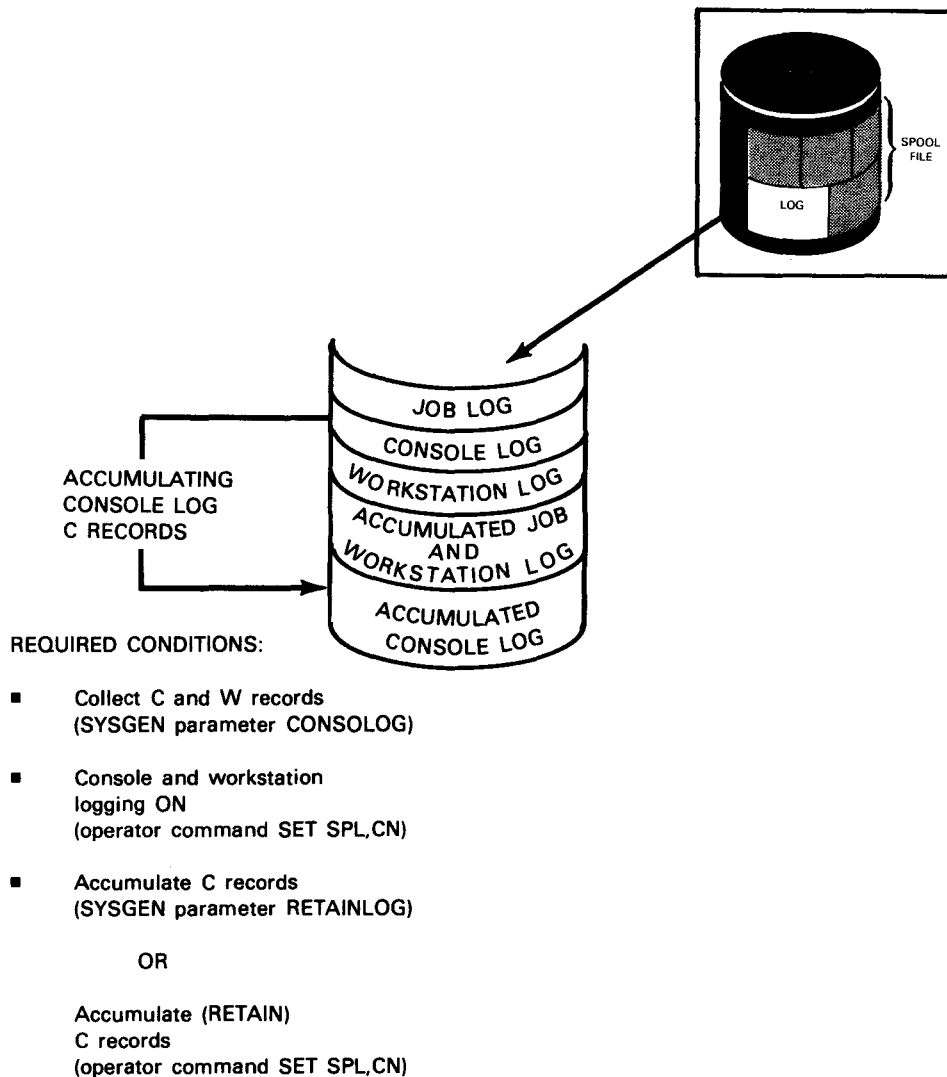


Figure 4-13. Spooling Conditions for Accumulating the Console Log

4.6.2. Transferring the Accumulated Job and Workstation Log for Future Processing

Your operator transfers the accumulated job and workstation log by executing the SL\$LOG program, using the RUN command with the DUMPLOG or DUMPLOGT job control stream (supplied by Sperry) and its parameter options. The job control stream calls a jproc that initiates the execution of SL\$LOG according to the options specified. These parameter options allow the operator to transfer log (L) and workstation (W and R) records only, accounting (A) records only, or all records in the accumulated log.

The job log's dump records are ignored, as they serve no purpose from a bookkeeping standpoint. In addition, the DUMPLOG parameter options allow your operator to specify whether the SYSLOG file on disk is to be established, added to, or *reinitialized* (effectively erased) when the accumulated log records are transferred. The DUMPLOGT parameter options allow your operator to specify the volume serial number of the tape to be used for the SYSLOG file and whether or not checkpoint records are to appear after each processed job and workstation log.

SL\$LOG is executed according to the parameter options your operator selects with DUMPLOG and DUMPLOGT, or according to the preset default condition for each parameter. With the exception of the tape or disk file being named SYSLOG, all of these default settings can be changed. You can write and file your own job control stream to execute SL\$LOG under your own processing conditions. To do this, you need only change the jproc call statement used by DUMPLOG or DUMPLOGT and include the options you want. SL\$LOG tape and disk files; the job log, workstation log, and console log; and the formats of the DUMPLOG and DUMPLOGT job control streams and jproc call statements are provided in Section 8.

4.6.3. Transferring the Accumulated Console Log for Future Processing

Your operator executes the SL\$LOG program to transfer the accumulated console log to the SYSLOG file. (Remember that you cannot dump both accumulated logs with a single execution of SL\$LOG.) The operator enters the RUN command with the DUMPLOG or DUMPLOGT job control stream (supplied by Sperry) and parameter options similar to those used for the accumulated job and workstation log. Here again, the job control stream calls a jproc to execute SL\$LOG according to the options specified. The DUMPLOG parameter options allow your operator to establish a SYSLOG file on disk, add more console log records to it, or reinitialize the SYSLOG disk file and transfer a newly accumulated console log file into it. Because the DUMPLOG parameter options permit the SYSLOG file to be reinitialized, we recommend that you process the transferred accumulated job and workstation log before you use SL\$LOG to transfer the accumulated console log. In this manner, your job logs are used to produce the accounting report and can be subsequently erased from SYSLOG without consequence.

Your operator uses the DUMPLOGT parameter options to specify the volume serial number of the tape to be used for the SYSLOG file and whether or not checkpoint records are to appear after each breakpointed (printed) session segment of the console log.

The SL\$LOG program transfers console logs in the same manner as we described for transferring job and workstation logs. Your operator either executes SL\$LOG by using the preset default condition for each parameter option or selects one of the available alternate options. You can change these default settings as we described. The information you'll need to do this is included in Section 8.



5. General Spooling Control

5.1. OVERVIEW

So far, we have discussed how input spooling and output spooling functions enable you to control files as they are spooled in and out of the spool file. In some cases, your operator changes the spooling environment for your entire system, via SET SPL commands, to control file processing. In addition to these capabilities, your operator and, with some restrictions, your programmer can control files residing in the spool file.

Spooling provides general console and workstation commands to display, hold, release, and delete spooled files, according to parameter options entered with the command. These options tailor the command to control all files, selected groups of files (such as all input files or all printer files), selected types of files (such as all files with a specific account number or form name, or designated for a specific device type), or a specific file in the spool file. Your operator uses any parameter options available with the command, whereas your programmer uses only those parameter options that control files associated with the programmer's identification. In addition, your programmer can use a job control statement to hold a job's output files until they are released via a command entry.

5.2. DISPLAYING THE STATUS OF SPOOLED FILES

Spooling enables you to obtain status information from the spool file about either a file that is being created or a completed file. The requested information is sent to the console or workstation in an initial display and an optional further display of more detailed information. The initial display gives you a list of the number of active files or lists the completed files (queued, on hold, and in progress) that meet your requirements. A file in progress is either being processed by the output writer or, for an input file, by a program.

Your operator enters parameter options with the spooling DISPLAY command to specify which files are to be included in the initial console display. After the initial display, however, your operator responds to the prompting messages that appear on the screen to select and tailor further console displays. You use the tailoring response options to display further information about some or all of the requested files in either a detailed or abbreviated format. You can, for example, display all completed print files that require a form named ABC and a 48-character business print cartridge. Or you may want to see the status of your job's punch output files - how many are queued in the spool file and whether an output writer is currently processing one of them. Your operator can also display status information about the current console log. The number of console log lines that have collected in the spool LOG file are displayed on the console screen with this option. (The display of workstation log information we describe for your programmer is not available to your operator.)

Your programmer uses the DISPLAY command in a similar manner to display file status and workstation log information on the workstation screen. Active or completed file information or the number of lines on a workstation screen used during the current session can be displayed at a workstation; a display of console log information is not permitted. Your programmer uses the DISPLAY command, for example, to see the status of the print output files to be created by a job initiated earlier from the workstation. First, your programmer displays information for the files currently being created by the programmer's job. This shows whether the job is currently running and how far along it is in creating output. Next, the job's completed print file information can be displayed to show which files are queued or being printed. If, in this example, no files (active or completed) are found for the job, your programmer displays the LOG file to check for the job's log. The display of the job log information means that the job has run and terminated. (This log information is available only when the job's log is in a hold state. We discuss holding spooled files in the following paragraphs.)

5.3. HOLDING SPOOLED FILES

Spooling enables you to place spooled files in a hold state. Once in the hold state, the files are unavailable for processing by the output writer or, for input files and logs from locally submitted jobs, by a program. You hold all active (being created) and completed files, or hold a selected group of these files (such as all printer or punch files). And, for completed files, you may additionally select a file group and hold a particular type of file (such as all print files with a specific account number) or hold a specific file. Remember, the files you hold have never been processed. This contrasts with the output files you retain, which are processed first with a copy retained in a hold state after processing. Once the file is in a hold state in either case, it cannot be processed until you choose to release it. (We discuss releasing held files in 5.4.)

You place files on hold to temporarily remove them from processing – perhaps to perform an intermediate action, permit a similar short file to be processed first, or allow more similar-type files to be created by a second job. Don't confuse this with your option to temporarily store output on tape, disk, or diskette by redirecting it. For example, we previously said you might want to use the DISPLAY command to see all completed print files that require a form named ABC and a 48-character business print cartridge. To expand this example and illustrate when you want to hold files, let's say that you are going to run several jobs that create 48-character business print output for ABC forms. By placing all files that meet these specifications on hold as they are closed and queued, you can wait until you run all of the jobs and then release their held output files for printing. While the files are held, the output writer won't try to process the first queued file and request that your operator mount the ABC form, or ask whether or not test lines should be printed.

After you run the last job, you display the output files to see whether all of the files are completed and on hold. When they are, and you are ready to do this special printing, you release the files to permit the output writer to process them on a first-in, first-out basis (burst mode).

As another example, you might hold a file that your test job creates so that you can display and check it. You then release the file or, as an alternative, delete it from the spool file. (We discuss deleting spooled files in 5.5.)

Your operator enters the spooling HOLD command or the output writer command (RP/PR/PU/PD) to place files on hold. HOLD is used to place any files being created as well as any queued files in a hold state, according to the parameter options included with the command. The active files are placed on hold when they are closed and queued. Your operator uses the output writer command to hold an output file that is currently being processed. In this case, the remaining unprocessed records from the output file are "requeued", then placed on hold in the spool file. Your programmer enters the HOLD command or output writer command (RP/PR/PU) at a workstation or uses the // SPOOL jproc to hold queued output files created under the programmer's identification.

5.4. RELEASING SPOOLED FILES

Spooling enables you to release spooled files that are being held by a HOLD or RP/PR/PU/PD command or a // SPOOL jproc. When you release a queued file from its hold state, you automatically call in an output writer in burst mode to process it. You release just one queued file from a group of files on hold, or you select file groups and file types to be released. ←

When, for example, a hold has been placed on all print files – for both files already queued and files to be subsequently created and queued – you can release at a later time only those print files currently being created, those queued print files created by a particular job, or only a specific print file.

You release files in a manner similar to the way you hold them. Your operator enters parameter options with the spooling **BEGIN** command to specify which files are to be released. The operator uses **BEGIN** to remove the hold state from a queued group of files, from a group of active files designated to be held when they are queued, or from a group of active and queued files. Your programmer has similar options to release those queued files created under the programmer's identification.

Referencing our previous example concerning the 48-character business print output files to be printed on ABC forms (see 5.2 on displaying files and 5.3 on holding files), you release all queued files with these characteristics by entering the **BEGIN** command and appropriate parameters. The output writer is loaded automatically to begin processing files. Similarly, you use **BEGIN** to release a held test file after you have checked it.

5.5. DELETING SPOOLED FILES

Spooling also enables you to delete spooled files from the spool file. You delete completed queued files waiting to be processed or completed files currently on hold. Files that are currently being created and completed files being processed by the output writer cannot be deleted.

We previously discussed how you might want to hold the output file created by your test job so that you can display and check it. Afterwards, you release the file for processing or, when appropriate, you delete it. Your operator and programmer delete completed files by using the **DELETE** spooling command. The parameter options they include with the command determine which files are deleted. For example, all queued printer files associated with a specific job or all queued input files can be removed from the spool file by using **DELETE**.

Your programmer and operator can also delete input files that were retained after a job processed them. If a retained input file is a diskette file, your programmer and operator can delete the file by using the **DELETE** command or the **DELETE** option of the **IN** command. If the retained input file is a card or tape file, your operator can delete the file by using the **DELETE** command or your programmer can do this by using the **DELETE** parameter of the **// DATA** job control statement.

6. Remote Spooling

6.1. OVERVIEW

In any discussion of input and output spooling, it is important to establish whether a local or remote location is sending or receiving the jobs and data. So far, we know how input readers read locally entered input and how output writers write output to local devices. Remote locations use the spooling system in a different manner.

Remote locations spool input and output by two methods, using the *remote batch processor* or *distributed data processing (DDP)*. The remote batch processor spools input (jobs or data) from punched cards and output to a printer or card punch. Disk, diskette, and tape files cannot be spooled to and from a remote site by using the remote batch processor.

When a remote site spools input by using distributed data processing, the input is always a disk file. DDP spools jobs when you want to run them on another system. The input job to be run can be filed in your system or prefiled on another system's disk (usually a library file). DDP also spools input jobs or data when you send them to be filed on another system's disk and the disk device isn't available. In this case, you send the job or data to the other system by the DDP indirect method so that it is placed in the spool file until it can be filed. When the device becomes available, DDP files the job or data into its designated file. The spooled output from a DDP job, like that using the remote batch processor, must be printed or punched.

Both the remote batch processor and distributed data processing use communications lines to handle all input from and output to remote sites. And, your operator controls remote batch and DDP input and output files by using the spooling commands we described in Section 5. There are other similarities between remote batch and distributed data processing spooling that we'll discuss later on. Our discussion here concerns how spooling relates to remote batch and distributed data processing. For further information on the remote batch processor and ICAM, refer to the ICAM utilities user guide; for information on DDP, refer to the distributed data processing concepts and facilities manual.

Now let's see how spooling with the remote batch processor works.

6.2. REMOTE SPOOLING USING THE REMOTE BATCH PROCESSOR

To spool remote input and output by using the remote batch processor, you must generate your system with:

- input and output spooling, and with remote batch processing included; and
- a communications network, which includes a type of communications adapter.

Basic spooling element relationships (earlier shown in Figure 2-1) are expanded in Figure 6-1 to include remote devices used by the remote batch processor.

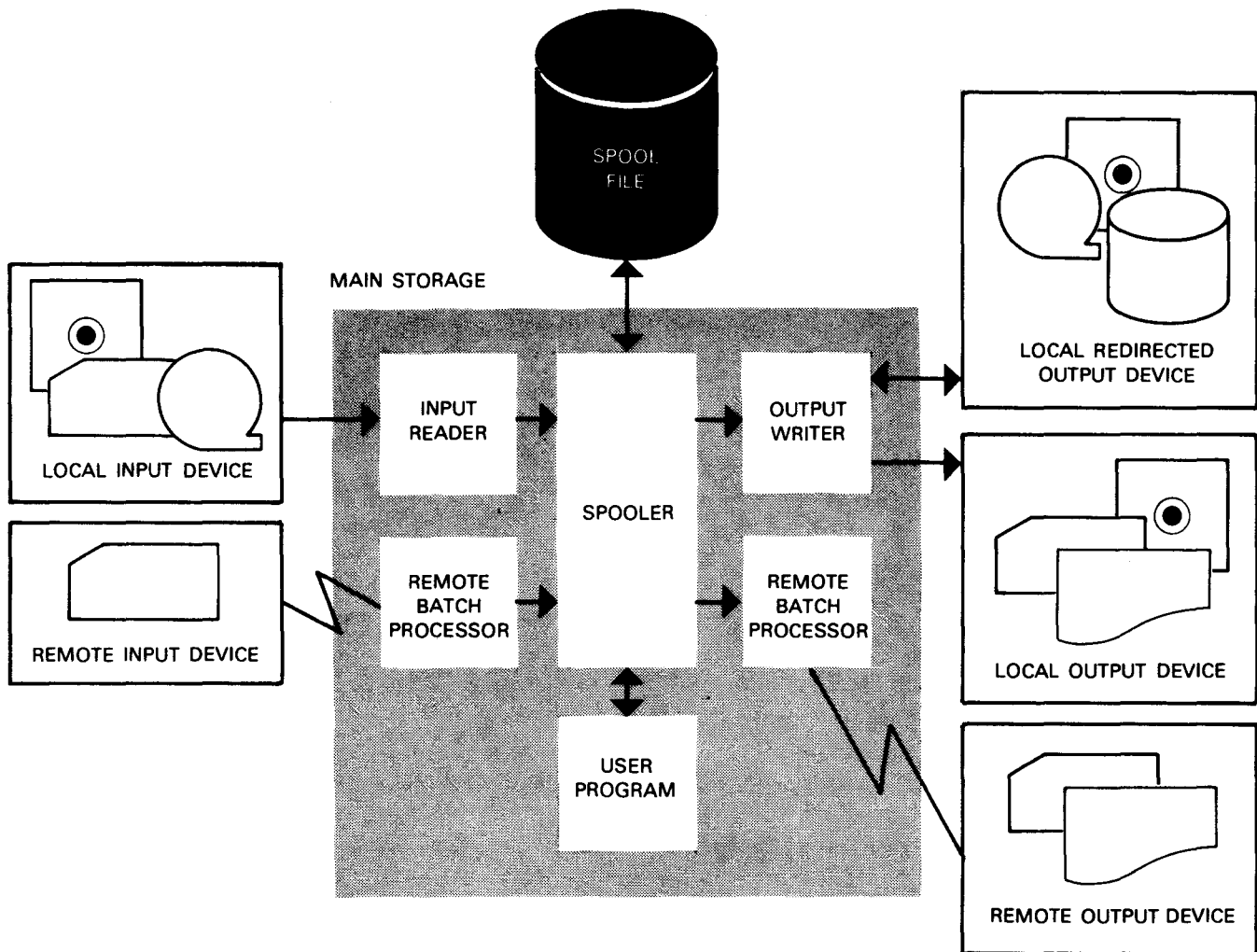


Figure 6-1. Spooling Element Relationships with Remote Batch Devices Included

As Figure 6-1 shows, the remote batch processor uses a communications line to read input from a remote card reader, then calls the spooler to write the remote batch input to the spool file. The remote batch processor provides the same function as an input reader for the punched card input. The figure also shows the remote batch processor writing output from the spooler to a remote printer or card punch across a communications line. When the communications network is automatically loaded for sending output to the remote batch processor, your operator is given the option of overriding the communications network name (pointer) selected at SYSGEN by using the SET IC command.

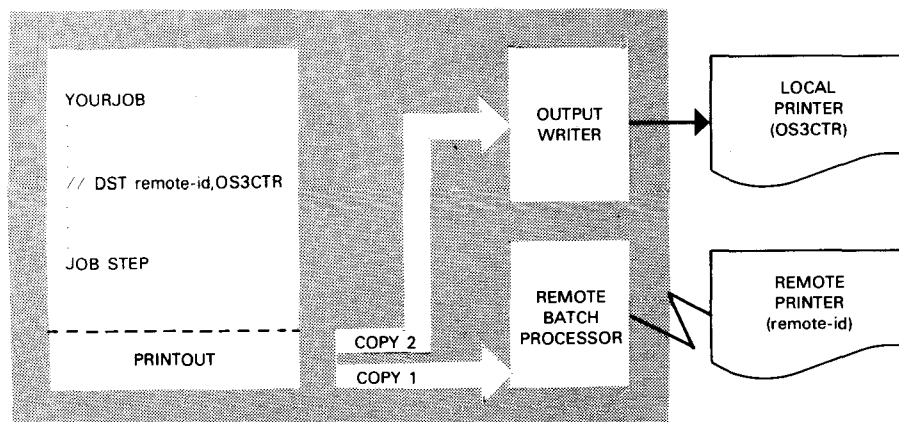
Once the remote batch processor input and output is in the spool file, you control and tailor it in the same way you do with locally spooled input and output. Your operator is able to display remote batch input and output file information in the spool file, as well as delete, hold, and release spooled output files for remote batch sites. The operator also controls a remote batch job in the same manner as a locally entered job.

Should a remote batch user want the operator to exercise control over a remote batch job, the user transmits a */RMSG remote batch command* on a card to the operator and specifies the control option desired. The ICAM utilities user guide contains a description of the */RMSG* command and parameter options.

When you enter a job from a remote location by using the remote batch processor, you can direct the job's output to multiple destinations. Your programmer uses a *// DST* job control statement to write a copy of the output file to the destinations included with the statement. When the statement is not included in the remote batch job, its output is automatically written to an appropriate device at the local site. When your programmer includes the *// DST* card, it must contain the OS3CTR (the local site) destination to have an output file copy sent there. A header for the output file is written with the file to each destination specified.

The log for any job, whether remotely entered or initiated locally, is always printed at the site that submitted the job. Log accumulation and transferring, however, remains a local function; that is, logs are accumulated for accounting and bookkeeping purposes at the site where the job was run.

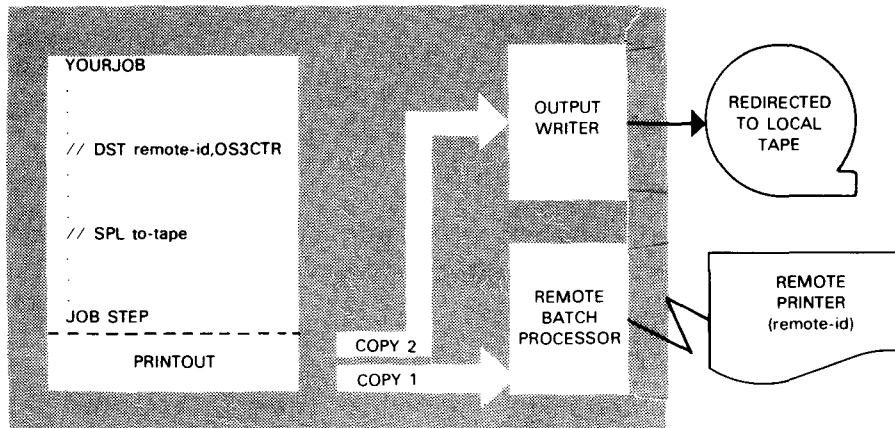
When you enter a remote batch job and identify remote and local output destinations with a *// DST* statement, the remote batch processor processes the remote file copy and the output writer processes the local output.



The output writer automatically writes the local copy to the printer. However, the remote batch processor writes job output automatically only to the remote site that submitted the job, provided that the site is still connected to the communications line. If the line is not connected, the remote site must dial up (connect) the line and request the output file. When a locally entered job produces output for a remote destination, the remote batch processor writes it upon request. When remote batch output files have collected in the spool file, the remote user may request all output files or only those with a specific user-id or job name. You use the /RMSG command (described in the ICAM utilities user guide) for this function.

As an option, your operator can manually load or direct an output writer to locally print or punch files designated for a remote destination. You use this option, for example, when the remote site is unable to receive the output.

Let's take a look at another example of remote batch and local spooling. Suppose YOURJOB includes a JOB STEP that creates an output file for printing. The job already has a // DST statement included with a printer remote-id and the OS3CTR destinations. For this job run only, your programmer includes a // SPOOL jproc to redirect the local output to a tape. The // SPOOL jproc overrides the // DST destinations just for local spooling. The remote batch processor ignores the // SPOOL and writes the output to the remote-id when requested, while the output writer redirects its output file copy to a local tape device.



As an alternative, your operator can use the PR or PU command to redirect the local output created by a remotely entered job.

6.3. REMOTE SPOOLING USING DISTRIBUTED DATA PROCESSING

To spool remote input and output by using distributed data processing, you must generate your system with:

- input and output spooling with DDP included (automatically includes remote batch spooling);
- a communications network, which includes a type of communications adapter; and
- interactive services.

The basic spooling element relationships shown earlier in Figure 2-1 are expanded in Figure 6-2 to include the distributed data processing spooling relationships.

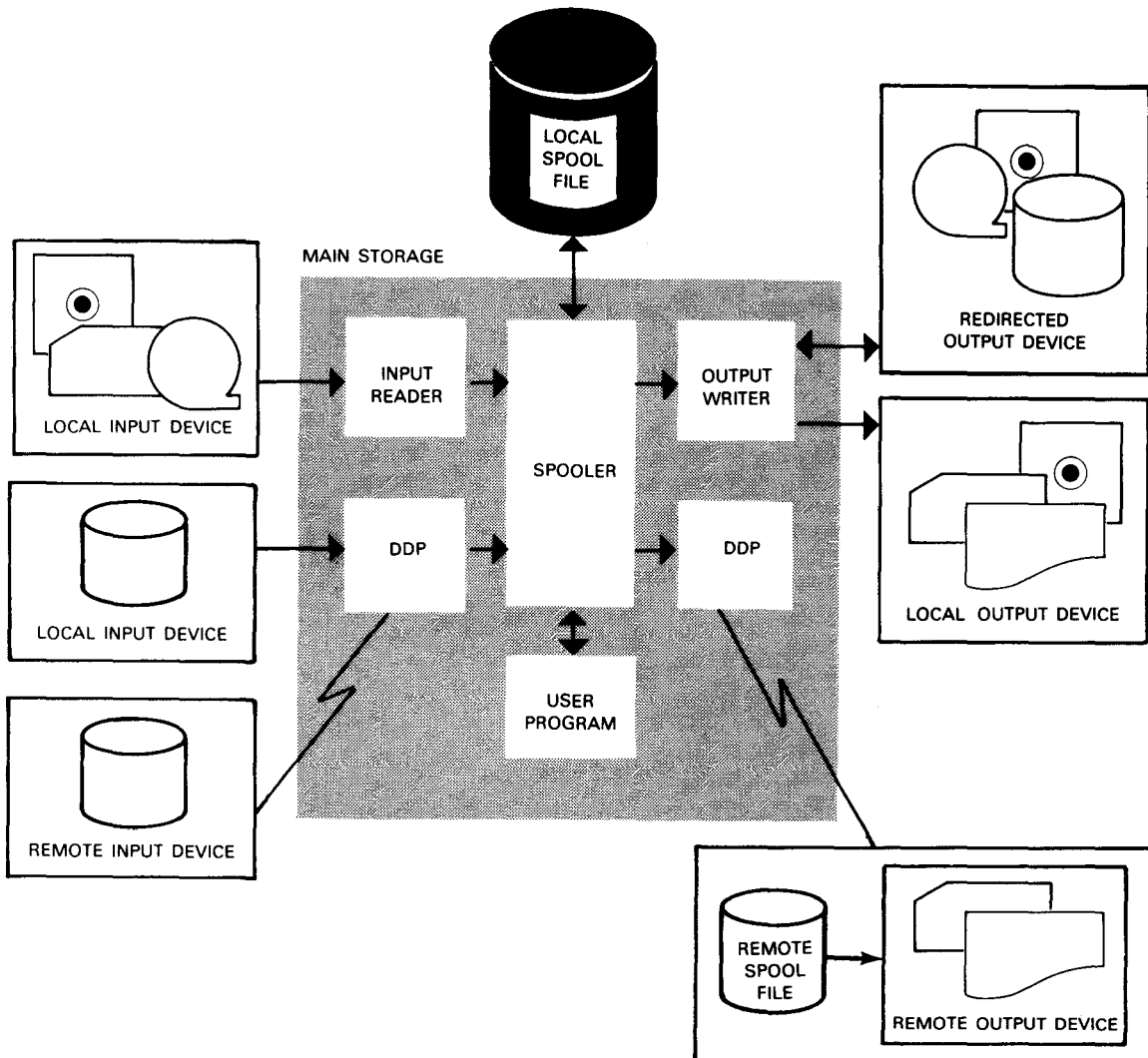
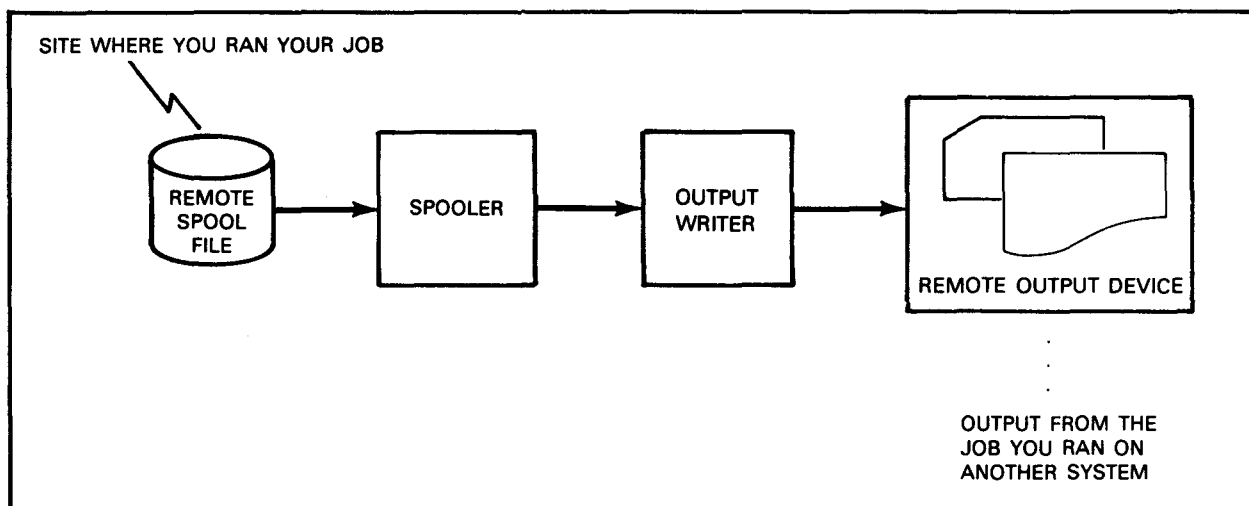


Figure 6-2. Spooling Element Relationships with Distributed Data Processing Devices Included

Figure 6-2 shows input to distributed data processing from two sources, either a local disk file or a remote disk file. When you submit a job from your system to be run on another system, DDP uses a communications line to read your job (the remote input). When you submit a job to be run on another system that is prefiled on that system, DDP reads your prefiled job (the local input) and no communications line is required. After reading the input, DDP calls the spooler to write it to the spool file. Like the remote batch processor, DDP provides the input reader function for remote spooling.

On occasions when you copy your jobs or data to another system's disk file and you specify the indirect method, DDP reads your remote input and calls the spooler to write it to the local spool file. In this case, DDP writes your input job or data from the spooler to the designated local disk file (provided the disk device is available).

Looking again at Figure 6-2, we see DDP writing output across a communications line from the spooler to your system's spool file (the remote spool file) so it can be printed or punched for you. Your system's output writer writes the output from the spooler to your printer or punch, according to the spooling conditions currently in effect in your system.



Once the DDP input and output is in a system's spool file - either to be run or filed from another system's spool file, or to be printed from your system's spool file - the system's operator controls and tailors it in the same way as with local and remote batch input and output. The operator can display, delete, hold, and release DDP output files, as well as control DDP jobs. The log for your job is always printed at your site. Log accumulation and transferring, however, remains a local function. That is, logs are accumulated for accounting and bookkeeping purposes at the site where the job was run.

7. Spooling Initialization and File Recovery

7.1. OVERVIEW

Spooling initialization sets up the spool file for spooling. It either creates a new spool file or recovers an existing spool file, according to your requirements. The initial program load (IPL) procedure, performed by your operator, initializes the resident portion of OS/3 (the supervisor) and includes the initialization of your spooling system. The spooling options you select during SYSGEN are established when the supervisor is loaded; however, your operator is given the opportunity to change some of the options during the procedure. These options are recorded in the system spool control table, along with the required internal pointers (cross-references), each time an IPL is performed.

Spooling initialization locates and verifies the spool file disk volume or volumes you established at SYSGEN. The spool file is automatically located on the volume identified as SYSRES unless you specify another device at SYSGEN. During the IPL, your operator has the option to override this setting. The supervisor displays a message indicating each established spooling volume to the operator. The message identifies the SYSGEN-specified volume as the default and prompts your operator to change the specification, if desired, by entering a different device address. Because up to eight volumes may hold the spool file, your operator may elect to use fewer disk volumes than the maximum identified or to use different spooling volumes from those you specified at SYSGEN. You also specify the spool file size (in cylinders) at SYSGEN; otherwise, initialization establishes the default value of 50.

In addition, initialization assigns space (allocates) if necessary and reestablishes cross-references (initializes) for the spool file when no recovery is required (*cold start*), or it recovers the spool file according to the option entered at SYSGEN (*warm start*). The IPL procedure permits your operator to select a cold start, which is normally established as the default option, or to specify a warm start recovery level for specific types of files. In the following paragraphs, we describe the options spooling initialization provides to allocate and initialize, or to recover the spool file.

7.2. COLD START

You select a cold start when the spool file is empty; therefore, no file recovery is needed. The spool file is empty because you processed spooled input and output files before you turned off your system in an orderly manner. When you use a cold start, the spool file disk space is allocated if necessary, then initialized on up to eight volumes according to your SYSGEN selection. During IPL, your operator is able to establish less than the number of volumes you select at SYSGEN.

During the IPL procedure, your operator is also permitted to retain, delete, and place on hold any jobs previously filed in the job queue. When your operator recovers the queue by retaining or holding the jobs, the spooled job log files are automatically recovered, regardless of the type of spooling initialization the operator selects. This means that even when your operator enters a cold start (no file recovery), the job log files are recovered if the job queue is recovered.

7.3. WARM START

Warm start spooling initialization assumes that the spool file is online (system shutdown has not occurred), and that it contains all of the complete and incomplete files previously resident, plus all information and cross-references required to re-create the system spool control table. You verify and recover the contents of the spool file according to the warm start recovery level entered by your operator, or by the default.

You specify a warm start to recover files at one of these levels:

- Recover completed files only.

All closed and queued files in the spool file, including logs, are recovered.

- Recover completed log files only.

All closed and queued logs in the spool file are recovered.

- Recover all completed files plus all incomplete files.

All files that were incomplete (not queued) when the system was shut down are recovered with the completed files. Console logs and workstation logs are included with these incomplete files; however, any console or workstation log records that were not copied from the spooling control and buffer space in main storage are not recovered.

- Recover all files (complete and incomplete) plus the complete console log.

With this level, called the *hot start*, the console log records that collected in the spooling control and buffer space are recovered along with those in the spool file. (Workstation log records that collected in spooling control and buffer space are not recovered, however.) You select the hot start method of recovery when your operator has taken a system dump of the previously loaded system. You use this recovery level only after a system crash has occurred. When there is a planned system shutdown and the spool file is to be recovered at a later time, your operator should breakpoint the console log and then select one of the other recovery levels.

When a warm start is specified, your operator must mount all spool file disk volumes that were identified during the IPL for the previous system session. After the warm start recovers the files in the spool file, your operator has the option to tailor how they are processed, via the output writer, as we described.



8. Producing Job Accounting Reports

8.1. SYSTEM LOG ACCUMULATION PROGRAM (SL\$LOG)

As we mentioned earlier (4.6), output spooling lets you save and accumulate job and workstation logs, and the console logs, in the spool LOG file. You use the system log accumulation program (SL\$LOG) to transfer the logs to your disk or tape file.

The SL\$LOG program copies the contents of the job and workstation log files or the console log files in the spool LOG file and transforms them into the format for input to the JOBLOG program. Any log file that has been deleted or placed on hold is not copied. Also, any file that has not been completely printed by the output writer at the time the SL\$LOG program is executed is not copied. In this case, however, such a file will be copied the next time the SL\$LOG program is executed, provided that the file has been printed.

When you have both the accumulated job and workstation log and the accumulated console log in LOG, you must execute SL\$LOG twice – once for each accumulated log. After the logs are transferred, SL\$LOG deletes them from the spool file.

Sperry provides you with the JOBLOG program, which produces a formatted accounting report using the job accounting (A) records. A record layout is provided for the console log (C) records, the job log (L) records, and the workstation log (W and R) records, enabling you to write a program producing your own report.

8.1.1. SL\$LOG Output Files

You use SL\$LOG to produce either a sequential disk file or a standard labelled tape containing 256-byte unblocked records. With either medium, the name (LFD) of the output file is SYSLOG.

When you're dumping to disk, you can use SL\$LOG to transfer the accumulated job and workstation log or the accumulated console log to a previously opened SYSLOG file. You must write your accounting and bookkeeping programs to use sequential access processing.

For tape output, you must mount a new tape with each SL\$LOG run. If you use the same tape, the previous log records are overwritten. Unless you specify otherwise when you run SL\$LOG, checkpoint records appear after each processed log subfile. In this case, your job accounting and bookkeeping programs should specify `CHKPTREC=YES` and `FILABL=STD` in the input file description.

8.1.2. SL\$LOG Job and Workstation Log Output

When SL\$LOG transfers your accumulated job and workstation log records to the SYSLOG file, you choose whether to have just the log and workstation records dumped, just the accounting records dumped, or all the accumulated records dumped. Byte 121 (print position 122) of the output identifies the type of record as follows:

A = accounting record

L = log record

W = workstation record (An R in this position identifies a record generated by a terminal functioning as a workstation.)

In addition, bytes 123-130 contain a time stamp in the format hh:mm:ss. If your system is configured without a timer, a sequential number replaces the time stamp. The time stamp information is not considered critical to the accounting records and is intended to be truncated if printed on a printer with 120 print positions.

8.1.2.1. Accounting Record Formats

Job accounting (A) records are 256 bytes long. In the first 132 bytes, SL\$LOG reproduces the image of various accounting records from the accumulated log. Then, in bytes 133-255, SL\$LOG appends information your job accounting programs can use as keys to sort the accounting records. If your system includes interactive services, interactive services usage records are included with the accumulated accounting records.

8.1.2.1.1. Accounting Data (Bytes 0-130)

Figure 8-1 shows the layout of bytes 0-130 of the accounting records on a printer format sheet. Note that the printer format's byte numbering begins at byte 1, thereby incrementing the record byte formats by 1. The byte format for these records is given in the following lists. Uppercase entries show what is actually printed in the field; entries in parentheses show the type of information.

1. Job Record

| <u>Bytes</u> | <u>Contents</u> |
|--------------|------------------|
| 0-3 | AC01 |
| 6-8 | JOB |
| 10-17 | (Job name) |
| 21-29 | ACCT. NO. |
| 31-34 | (Account number) |

| <u>Bytes</u> | <u>Contents</u> |
|--------------|-------------------------------|
| 41-56 | ASSIGNED MEMORY= |
| 57-64 | (Storage size in decimal) |
| 66-76 | BYTES (PLUS |
| 78-83 | (Size of prologue in decimal) |
| 85-98 | BYTE PROLOGUE) |
| 102-109 | (Run date in yy/mm/dd format) |
| 112-114 | JOB |
| 116 | # |
| 117-118 | (Job slot number) |
| 121 | A |
| 123-130 | (Time in hh:mm:ss format) |
| 133-255 | (See 8.1.2.1.2.) |

2. System Configuration Record

| <u>Bytes</u> | <u>Contents</u> |
|--------------|--------------------------------|
| 0-3 | AC02 |
| 6-10 | (Machine type) |
| 13-18 | (Supervisor name) |
| 21-22 | (Supervisor version) |
| 23 | . |
| 24 | (Supervisor revision level) |
| 25 | . |
| 26 | (Supervisor subrevision level) |
| 121 | A |
| 123-130 | (Time in hh:mm:ss format) |
| 133-255 | (See 8.1.2.1.2.) |

3. Job Step Record

| <u>Bytes</u> | <u>Contents</u> |
|--------------|--|
| 0-3 | AC11 |
| 6-11 | STEP # |
| 12-14 | (Step number) |
| 17 | (|
| 18-25 | (Step name) |
| 26-31 |) USED |
| 33-40 | (Storage used by job step in decimal) |
| 42-73 | BYTES ELAPSED WALL CLOCK TIME= |
| 74-85 | (Elapsed wall clock time in hh:mm:ss.mil format) |
| 90-105 | TOTAL SVC CALLS= |
| 106-113 | (Total SVC calls) |
| 121 | A |
| 123-130 | (Time in hh:mm:ss format) |
| 133-255 | (See 8.1.2.1.2.) |

4. Step CPU Record

| <u>Bytes</u> | <u>Contents</u> |
|--------------|--|
| 0-3 | AC12 |
| 10-19 | TERM CODE= |
| 20-22 | (Job step termination code) |
| 29-44 | SWITCH PRIORITY= |
| 45-46 | (Switch priority) |
| 50-62 | CPU TIME USED |
| 73 | = |
| 74-85 | (CPU time used by job step in hh:mm:ss.mil format) |

| <u>Bytes</u> | <u>Contents</u> |
|--------------|---------------------------|
| 90-105 | TRANSIENT CALLS= |
| 106-113 | (Total transient calls) |
| 121 | A |
| 123-130 | (Time in hh:mm:ss format) |
| 133-255 | (See 8.1.2.1.2.) |

5. Spool Record

| <u>Bytes</u> | <u>Contents</u> |
|--------------|---|
| 0-3 | <u>AC10</u> |
| 6-8 | LFD |
| 10 | - |
| 12-19 | LFD-name |
| 21 | ' |
| 22-30 | FORM NAME |
| 32 | - |
| 34-41 | (Form name) |
| 42 | ' |
| 44-49 | COPIES |
| 51 | - |
| 53-56 | (Number of copies) |
| 57 | ' |
| 58-64 | RECORDS (for data set label diskette output only) |
| 59-63 | { PAGES } { CARDS } |
| 65 | - |
| 67-74 | (<u>Number of pages, cards, or records</u>) |
| 75 | ' |

| <u>Bytes</u> | <u>Contents</u> |
|--------------|---------------------------|
| 77-80 | STEP |
| 82 | = |
| 83-85 | (Step number) |
| 121 | A |
| 123-130 | (Time in hh:mm:ss format) |
| 133-255 | (See 8.1.2.1.2.) |

6. UPSI Byte Setting Record

| <u>Bytes</u> | <u>Contents</u> |
|--------------|---------------------------|
| 0-3 | AC13 |
| 6-9 | UPSI |
| 11-17 | SETTING |
| 19 | X |
| 20 | . |
| 21-22 | (Hexadecimal value) |
| 23 | . |
| 121 | A |
| 123-130 | (Time in hh:mm:ss format) |
| 133-255 | (See 8.1.2.1.2.) |

7. Step Device Record

| <u>Bytes</u> | <u>Contents</u> |
|--------------|---------------------------|
| 0-3 | AC19 |
| 17-29 | DEVICE EXCP'S |
| 35-37 | (Device 1) |
| 38 | = |
| 39-46 | (EXCP-count for device 1) |

| <u>Bytes</u> | <u>Contents</u> |
|--------------|---------------------------|
| 49-51 | (Device 2) |
| 52 | = |
| 53-60 | (EXCP-count for device 2) |
| 63-65 | (Device 3) |
| 66 | = |
| 67-74 | (EXCP-count for device 3) |
| 77-79 | (Device 4) |
| 80 | = |
| 81-88 | (EXCP-count for device 4) |
| 91-93 | (Device 5) |
| 94 | = |
| 95-102 | (EXCP-count for device 5) |
| 121 | A |
| 123-130 | (Time in hh:mm:ss format) |
| 133-255 | (See 8.1.2.1.2.) |

8. Job Total Elapsed Wall Clock Time Record

| <u>Bytes</u> | <u>Contents</u> |
|--------------|--|
| 0-3 | AC21 |
| 6-15 | JOB TOTALS |
| 20-23 | USED |
| 25-32 | (Storage used in decimal) |
| 34-38 | BYTES |
| 44-73 | <u>TOTAL ELAPSED WALL CLOCK TIME=</u> |
| 74-85 | <u>(Elapsed time in hh:mm:ss.mil format)</u> |
| 90-109 | TOTAL JOB SVC CALLS= |

| <u>Bytes</u> | <u>Contents</u> |
|--------------|---------------------------|
| 110-117 | (Total SVC calls) |
| 121 | A |
| 123-130 | (Time in hh:mm:ss format) |
| 133-255 | (See 8.1.2.1.2.) |

9. Job Wall Clock Time of Steps Record

| <u>Bytes</u> | <u>Contents</u> |
|--------------|-------------------------------|
| 0-3 | AC22 |
| 44-73 | WALL CLOCK TIME OF ALL STEPS= |
| 74-85 | (Time in hh:mm:ss.mil format) |
| 90-109 | JOB TRANSIENT CALLS= |
| 110-117 | (Total transient calls) |
| 121 | A |
| 123-130 | (Time in hh:mm:ss format) |
| 133-255 | (See 8.1.2.1.2.) |

10. Job CPU Time Record

| <u>Bytes</u> | <u>Contents</u> |
|--------------|--------------------------------------|
| 0-3 | AC23 |
| 44-73 | <u>TOTAL CPU TIME OF ALL STEPS=</u> |
| 74-85 | <u>(Time in hh:mm:ss.mil format)</u> |
| 90-109 | TOTAL JOB EXCP'S= |
| 110-117 | (Total EXCP-count) |
| 121 | A |
| 123-130 | (Time in hh:mm:ss format) |
| 133-255 | (See 8.1.2.1.2.) |

11. Interactive Services Logon Record

| <u>Bytes</u> | <u>Contents</u> |
|--------------|-------------------------------|
| 0-3 | AC50 |
| 6-13 | USER-ID= |
| 14-19 | (User-id) |
| 23-30 | ACCT NO= |
| 31-38 | (Account number) |
| 42-49 | LOGON AT |
| 51-62 | (Time in hh:mm:ss:mil format) |
| 66-74 | LOGOFF AT |
| 76-87 | (Time in hh:mm:ss:mil format) |
| 91-102 | CONNECT TIME |
| 104-115 | (Time in hh:mm:ss:mil format) |
| 121 | A |
| 123-130 | (Time in hh:mm:ss format) |
| 133-255 | (See 8.1.2.1.4.) |

12. Interactive Services CPU Time Record

| <u>Bytes</u> | <u>Contents</u> |
|--------------|-------------------------------|
| 0-3 | AC51 |
| 6-19 | CPU TIME USED= |
| 20-31 | (Time in hh:mm:ss:mil format) |
| 35-48 | TASK PRIORITY= |
| 49-50 | (Task priority) |
| 54-58 | DATE= |
| 59-66 | (Date in yy/mm/dd format) |

| | |
|---------|---------------------------|
| 76-92 | NUMBER OF EXCP'S= |
| 93-100 | (Total EXCP'S) |
| 121 | A |
| 123-130 | (Time in hh:mm:ss format) |
| 133-255 | (See 8.1.2.1.4.) |

13. Interactive Services Workstation Record

| <u>Bytes</u> | <u>Contents</u> |
|--------------|---------------------------|
| 0-3 | AC52 |
| 6-15 | NUMBER OF: |
| 19-27 | COMMANDS= |
| 28-32 | (Total commands) |
| 34-48 | FILES ACCESSED= |
| 49-53 | (Total files accessed) |
| 56-66 | SVC CALLS= |
| 67-74 | (Total SVC calls) |
| 77-92 | TRANSIENT CALLS= |
| 93-100 | (Total transient calls) |
| 121 | A |
| 123-130 | (Time in hh:mm:ss format) |
| 133-255 | (See 8.1.2.1.4.) |

14. Interactive Services Device EXCP Record

| <u>Bytes</u> | <u>Contents</u> |
|--------------|-----------------|
| 0-3 | AC53 |
| 16-28 | DEVICE EXCP'S |
| 34-36 | (Device 1) |
| 37 | = |

| | |
|---------|---------------------------|
| 38-45 | (EXCP-count for device 1) |
| 48-50 | (Device 2) |
| 51 | = |
| 52-59 | (EXCP-count for device 2) |
| 62-64 | (Device 3) |
| 65 | = |
| 66-73 | (EXCP-count for device 3) |
| 76-78 | (Device 4) |
| 79 | = |
| 80-87 | (EXCP-count for device 4) |
| 90-92 | (Device 5) |
| 93 | = |
| 94-101 | (EXCP-count for device 5) |
| 121 | A |
| 123-130 | (Time in hh:mm:ss format) |
| 133-255 | (See 8.1.2.1.4.) |

8.1.2.1.2. Accounting Data Sorting Keys (Bytes 133-255)

The sorting keys begin in byte 133 and include information of the job name, account number, job number, and type of record.

The format is:

| <u>Bytes</u> | <u>Contents</u> |
|--------------|---------------------------|
| 133-140 | (Job name) |
| 141-144 | (Account number) |
| 145-148 | (Job number) |
| 149-152 | (Record sequence number) |
| 153-160 | (Date in yy/mm/dd format) |

| <u>Bytes</u> | <u>Contents</u> |
|--------------|---|
| 161-163 | (Job step number*) |
| 164 | (Type of record: X'01' Job record X'02' Job step record X'03' Step CPU time record X'04' Spool record X'05' Step device utilization record X'07' Job total elapsed wall clock time record X'08' Job wall clock time of steps record X'09' Job CPU time of steps record) |
| 165 | (Blank) |
| 166-255 | (Reserved) |

8.1.2.1.3. Interactive Services Sorting Keys (Bytes 133-255)

The sorting keys begin in byte 133 and include information such as the user's ID, account number, logon time, record type, and the date.

The format is:

| <u>Bytes</u> | <u>Contents</u> |
|--------------|---|
| 133-138 | (User's ID) |
| 141-144 | (Account number) |
| 145-148 | (Record type: IS1 Interactive services logon record IS2 Interactive services CPU time record IS3 Interactive services workstation record IS4 Interactive services device EXCP record) |
| 149-152 | (Sequence number) |

*For records AC21, AC22, and AC23, this field will contain X'FFFFFF' so that they appear as the last records after sorting.

| <u>Bytes</u> | <u>Contents</u> |
|--------------|---------------------------------|
| 153-160 | (LOGON time in hh:mm:ss format) |
| 164 | (IS identification (X'C9')) |
| 165 | I |
| 166-169 | (Date in mmdd format) |
| 170-175 | (User's ID) |
| 176-181 | (LOGON time in hh:mm format) |
| 182-255 | (Reserved) |

8.1.2.2. Log Record Format

For the job's log (L) records, SL\$LOG reproduces the image of each job's job control statements and system messages. As with the accounting records, SL\$LOG appends information, in bytes 133-152, to the end of each log record for use as sorting keys. These L records are available for use by your own programs.

The format of the log record is:

| <u>Bytes</u> | <u>Contents</u> |
|--------------|--|
| 0-120 | (Image of the job control statement or system message) |
| 121 | L |
| 123-130 | (Time in hh:mm:ss format) |
| 133-140 | (Job name) |
| 141-144 | (Account number) |
| 145-148 | (Job number) |
| 149-152 | (Record sequence number) |
| 153-255 | (Blank) |

8.1.2.3. Workstation Log Record Format

For workstation log (W and R) records, SL\$LOG reproduces the image of all system commands and messages entered from the workstation and all system messages directed to the workstation. Bytes 123-130 contain a time stamp in the format hh::mm:ss. If your system is configured without a timer, a sequential number replaces the time stamp. The W and R records are available for use by your own programs.

The format for the workstation log record is:

| <u>Bytes</u> | <u>Contents</u> |
|--------------|---|
| 0-61 | (Image of the workstation message) |
| 121 | W (workstation) or R (terminal used as a workstation) |
| 132-137 | (Workstation user's id or terminal's ICAM name) |
| 138-139 | (Blank) |
| 140-143 | (Account number) |
| 144-148 | (Blank) |
| 149-152 | (Record sequence number) |
| 153-255 | (Blank) |

8.1.3. SL\$LOG Console Log Output

When you accumulate console logs after they are breakpointed and printed, you use the SL\$LOG program to dump them to the SYSLOG file. Once in SYSLOG, the console log records are available for use by your own programs.

Byte 121 of the output console log record contains a C to identify console log as the record type. Bytes 123-130 contain a time stamp in the format hh:mm:ss. If your system is configured without a timer, a sequential number replaces the time stamp.

The format for the console log record is:

| <u>Bytes</u> | <u>Contents</u> |
|--------------|-----------------------------------|
| 0-61 | (Image of system console message) |
| 121 | C |
| 123-130 | (Time in hh:mm:ss format) |
| 149-152 | (Record sequence number) |
| 153-255 | (Blank) |

8.1.4. Executing SL\$LOG to Transfer Job and Workstation Log or Console Log Records to Disk

A canned job control stream, named DUMPLOG, is supplied with your spooling system to enable the SL\$LOG program to be easily executed by your operator. The canned job control stream consists of the following three job control statements:

```
// JOB DUMPLOG
// GBL D,F
// DUMPLOG DUMP=&D,FILE=&F
/ &
```

Your operator enters the RUN command with DUMPLOG and parameter options. These options allow the operator to dump job and workstation log or console log records and to specify whether the output disk file is to be established, added to, or reinitialized when the log records are being transferred. The parameters your operator uses to make these selections equate to the D and F keywords and are described in the operator's spooling reference (Appendix C).

The canned job control stream calls a jproc, also named DUMPLOG, to initiate the execution of SL\$LOG under the following conditions:

- If no D specification is supplied by the operator, job accounting and log records, and workstation log records, are transferred.
- The logical file name (LFD) and file identifier (LBL) for the output file used for the job is SYSLOG.
- IF F is specified as F=ALLOC, SYSLOG is established as a 1-cylinder file on the SYSRES volume, with a dynamic expansion increment of one cylinder.
- If no F specification is supplied by the operator, SL\$LOG assumes that SYSLOG has already been established and is merely being added to.

You can change all of these default processing options with the exception of the SYSLOG lfdname for the output file. To do this, you write and file your own job control stream to execute SL\$LOG, and include the options you desire in the DUMPLOG call statement. Sperry also recommends that you change the name of your control stream so that you don't destroy the integrity of the OS/3 software. Remember, however, to tell your operator the new name of the job to be run when it is time to dump the spool LOG file, as well as the new parameters to be specified in the RUN command.

The format of the DUMPLOG jproc call statement is:

```
// [symbol] DUMPLOG [ DUMP= { ACT } [ LOG ] [ ALL ] [ CON ] ] [ , FILE= { ALLOC } ] [ , VOL { vsn } ] [ , LBL= { file-identifier } ] [ , CYLS= { n } ] [ , INC= { n } ]
```

The symbol field is not used. The DUMP keyword specifies the type of log records you want dumped from the spool file. If ACT, LOG, or ALL is specified, the spool LOG file is accessed and the job accounting (A) records, or the job log (L) and workstation log (W and R) records, are dumped. If CON is specified, the spool LOG file is accessed and the printed (breakpointed) console log (C) records are dumped.

The FILE keyword allows the jproc to generate the job control statements needed to initially allocate SYSLOG (FILE=ALLOC), reinitialize SYSLOG (FILE=INIT), or add to SYSLOG if the FILE keyword is omitted.

The VOL keyword identifies the disk volume on which you want the SYSLOG file located. The LBL keyword assigns a file identifier to the SYSLOG file. And the CYLS and INC keywords specify the initial size of SYSLOG and its dynamic expansion increment, in terms of cylinders. These two keywords are meaningful only when the FILE=ALLOC specification is present.

Note that no provision is made for you to change the logical file name (LFD) of the output disk file (SYSLOG), as SYSLOG is the only output file name recognized by the SL\$LOG program.

If you do decide to change any of the default processing options established for SL\$LOG, it is a good idea to equate the DUMP and FILE keywords to the global variable symbols D and F, respectively, so that the type of records to be dumped and the processing condition of the SYSLOG file can still be selected at run time. Thus, the control stream you would file should appear as follows:

```
// JOB jobname
// GBL D,F
// DUMPLOG DUMP=&D,FILE=&F [ ,VOL={vsn} ] [ ,LBL=file-identifier ] [ ,CYLS=n ] [ ,INC=n ]
//                                     { RUN }
/&
// FIN
```

8.1.5. Executing SL\$LOG to Transfer Job and Workstation Log or Console Log Records to Tape

Another canned job control stream, named DUMPLOGT, is supplied with your spooling system to enable your operator, using SL\$LOG, to dump log records to tape instead of disk. This canned job control stream consists of the following job control statements:

```
// JOB DUMPLOGT
// GBL D,V,C
// DUMPLOGT DUMP=&D,TVOL=&V,CKPT=&C
/&
```

Your operator enters the RUN command with DUMPLOGT and parameter options. These options allow the operator to dump job and workstation log or console log records, to specify the volume serial number of the tape to be created, and to establish whether or not checkpoint records are to appear after each job's log, workstation log, or breakpointed segment of the console log, as applicable. The parameters your operator uses to make these selections equate to the D, V, and C keywords and are described in the operator's spooling reference (Appendix C).

The canned job control stream calls a jproc, also named DUMPLOGT, to initiate the execution of SL\$LOG under the following conditions:

- If no D specification is supplied by the operator, job accounting and log records, and workstation log records, are transferred.
- If no V specification is supplied, the volume serial number of the output tape is assumed to be SYSLOG.
- If no C specification is supplied, checkpoint records are written on the output file.
- The logical file name (LFD) and file identifier (LBL) for the output file used for the job is SYSLOG.

As before, you have the ability to change all of the default processing options, except the output file name; that must remain SYSLOG.

The format of the DUMPLOGT jproc call statement is:

```
//[symbol] DUMPLOGT [DUMP={ACT  
LOG  
ALL  
CON}] [TVOL={vsn  
SYSLOG}]  
[LBL={file-identifier  
SYSLOG}] [CKPT={*  
(N)}]
```

The symbol field is not used. The DUMP keyword specifies the type of log records you want dumped from the spool file. The TVOL keyword identifies the tape volume on which you want the SYSLOG file located. The LBL keyword assigns a file identifier to the SYSLOG file. And the CKPT keyword indicates whether checkpoint records are to be written on the output file. (Note that T is appended to DUMPLOG to indicate tape output.)

The procedure for changing any of the default processing options for SL\$LOG when using tape for its output is the same as that described for disk. However, when you use tape, the DUMP, TVOL, and CKPT keywords should be equated to global variable symbols to allow these options to be selected at run time.

8.2. JOB LOG REPORT PROGRAM

The job log report program (JOBLOG) processes the SYSLOG tape or disk file and produces a formatted accounting report. This accounting report shows you the following job-related information.

- Job run date
- Account number
- Job name

- Job step name
- Time on and time off
- Elapsed time
- Allocated main storage
- Job termination code
- Switching priority (at start of job)
- Output spooling information (form name, copies, pages, and records)
- Devices used by the job
- Account summary of charges (job sorting option)
- Interactive services usage summary (where applicable)

The SYSLOG file used as input to JOBLOG is created by the SL\$LOG program. Therefore, before you can execute JOBLOG, you must have created the SYSLOG file with SL\$LOG.

8.2.1. Accounting Report Types

JOBLOG allows you to optionally select the type of job accounting reports to be produced. You may have the jobs presented in the order in which they were submitted, grouped by account number and job name, or grouped by account number only. When you select the second option, subtotals are taken if the account number or job name changes. When you specify the last option, subtotals are taken if the account number changes, and a special summary of charges by account number is produced.

Your operator executes the JOBLOG program by using the RUN command with the canned job control stream JBLOG for disk input or JBLOGT for tape input and its parameter options. The parameters your operator includes with the command, or their default settings, are used to select the type of report you want produced. As an option, you can execute JOBLOG with a job control stream you create to meet your specific needs. You'll find sample job control streams in 8.2.3 to assist you in tailoring your accounting reports, along with the JBLOG and JBLOGT operator command formats.

Based on the report type selected, JOBLOG uses the OS/3 independent sort/merge routine to order the SYSLOG file for your accounting report. Figure 8-2 shows an example of a job accounting report that has the jobs grouped by account number, job name, date, and job number. When you sort jobs for your job accounting report by account number only, an account summary listing (Figure 8-3) is included with the report. If your system includes interactive services, an interactive services usage summary report will be produced when the JOBLOG program is executed. Figure 8-4 shows an example of an interactive services usage summary report that is sorted by account number and user ID.

| OS/3 JOB LOG REPORT | | | | | | | | | | | | | 8/04/00 | | PAGE 1 | |
|---|----------|---------|-------------------|------------------|-------------------|------------------|-------------------|------------------|-----------|---------------|----------------------|--------|---------|---------|--------|--|
| RUN DATE | ACCT NR. | JOBNAME | JOB STEP | TIME ON HH MM SS | TIME OFF HH MM SS | ELAPSED HH MM SS | CPU TIME HH MM SS | MEMORY ALLOCATED | TERM CODE | SWITCH PRIOR. | SPOOLING INFORMATION | | | | | |
| | | | | | | | | | | | FORM | COPIES | PAGES | RECORDS | | |
| 08/04 | | ASM | ASM | 09:52:17 | 09:55:08 | 00:02:42 | 00:00:08 | 22,016 | 020 | 10 | STAND1 | 1 | 1 | | | |
| DEVICES: 303-00001228 PRT=00000005 | | | | | | | | | | | | | | | | |
| JOBS PROCESSED: 1 TOTAL ELAPSED TIME: 00 02 42 TOTAL PAGES SPOOLED: 1 | | | | | | | | | | | | | | | | |
| STEPS PROCESSED: (1) TOTAL CPU TIME: 00 00 08 TOTAL RECDs SPOOLED: 0 | | | | | | | | | | | | | | | | |
| 08/04 | | CORH | CORH | 09:54:20 | 09:55:37 | 00:00:52 | 00:00:03 | 24,064 | 000 | 10 | STAND1 | 2 | 1 | 52 | | |
| DEVICES: 303-00000339 001-00000053 PRT=00000056 PCH=00000052 | | | | | | | | | | | | | | | | |
| JOBS PROCESSED: 1 TOTAL ELAPSED TIME: 00 00 52 TOTAL PAGES SPOOLED: 2 | | | | | | | | | | | | | | | | |
| STEPS PROCESSED: (1) TOTAL CPU TIME: 00 00 03 TOTAL RECDs SPOOLED: 52 | | | | | | | | | | | | | | | | |
| 08/04 | | LIBRINT | LIBS | 09:52:10 | 09:52:50 | 00:00:29 | 00:00:01 | 25,600 | 020 | 10 | STAND1 | 1 | 0 | | | |
| DEVICES: 303-00000130 PRT=00000001 | | | | | | | | | | | | | | | | |
| JOBS PROCESSED: 1 TOTAL ELAPSED TIME: 00 00 29 TOTAL PAGES SPOOLED: 0 | | | | | | | | | | | | | | | | |
| STEPS PROCESSED: (1) TOTAL CPU TIME: 00 00 01 TOTAL RECDs SPOOLED: 0 | | | | | | | | | | | | | | | | |
| 08/04 | | PRINT1 | SMDIPR | 09:51:57 | 09:52:36 | 00:00:39 | 00:00:01 | 19,968 | 000 | 10 | STAND1 | 1 | 0 | | | |
| DEVICES: 303-00000070 | | | | | | | | | | | | | | | | |
| JOBS PROCESSED: 1 TOTAL ELAPSED TIME: 00 00 39 TOTAL PAGES SPOOLED: 0 | | | | | | | | | | | | | | | | |
| STEPS PROCESSED: (1) TOTAL CPU TIME: 00 00 01 TOTAL RECDs SPOOLED: 0 | | | | | | | | | | | | | | | | |
| 08/04 | | PRINT2 | **JOB CANCELLED** | | | | | | | | | | | | | |
| JOBS PROCESSED: 0 TOTAL ELAPSED TIME: 00 00 00 TOTAL PAGES SPOOLED: 0 | | | | | | | | | | | | | | | | |
| STEPS PROCESSED: (0) TOTAL CPU TIME: 00 00 00 TOTAL RECDs SPOOLED: 0 | | | | | | | | | | | | | | | | |
| 08/04 | | RFG | RPG11 | 09:54:08 | 09:54:43 | 00:00:21 | 00:00:01 | 31,744 | 051 | 10 | | | | | | |
| DEVICES: 303-00000139 | | | | | | | | | | | | | | | | |
| JOBS PROCESSED: 1 TOTAL ELAPSED TIME: 00 00 21 TOTAL PAGES SPOOLED: 0 | | | | | | | | | | | | | | | | |
| STEPS PROCESSED: (1) TOTAL CPU TIME: 00 00 01 TOTAL RECDs SPOOLED: 0 | | | | | | | | | | | | | | | | |
| 08/04 | | INPT | TSTROR | DATA | 09:56:44 | 09:57:01 | 00:00:13 | 00:00:02 | 25,600 | 020 | 10 | STAND1 | 1 | 2 | | |
| DEVICES: 303-00000343 PRT=00000010 | | | | | | | | | | | | | | | | |
| JOBS PROCESSED: 1 TOTAL ELAPSED TIME: 00 00 13 TOTAL PAGES SPOOLED: 2 | | | | | | | | | | | | | | | | |
| STEPS PROCESSED: (1) TOTAL CPU TIME: 00 00 02 TOTAL RECDs SPOOLED: 0 | | | | | | | | | | | | | | | | |
| OS/3 JOB LOG SUMMARY 8/04/00 | | | | | | | | | | | | | | | | |
| REPORTING PERIOD: 09:51 77/08/04 THRU 09:57 77/08/04 | | | | | | | | | | | | | | | | |
| JOBS PROCESSED: 6 TOTAL ELAPSED TIME: 00 05 08 TOTAL PAGES SPOOLED: 5 | | | | | | | | | | | | | | | | |
| STEPS PROCESSED: (6) TOTAL CPU TIME: 00 00 16 TOTAL RECDs SPOOLED: 52 | | | | | | | | | | | | | | | | |
| DEVICES USED/TOTAL EXCP'S: | | | | | | | | | | | | | | | | |
| 303-00002245 PRT=00000076 001-00000053 PCH=00000052 | | | | | | | | | | | | | | | | |

Figure 8-2. Typical Job Accounting Report

| OS/3 ACCOUNT SUMMARY LISTING | | | | | | | 3/22/00 | |
|--|------|---------|--------------|----------|-------------|---------------|---------|--|
| REPORTING PERIOD: DR:16 00/02/22 THRU 18:01 00/03/22 | | | | | | | | |
| ACCT # | JOBS | (STEPS) | ELAPSED TIME | CPU TIME | SPOOL PAGES | SPOOL RECORDS | | |
| VONE | 2 | 2 | 09:44:57 | 01:04:37 | 61 | 581 | | |
| A020 | 3 | 3 | 00:05:10 | 00:00:24 | 9 | 0 | | |
| A064 | 2 | 2 | 00:43:47 | 00:01:32 | 4 | 0 | | |
| A087 | 1 | 2 | 00:46:38 | 00:03:27 | 7 | 0 | | |
| A089 | 1 | 1 | 00:57:31 | 00:04:10 | 95 | 0 | | |
| A362 | 18 | 29 | 05:59:09 | 00:31:22 | 837 | 0 | | |
| A591 | 51 | 224 | 27:23:17 | 02:47:21 | 2016 | 2 | | |
| B550 | 7 | 16 | 01:56:36 | 00:11:59 | 108 | 0 | | |
| K906 | 2 | 2 | 01:08:35 | 00:04:32 | 115 | 0 | | |
| K964 | 1 | 2 | 00:50:59 | 00:07:40 | 18 | 0 | | |
| L907 | 2 | 4 | 00:48:35 | 00:16:55 | 8 | 0 | | |
| S912 | 1 | 1 | 00:00:22 | 00:00:02 | 2 | 0 | | |
| 6983 | 2 | 2 | 00:57:18 | 00:02:51 | 63 | 0 | | |
| TOT: | 93 | 292 | 51:24:54 | 05:16:52 | 3363 | 583 | | |

Figure 8-3. Typical Account Summary Listing

| INTERACTIVE SERVICES USAGE SUMMARY | | | | | | | | | | | PAGE 1 |
|------------------------------------|----------|---------|-----------|------------|--------------|----------|-------|----------|----------------|-------|------------|
| RUN DATE | ACCT NO. | USER ID | LOGGED ON | LOGGED OFF | CONNECT TIME | CPU TIME | EXCPS | COMMANDS | FILES ACCESSED | SVCS | TRANSIENTS |
| 05/04 | | BISBEE | 9:00:57 | 9:31:14 | 0:30:19 | 0:00:34 | 2790 | 8 | 12 | 4121 | 182 |
| 05/04 | | BISBEE | 9:59:49 | 10:07:48 | 0:07:59 | 0:00:13 | 1167 | 7 | 5 | 1673 | 79 |
| 05/04 | | BISBEE | 11:24:53 | 11:55:00 | 0:30:07 | 0:00:30 | 2323 | 7 | 10 | 3407 | 142 |
| 05/04 | | BISBEE | 14:41:14 | 15:22:27 | 0:41:13 | 0:00:19 | 1337 | 22 | 6 | 2223 | 112 |
| | | BISBEE | USER | TOTALS | 1:49:38 | 0:01:36 | 7617 | 44 | 33 | 11514 | 515 |
| 05/04 | | COFFEY | 8:59:02 | 8:59:15 | 0:00:13 | 0:00:00 | 72 | 2 | 0 | 100 | 6 |
| 05/04 | | COFFEY | 8:59:43 | 11:04:15 | 2:04:31 | 0:01:46 | 5596 | 4 | 8 | 7728 | 170 |
| 05/04 | | COFFEY | 14:09:44 | 14:55:47 | 0:46:03 | 0:01:19 | 3916 | 3 | 8 | 5444 | 147 |
| | | COFFEY | USER | TOTALS | 2:50:47 | 0:03:05 | 9584 | 9 | 16 | 13281 | 283 |
| 05/04 | | CPY | 9:42:34 | 9:49:31 | 0:06:57 | 0:00:04 | 574 | 7 | 3 | 778 | 40 |
| 05/04 | | CPY | 9:49:47 | 11:30:56 | 1:41:09 | 0:06:59 | 41778 | 8 | 205 | 63809 | 2782 |
| | | CPY | USER | TOTALS | 1:48:06 | 0:07:03 | 42312 | 15 | 208 | 64587 | 2822 |

| INTERACTIVE SERVICES USAGE SUMMARY | | | | | | | | | | | PAGE 2 |
|------------------------------------|----------|---------|-----------|------------|--------------|----------|-------|----------|----------------|-------|------------|
| RUN DATE | ACCT NO. | USER ID | LOGGED ON | LOGGED OFF | CONNECT TIME | CPU TIME | EXCPS | COMMANDS | FILES ACCESSED | SVCS | TRANSIENTS |
| 05/04 | | LESLIE | 13:33:50 | 13:40:09 | 0:06:19 | 0:00:12 | 1142 | 3 | 6 | 1778 | 85 |
| | | LESLIE | USER | TOTALS | 0:06:19 | 0:00:12 | 1142 | 3 | 6 | 1778 | 85 |
| 05/04 | | LOGOFF | 9:27:00 | 9:27:20 | 0:00:19 | 0:00:02 | 212 | 3 | 1 | 343 | 21 |
| | | LOGOFF | USER | TOTALS | 0:00:19 | 0:00:02 | 212 | 3 | 1 | 343 | 21 |
| 05/04 | | LOGON | 14:47:38 | 15:10:39 | 0:23:01 | 0:01:07 | 4318 | 4 | 14 | 6849 | 194 |
| | | LOGON | USER | TOTALS | 0:23:01 | 0:01:07 | 4318 | 4 | 14 | 6849 | 194 |
| 05/04 | | MARTY | 16:22:18 | 16:33:49 | 0:11:31 | 0:00:20 | 1553 | 8 | 6 | 2342 | 99 |
| | | MARTY | USER | TOTALS | 0:11:31 | 0:00:20 | 1553 | 8 | 6 | 2342 | 99 |
| 05/04 | | MRT | 11:40:28 | 11:51:49 | 0:11:20 | 0:00:17 | 1821 | 11 | 14 | 2827 | 155 |
| 05/04 | | MRT | 13:07:45 | 15:53:10 | 2:45:25 | 0:07:21 | 25678 | 33 | 43 | 35472 | 1197 |
| 05/04 | | MRT | 16:31:37 | 17:32:48 | 1:01:11 | 0:03:12 | 11954 | 28 | 29 | 17183 | 697 |

| INTERACTIVE SERVICES USAGE SUMMARY | | | | | | | | | | | PAGE 3 |
|------------------------------------|----------|-------------|-----------|------------|--------------|----------|--------|----------|----------------|--------|------------|
| RUN DATE | ACCT NO. | USER ID | LOGGED ON | LOGGED OFF | CONNECT TIME | CPU TIME | EXCPS | COMMANDS | FILES ACCESSED | SVCS | TRANSIENTS |
| 05/04 | | RON | 12:22:24 | 12:45:32 | 0:23:08 | 0:00:27 | 1446 | 7 | 5 | 2211 | 90 |
| | | RON | USER | TOTALS | 0:23:08 | 0:00:27 | 1446 | 7 | 5 | 2211 | 90 |
| 05/04 | | RTS | 14:54:24 | 15:13:09 | 0:18:45 | 0:01:13 | 6521 | 10 | 28 | 9676 | 477 |
| | | RTS | USER | TOTALS | 0:18:45 | 0:01:13 | 6521 | 10 | 28 | 9676 | 477 |
| 05/04 | | STERN | 13:10:45 | 16:30:33 | 3:19:48 | 0:01:38 | 8334 | 26 | 38 | 12273 | 527 |
| | | STERN | USER | TOTALS | 3:19:48 | 0:01:38 | 8334 | 26 | 38 | 12273 | 527 |
| 05/04 | | SUSAN | 11:40:24 | 11:53:00 | 0:12:35 | 0:01:46 | 8264 | 3 | 8 | 13185 | 1106 |
| | | SUSAN | USER | TOTALS | 0:12:35 | 0:01:46 | 8264 | 3 | 8 | 13185 | 1106 |
| 05/04 | | TAS | 8:36:28 | 8:46:51 | 0:10:22 | 0:00:36 | 2699 | 14 | 13 | 4140 | 193 |
| 05/04 | | TAS | 11:20:11 | 11:20:55 | 0:00:44 | 0:00:02 | 241 | 4 | 2 | 373 | 25 |
| 05/04 | | TAS | 11:21:09 | 11:21:51 | 0:00:42 | 0:00:02 | 199 | 6 | 2 | 337 | 23 |
| 05/04 | | TAS | 11:22:01 | 11:23:23 | 0:01:22 | 0:00:01 | 97 | 7 | 0 | 145 | 8 |
| | | TAS | USER | TOTALS | 0:13:10 | 0:00:41 | 3276 | 31 | 17 | 4995 | 249 |
| 05/04 | | WAE999 | 16:07:55 | 16:13:18 | 0:05:22 | 0:00:14 | 1661 | 10 | 10 | 2494 | 128 |
| | | WAE999 | USER | TOTALS | 0:05:22 | 0:00:14 | 1661 | 10 | 10 | 2494 | 128 |
| | | ACCOUNT | TOTALS | | 5:33:15 | 0:07:08 | 34942 | 114 | 132 | 53177 | 2859 |
| | | ***** GRAND | TOTALS | | 22:45:21 | 0:41:02 | 182675 | 375 | 673 | 273302 | 11395 |

Figure 8-4. Typical Interactive Services Usage Summary

8.2.2. Executing JOBLOG

Once your job log SYSLOG file is created by SL\$LOG, you execute JOBLOG. Using the SYSLOG file as input, JOBLOG calls on the OS/3 independent sort/merge routine to create a sorted SYSLOG output file. After the sorting process is complete, JOBLOG uses the sorted file to produce your accounting report. Formats 1 and 2 are the operator commands used to execute JOBLOG; format 1 shows JOBLOG using disk input, while format 2 uses tape input.

Format 1:

$$\left\{ \begin{array}{l} \text{RUN} \\ \text{RV} \end{array} \right\} \text{JOBLOG} \left[\dots, V = \left\{ \begin{array}{l} \text{vs n} \\ \text{RES} \end{array} \right\}, L = \left\{ \begin{array}{l} \text{file-identifier} \\ \text{SYSLOG} \end{array} \right\}, S = \left\{ \begin{array}{l} \text{A} \\ \text{B} \\ \text{C} \end{array} \right\} \right]$$

The V parameter (1-6 characters) indicates the volume serial number of the disk pack containing the SYSLOG input file. This volume is also used for the sorted SYSLOG output file. The L parameter (1-6 characters) specifies the file identifier (LBL) used for the SYSLOG file when SL\$LOG created it.

The S parameter indicates the sort option. You specify A to sort the jobs in the order submitted, B to sort jobs by account number and job name, or C to sort jobs by account number only. If your system includes interactive services, you specify A to sort the interactive services usage summary report in the order the interactive sessions were initiated, B to sort the report by account number and user ID, or C to sort the report by account number.

Format 2:

$$\left\{ \begin{array}{l} \text{RUN} \\ \text{RV} \end{array} \right\} \text{JBLOGT} \left[\dots, V = \left\{ \begin{array}{l} \text{vs n} \\ \text{SYSLOG} \end{array} \right\}, S = \left\{ \begin{array}{l} \text{A} \\ \text{B} \\ \text{C} \end{array} \right\} \right]$$

The V parameter (1-6 characters) indicates the volume serial number of the tape containing the SYSLOG input file. The sort/merge routine requires two tapes for sorting SYSLOG - the SYSLOG input tape and a tape for the sorted SYSLOG output. Both tapes must be mounted, but only the volume serial number for the input tape is specified.

The S parameter indicates the sort option. You specify A to sort the jobs in the order submitted, B to sort jobs by account number and job name, or C to sort jobs by account number only. If your system includes interactive services, you specify A to sort the interactive services usage summary report in the order the interactive sessions were initiated, B to sort the report by account number and user ID, or C to sort the report by account number. (Note that T is appended to JBLOG to indicate tape input.)

Example 1:

The name and file identifier for the job accounting file is SYSLOG and it resides on the SYSRES volume. You wish to have the report sorted by account number and job name. You use the following command:

```
RV JBLOG
```

Example 2:

The SYSLOG job accounting file resides on tape with a volume serial number called ACCTG. The file is to be sorted in the order in which the jobs were submitted. You use the following command:

```
RV JBLOGT,,V=ACCTG,S=A
```

8.2.3. Sample Job Control Streams

The following job control streams are offered to assist you in creating your own job control streams to produce an accounting report to meet your specific needs. Note that you must assign work areas if you use the OS/3 independent sort/merge routine.

The first job control stream sorts the jobs in the order in which they were submitted (option A) with the SYSLOG file residing on disk.

```
// JOB JOBLOG
// OPTION JOBDU
// DVC 20 // LFD PRNTR
// DM01 WORK1
// DM02 WORK2
// DVC 50 // VOL REL050 // LBL SYSLOG // LFD SORTIN1
// DVC 50 // VOL REL050 // LBL OUTLOG // LFD X // SCR X
// DVC 50 // VOL REL050 // EXT ST,C,1,CYL,1
// LBL OUTLOG // LFD SORTOUT
// EXEC SORT
/$
SORT FIELDS=(166,1,,,167,15,,,154,8,,,146,4,,,162,3,,,165,1)
RECORD RCSZ=256,TYPE=F,DEBLANK=(L,122)
INPFIL BLKSIZE=256
OUTFIL BLKSIZE=256,VOS=(100)
END
/*
// OPTION JOBDDUMP
// DVC 20 // LFD OUT-FILE
// DVC 50 // VOL REL050 // LBL OUTLOG // LFD DISK-IN
// EXEC JOBLOG
// ACCT-FIL TEMP1
// PARAM DISK
/&
// FIN
```

The following job control stream sorts jobs in account number and job name order (option B) with the SYSLOG file residing on disk.

```
// JOB JOBLLOG
// OPTION JOBDU
// DVC 20 // LFD PRNTR
// DM01 WORK1
// DM02 WORK2
// DVC 50 // VOL REL050 // LBL SYSLOG // LFD SORTIN1
// DVC 50 // VOL REL050 // LBL OUTLOG // LFD X // SCR X
// DVC 50 // VOL REL050 // EXT ST,C,1,CYL,1
// LBL OUTLOG // LFD SORTOUT
// EXEC SORT
/$
SORT FIELDS=(166,1,,142,4,,134,8,,154,8,,146,4,,162,3,,165,1)
RECORD RCSZ=256,TYPE=F,DEBLANK=(L,122)
INPFIL BLKSIZE=256
OUTFIL BLKSIZE=256,VOS=(100)
END
/*
// OPTION JOBDUMP
// DVC 20 // LFD OUT-FILE
// DVC 50 // VOL REL050 // LBL OUTLOG // LFD DISK-IN
// EXEC JOBLLOG
// ACCT-FIL TEMP1
// PARAM DISK,GROUP
/&
// FIN
```

The next job control stream uses option A but with SYSLOG residing on tape.

```
// JOB JOBLLOG
// OPTION DUMP
// DVC 20 // LFD PRNTR
// DVC 90 // VOL SYSLOG // LFD SORTIN1
// DVC 91 // VOL N.OUTLOG(PREP) // LFD SORTOUT
// DM01 WORK1
// DM02 WORK2
// EXEC SORT
/$
SORT FIELDS=(166,1,,167,15,,154,8,,146,4,,162,3,,165,1)
RECORD RCSZ=256,TYPE=F,DEBLANK=(L,122)
INPFIL BLKSIZE=256,OPEN=RWD,CLOSE=RWD
OUTFIL BLKSIZE=256,OPEN=RWD,CLOSE=RWD
END
/*
```

(continued)

```
// OPTION JOBDUMP
// DVC 20 // LFD OUT-FILE
// DVC 91 // VOL OUTLOG // LFD TAPE-IN
// EXEC JOBLG
// ACCT-FIL TEMP1
// PARAM TAPE
/&
// FIN
```

The last job control stream uses option B with SYSLOG residing on tape.

```
// JOB JOBLG
// OPTION DUMP
// DVC 20 // LFD PRNTR
// DVC 90 // VOL SYSLOG // LFD SORTIN1
// DVC 91 // VOL N.OUTLOG(PREP) // LFD SORTOUT
// DM01 WORK1
// DM02 WORK2
// EXEC SORT
/$
SORT FIELDS=(166,1,,,142,4,,,134,8,,,154,8,,,146,4,,,162,3,,,165,1)
RECORD RCSZ=256,TYPE=F,DEBLANK=(L,122)
INPFIL BLKSIZE=256,OPEN=RWD,CLOSE=RWD
OUTFIL BLKSIZE=256,OPEN=RWD,CLOSE=RWD
OPTION SHARE=SM01,RESERV=SM03
END
/*
// OPTION JOBDUMP
// DVC 20 // LFD OUT-FILE
// DVC 91 // VOL OUTLOG // LFD TAPE-IN
// EXEC JOBLG
// ACCT-FIL TEMP1
// PARAM TAPE,GROUP
/&
// FIN
```

For a description of the sort parameters, refer to the sort/merge user guide/programmer reference.

When your SYSLOG file resides on magnetic tape, you must make the appropriate changes in the device assignment sets and change the JOBLG program // PARAM card from DISK to TAPE. If the // PARAM card is missing or incorrect, the following message is displayed:

PARAM CARD MISSING OR INCORRECT FORMAT. JOB ABORTED.



9. Establishing and Controlling Your Spooling and Job Accounting System

9.1. GENERAL

Table 9-1 provides a summary of the OS/3 spooling system capabilities. Each spooling option is listed with its corresponding method of establishment and control. First we'll discuss how the levels of control are used and how you establish spooling options during system generation; then we'll look at the summary table.

9.2. SPOOLING AND JOB ACCOUNTING CONTROL LEVELS

Spooling provides you with four levels of control:

1. Establishing your spooling system requirements with system generation parameters
2. Controlling spooling in a batch environment with job control statements, a remote batch command, and a macroinstruction
3. Controlling, with restrictions, some spooling functions with workstation commands
4. Controlling all spooling functions throughout your system with operator commands

The first level of control is used during SYSGEN, when you establish the spooling options to be used as defaults (level 1).

Your programmer controls spooling options in a batch environment at the second level by using job control statements, a remote batch command, and a macroinstruction. Level 2 specifications override level 1 defaults.

The programmer also has the option of controlling spooling options at level 3 from an interactive workstation. For summary purposes, Table 9-1 shows your programmer as the workstation user; however, all workstation controls are available to any workstation user (for example, you, your programmer, and your operator). Some of your operator's spooling commands can be entered from the workstation with certain limitations. The command affects only those workstation user jobs that the workstation user initiates. Using these conditions, a level 3 workstation entry overrides a level 1 and level 2 specification.

Your programmer (workstation user) may control the spooling environment by using the general editor (EDT). With EDT, it is possible to read selected files from the spool file and other system libraries into workstation space, in order to manipulate them. Any file with the workstation user's identification can be manipulated in any manner - from adding, deleting, or modifying a single character up to an entire file. These basic EDT functions are included in the table as a means to control spooling. See your general editor (EDT) user guide/programmer reference for a description of all EDT capabilities and functions.

Your operator controls spooling at level 4, which overrides entries at any other level. Commands for the entire spooling system, as well as the input reader commands and the output writer commands and messages, are entered at the console. The operator's spooling system commands may include:

- Spool file *directories* (type of queue) to identify one of the input or output directories in the system pool control table (such as the input, print output, remote punch output, or distributed data processing print output directories).
- *Modifiers* (parameter options) to be used as selection criteria for file processing (for example, processing files created by a job name, files for a form name, files with an account number, etc).

The operator's output writer commands (to manually load the writer) and messages (to manually direct an active writer) may include a *function* (parameter option) to identify the mode of operation or the processing criteria desired for that specific output writer. For additional control, your operator includes spooling system modifiers with the function.

9.3. ESTABLISHING OPTIONS DURING SYSTEM GENERATION

The spooling and job accounting options you establish during system generation are either part of the supervisor generation (SUPGEN) or part of the input/output generation (I/OGEN).

To establish spooling in your system, you specify one of the following SUPGEN spooling levels:

- Output spooling to local devices
- Input and output spooling to local devices
- Input and output spooling to local and remote batch devices (remote batch processing is included). With this specification, you have the option to restrict spooling to output only.
- Input and output spooling with distributed data processing included (automatically includes remote batch spooling). With this specification, you have the option to restrict spooling to output only.

Once this spooling level is entered, additional SUPGEN parameters establish your spooling and job accounting system, excluding real and virtual devices, as your operator initializes it during the IPL procedure.

I/OGEN allocates the number of virtual input and output devices, by device type, to be used by the supervisor, provided you have also identified a similar type of real device. When you include virtual devices during I/OGEN, jobs are scheduled based on the number of virtual devices rather than the number of real devices.

I/OGEN also is used to specify the number of locally or remotely connected workstations and associated auxiliary printers.

Table 9-1. Summary of OS/3 Spooling and Job Accounting Capabilities and Controls (Part 1 of 7)

| Spooling Specifications | System Administrator | Programmer | | Operator |
|--|---|---|--|----------------|
| | SYSGEN | Batch | Interactive Workstation (User-id Only) | Console |
| Options Available Starting at Level 1 - SUPGEN | Level 1 | Level 2 | Level 3 | Level 4 |
| 1. Establish spooling: <ul style="list-style-type: none"> ▪ Output only ▪ Input and output ▪ Input/output plus remote batch, with option to suppress input spooling ▪ Input/output/remote batch plus distributed data processing, with option to suppress input spooling ▪ Do not establish spooling. | SPOOLING parameter and SPOOLNOINPUT parameter | | | |
| 2. Specify spool file volume serial number. | SPOOLVSN parameter | | | IPL procedure |
| 3. Specify multivolume spool file volume serial numbers. | SPOOLVSN parameter | | | IPL procedure |
| 4. Specify spool file space (cylinders) for each spool file volume serial number. | SPOOLCYL parameter | | | |
| 5. Specify control and buffer space in main storage. | SPOOLBUFR parameter | | | |
| 6. Specify output file space in main storage. | SPOOLWBUFR parameter | // SPOOL jproc (files) and // JOB (jobs) statements | | |
| 7. Specify cross-references (pointers). | SPOOLMAP parameter | | | |
| 8. Specify communications network pointer. | SPOOLICAM parameter | | | SET IC command |

Table 9-1. Summary of OS/3 Spooling and Job Accounting Capabilities and Controls (Part 2 of 7)

| Spooling Specifications | System Administrator | Programmer | | Operator |
|---|-----------------------|--|--|-------------------------------------|
| | SYSGEN | Batch | Interactive Workstation (User-id Only) | Console |
| Level 1 Outline - SYSGEN (cont) | Level 1 | Level 2 | Level 3 | Level 4 |
| 9. Compressed output records (blanks deleted and re-created). | SPOOLCOMP parameter | // SPOOL jproc | | |
| 10. Specify test lines message to operator. | SPOOLTEST parameter | // SPOOL jproc | | SET SPL command |
| 11. Update directories in system spool control table. | SPOOLUPDATE parameter | // SPOOL jproc | | |
| 12. Specify output writer mode of operation (nonburst/burst). | SPOOLBURST parameter | | RP, PR, PU commands | SET SPL and RP, PR, PU, PD commands |
| 13. Specify burst mode tailoring parameters for output writer file processing. | SPOOLMODE parameter | | RP, PR, PU commands | SET SPL and RP, PR, PU, PD commands |
| 14. Specify output file headers. | SPOOLHDR parameter | // SPOOL jproc (files) and // JOB (jobs) statement | | SET SPL command |
| 15. Specify job log printing criteria for a terminated job; turn on and off, and specify log types (L and A). | SPOOLPRT parameter | // JOB statement | | SET SPL command |
| 16. Maintain job log accounting records. | JOBACCT parameter | | | |
| 17. Specify job logs and workstation logs accumulated for future use. | SYSLOG parameter | | | SET SPL command |
| 18. Specify console log and workstation log collection; turn on and off. | CONSOLOG parameter | | | SET SPL command |
| 19. Specify console log printing; turn on and off. | CONPRINT parameter | | | SET SPL command |
| 20. Specify console logs accumulated for future use | RETAINLOG parameter | | | SET SPL command |
| 21. Specify spooling initialization; cold start and warm start | SPOOLRECV parameter | | | IPL procedure |

Table 9-1. Summary of OS/3 Spooling and Job Accounting Capabilities and Controls (Part 3 of 7)

| Spooling Specifications | System Administrator | Programmer | | Operator |
|---|-------------------------|-------------------|--|-----------------------------|
| | SYSGEN | Batch | Interactive Workstation (User-id Only) | Console |
| Options Available Starting at Level 1 - I/OGEN | Level 1 | Level 2 | Level 3 | Level 4 |
| 1. Specify number of virtual readers. | READER parameter | | | |
| 2. Specify number of virtual printers. | PRINTER parameter | | | |
| 3. Specify number of virtual punches. | PUNCH parameter | | | |
| 4. Specify number of locally connected workstations. | WORKSTATION parameter | | | |
| 5. Specify number of auxiliary printers associated with locally connected workstations. | AUXPRINTER parameter | | | |
| 6. Specify number of remotely connected workstations. | REWORKSTATION parameter | | | |
| 7. Specify number of auxiliary printers associated with remotely connected workstations. | REMPRINTER parameter | | | |
| | | Programmer | | Operator |
| | | Batch | Interactive Workstation (User-id Only) | Console |
| Additional Options Available Starting at Level 2 (Job Control Statements, a Remote Batch Command, and a Macroinstruction) | | Level 2 | Level 3 | Level 4 |
| 1. Retain an input file after processing. | | // DATA statement | | IN command |
| 2. Specify automatic job scheduling for an input file. | | // RUN statement | | |
| 3. Specify multiple input files. | | // DATA statement | | |
| 4. Retain an output file after processing. | | // SPOOL jproc | RP, PR, and PU commands | RP, PR, PU, and PD commands |

Table 9-1. Summary of OS/3 Spooling and Job Accounting Capabilities and Controls (Part 4 of 7)

| Spooling Specifications | | Programmer | | Operator |
|--|--|---|--|-------------------------|
| | | Batch | Interactive Workstation (User-id Only) | Console |
| Level 2 Options (cont) | | Level 2 | Level 3 | Level 4 |
| 5. Breakpoint printer and punch files. | | // SPOOL jproc (number of accumulated pages) and BRKPT (immediate) macroinstruction | BRKPT command for local user-id | BRKPT command |
| 6. Hold queued output files from processing (local and remote). | | // SPOOL jproc | HOLD command for local user-id | HOLD command |
| 7. Specify expected maximum number of records in a large output file (prevents job halts and message to operator). | | // SPOOL jproc | | |
| 8. Specify multiple copies of an output file (local). | | // SPOOL jproc | RP, PR, and PU commands | RP, PR and PU commands |
| 9. Write diskettes in data set label mode. | | // DVC statement | | |
| 10. Redirect output to tape, disk, or format label diskette. | | // SPOOL jproc | | RP, PR, and PU commands |
| 11. Specify identifier for a job's log in accumulated LOG file. | | // OPTION statement | | |
| 12. Specify that job's log is printed with job's output, using specific printer in nonburst mode. | | // OPTION statement | | |
| 13. Request operator control over a remote job. | | /RMSG (on card) remote batch command | | |
| 14. Specify multiple output destinations for a remote job. | | // DST statement | | |
| 15. Redirect output to override the // DST for a remote job's local destination. | | // SPOOL jproc | | PR and PU commands |
| 16. Request collected output files for remote user. | | /RMSG (on card) remote batch command | | |
| 17. Delete spool subfiles of same label name before creating the new subfile. | | // DATA statement | | IN command |

Table 9-1. Summary of OS/3 Spooling and Job Accounting Capabilities and Controls (Part 5 of 7)

| Spooling Specifications | | Programmer | | Operator |
|---|--|--|--|-------------------------|
| | | Batch | Interactive Workstation (User-id Only) | Console |
| Level 2 Options (cont) | | Level 2 | Level 3 | Level 4 |
| 18. Direct print output to workstation auxiliary printer. | | // ROUTE statement and // OPTION statement | | |
| | | Programmer | | Operator |
| | | | Interactive Workstation (User-id Only) | Console |
| Additional Options Available Starting at Level 3 (Workstation Commands) | | | Level 3 | Level 4 |
| 1. Specify workstation log printing; turn on and off. | | | LOGON command | SET SPL command |
| 2. Display status of active and completed input and output files in spool file (local). | | | DISPLAY command | DISPLAY command |
| 3. Display status of workstation log lines in spool file. | | | DISPLAY command | |
| 4. Delete queued files and files on hold, input and output, from spool file (local). | | | DELETE command | DELETE command |
| 5. Release queued output files from hold state (local). | | | BEGIN command | BEGIN command |
| 6. Breakpoint workstation log to automatically print. | | | BRKPT command | |
| 7. Specify job initiator message to breakpoint printer or punch file after specified number of records created. | | | Message response option | Message response option |
| 8. Use EDT to access and manipulate selected files in the spool file. Basically includes: | | | All EDT functions for user-id files | |
| <ul style="list-style-type: none"> ■ Additions to file ■ Changes to file ■ Delete file ■ Display file | | | | |
| 9. Manually load or direct an output writer. | | | RP, PR, and PU commands | |

Table 9-1. Summary of OS/3 Spooling and Job Accounting Capabilities and Controls (Part 6 of 7)

| Spooling Specifications | | | | Operator |
|--|--|--|--|-----------------------------|
| | | | | Console |
| Additional Options Available Starting at Level 4 (Console Commands) | | | | Level 4 |
| 1. Display status of active and completed input and output files in spool file (local and remote). | | | | DISPLAY command |
| 2. Display status of console log lines in spool file. | | | | DISPLAY command |
| 3. Display status of system operating mode (nonburst/burst). | | | | DISPLAY command |
| 4. Delete queued files and files on hold, input and output, from spool file (local and remote). | | | | DELETE command |
| 5. Hold active and queued input and output files from processing (local and remote). | | | | HOLD command |
| 6. Hold output files currently being processed (local and remote). | | | | PR, PU, and PD commands |
| 7. Release active and queued input and output files from hold state (local and remote). | | | | BEGIN command |
| 8. Breakpoint console log to automatically print. | | | | BRKPT command |
| 9. Load an input reader. | | | | IN command |
| 10. Read data set label mode diskette input. | | | | IN command |
| 11. Read input from 80-, 51-, 66-, or 96-column cards. | | | | IN command |
| 12. Read input data from a single volume unlabeled tape. | | | | IN command |
| 13. Manually load or direct an output writer. | | | | RP, PR, PU, and PD commands |
| 14. Swap devices (printer and punch). | | | | RP, PR, and PU commands |
| 15. Reintroduce redirected output for printer or punch. | | | | RP, PR, and PU commands |
| 16. Locally print or punch files to override remote destination designated. | | | | PR and PU commands |

Table 9-1. Summary of OS/3 Spooling and Job Accounting Capabilities and Controls (Part 7 of 7)

| Spooling Specifications | | | | Operator |
|--|--|--|--|---|
| | | | | Console |
| Level 4 Options (cont) | | | | Level 4 |
| 17. Transfer accumulated job and workstation log from spool LOG file to SYSLOG tape file; options are permitted to specify log types (L, A, and W/R); tape volume serial number; and checkpoint records. | | | | RUN DUMPLOGT to execute SL\$LOG program |
| 18. Transfer accumulated job and workstation logs from spool LOG file to SYSLOG disk file; options are permitted to specify log types (L, A, and W/R) and to establish, add to, or reinitialize SYSLOG. | | | | RUN DUMPLOG to execute SL\$LOG program |
| 19. Transfer accumulated console log from spool LOG file to SYSLOG tape file; options are permitted to specify tape volume serial number and checkpoint records. | | | | RUN DUMPLOGT to execute SL\$LOG program |
| 20. Transfer accumulated console log from spool LOG file to SYSLOG disk file; options are permitted to establish, add to, or reinitialize SYSLOG. | | | | RUN DUMPLOG to execute SL\$LOG program |
| 21. Produce a job accounting report from SYSLOG tape file; options are permitted to sort jobs in order submitted, in accounting number and job name order, or in accounting number order. | | | | RUN JBlogT to execute JOBLOG program |
| 22. Produce a job accounting report from SYSLOG disk file; options are permitted to sort jobs in order submitted, in accounting number and job name order, or in accounting number order. | | | | RUN JBlog to execute JOBLOG program |



Appendix A. Format of Output Writer Tape

■ Header Records

The header records are the equivalent of the nonstandard labels. A minimum of two records exists for each redirected output file on the tape. These records are the file header ID and the subdirectory entry. In addition, vertical format buffer (VFB) and load code buffer (LCB) records may also exist.

- File Header ID

The first record of each file contains a file header ID. You should note that VOL labels on tape are not preserved. The format of this 80-byte record is:

| <u>Field Number</u> | <u>Location</u> | <u>Contents</u> |
|---------------------|-----------------|------------------------------------|
| 1 | 1-19 | *** OUTPUT FILE *** |
| 2† | 20 | B (blocked) |
| 3 | 21-22 | PR (print file) PU (punch file) |

† For tapes created before the blocked tape enhancement, this value will contain a blank.

- Subdirectory Entry Record

This is a 256-byte record containing subdirectory information.

- VFB Record

If the original spooled file was created by using a VFB job control statement, the unconverted VFB is written to tape. This record is 256 bytes in length.

- LCB Record

If the original spooled file was created by using an LCB job control statement, the unconverted LCB is written to tape. This record is 256 bytes in length. If the LCB was a multisector LCB, two LCB records are written.

■ Tape Mark

A tape mark is written after the last header record.

■ Data Records

Data records are variable blocked records having a maximum block size of 256 bytes with the following format:

| <u>Field Number</u> | <u>Location</u> | <u>Contents</u> |
|---------------------|-----------------|---|
| 1 | 1-2 | Variable block header (size of block) |
| 2 | 3-4 | Unused |
| 3 | 5-6 | Variable record header (size of record + 4) |
| 4 | 7-8 | Backward displacement to previous record (first record=80) |
| 5 | 9 | BCW command code |
| 6 | 10 | System user CCB indicator |
| 7 | 11-12 | Unused |
| 8 | 13-14 | Page number/card number |
| 9 | 15-n | Compressed record |

where:

BCW command code

Is the original BCW command code used by the program that created the output line or card.

System user CCB indicator

Is the value of the fourth byte of the CCB used by the program to create the output line or card.

Page number

Is the page number of the line that has been printed.

Fields 3-9 are repeated for each print or punch image.

■ End of File

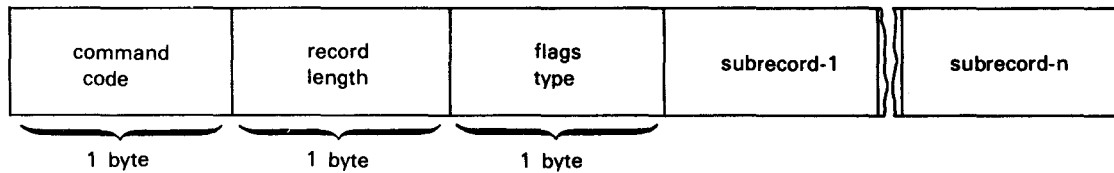
Each time a file is completed, two tape marks are written; then the tape is backspaced one block in case other files are to be redirected to this tape.

- End of Volume

If end of volume occurs, a tape mark is written, followed by an 80-byte record containing EOVI as the first four bytes. Two more tape marks are written.

- Compressed Records

The spooler creates compressed records on tape by using a data compression/expansion procedure. The format of a compressed record is:



where:

Command code

Is the BCW code used to originally write the record.

Record length

Is the sum total of all subrecords that comprise the print line or card image.

Flags type

Contains the compression record types:

X'25' = compressed record

X'24' = noncompressed record

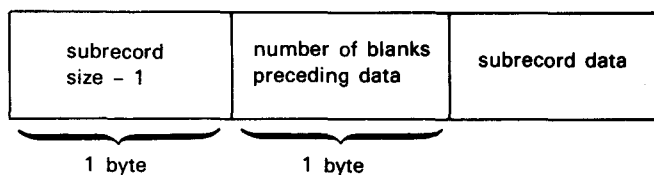
In addition, the following values and flags may be logically united ("ORed") to indicate:

X'80' = First page of new record

X'40' = Original request was system user.

- Subrecords

The format of a subrecord is:





Appendix B. Format of Output Writer Disk and Format Label Diskette

■ Header Record

Each output file that you redirect to a disk or diskette is indexed in \$Y\$POOL and saved in a sequentially numbered \$SPOOL file. The file is composed of at least one header record. This record is the 256-byte subdirectory entry. In addition, vertical format buffer (VFB) and load code buffer (LCB) records may also exist.

- Subdirectory Entry Record

This is a 256-byte record containing subdirectory information.

- VFB Record

If the original spooled file was created by using a VFB job control statement, the unconverted VFB is written to disk/diskette. This record is 256 bytes in length.

- LCB Record

If the original spooled file was created by using an LCB job control statement, the unconverted LCB is written to disk/diskette. This record is 256 bytes in length.

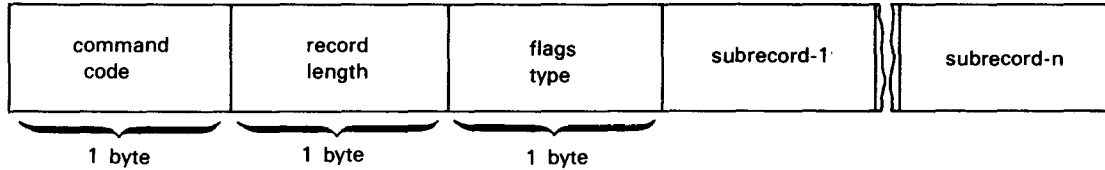
■ Data Record

A data record may contain more than one print or punch image. Fields 2 and 4 are repeated for as many compressed records as can be inserted into a 1024-byte block. A data record has the following format:

| <u>Field Number</u> | <u>Location</u> | <u>Contents</u> |
|---------------------|-----------------|---------------------------|
| 1 | 1-2 | Data block size |
| 2 | 3-4 | Page or card number |
| 3 | 5 | System user CCB indicator |
| 4 | 6-n | Compressed record |

■ Compressed Records

The spooler creates compressed records on disk and diskette by using a data compression/expansion procedure. The format of a compressed record is:



where:

Command code

Is the BCW code used to originally write the record.

Record length

Is the sum total of all subrecords that comprise the print line or card image.

Flags type

Contains the compression record types:

X'25' = compressed record

X'24' = noncompressed record

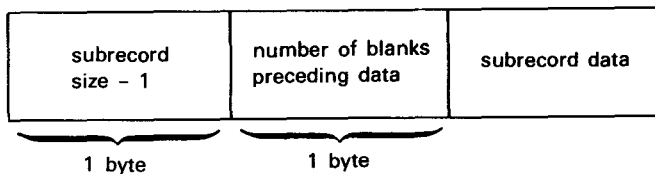
In addition, the following values and flags may be logically united ("ORed") to indicate:

X'80' = First page of new record

X'40' = Original request was system user.

■ Subrecords

The format of a subrecord is:



NOTES:

1. Diskette files using the extend or combined file processing must not use spooling.
2. Disk files require dynamic buffer management.

**Appendix C. Operator's Spooling
Reference**

C. OPERATOR'S SPOOLING REFERENCE

| | | |
|---------------|---|-------|
| C.1. | OVERVIEW | C-3 |
| C.2. | GENERAL SPOOLING COMMANDS AND MESSAGES | C-4 |
| C.2.1. | Displaying the Status of Spool Subfiles (DISPLAY ACT/SPL/CNSLG) | C-6 |
| C.2.2. | Displaying the Spooling Mode of Operation (DISPLAY SPL,STATUS) | C-11 |
| C.2.3. | Deleting Closed Spool Subfiles (DELETE) | C-12 |
| C.2.4. | Placing Spool Subfiles on Hold (HOLD) | C-13 |
| C.2.5. | Releasing Spool Subfiles (BEGIN) | C-15 |
| C.2.6. | Breakpointing Active Spool Subfiles (BRKPT) | C-17 |
| C.2.6A. | Changing the Device Type and/or the Number of Copies for Spooled Files (CH SPL) | C-19 |
| C.2.7. | Changing the System Spooling Criteria (SET SPL,BURST/NBURST) | C-20b |
| C.2.8. | Changing the System Accumulation of Job Log and Workstation Log Files (SET SPL,DUMP/ENDDUMP) | C-20c |
| C.2.9. | Changing the Spool File Printout Format (SET SPL,HEADER/NOHDR) | C-20c |
| C.2.10. | Changing the Job Log and Workstation Log File Print Specifications (SET SPL,NOACT/NOLOG/NOPRINT/PRINT) | C-21 |
| C.2.11. | Changing the Console Log and Workstation Log File Specifications (SET SPL,CNSLG) | C-22 |
| C.2.12. | Suppress or Display Forms Change Message (SET SPL,TEST/NOTEST) | C-24 |
| C.3. | INPUT READER COMMAND | C-24 |
| C.4. | OUTPUT WRITER COMMANDS AND MESSAGES | C-26 |
| C.4.1. | Loading an Output Writer (PR/PU/PD) | C-27 |
| C.4.2. | Loading an Output Writer for Nonauxiliary Destined Print Files Created on Behalf of a Workstation User (PR,UID) | C-28 |
| C.4.3. | Loading an Output Writer for Output Destined for a Workstation Auxiliary Printer (RP,UID) | C-29 |
| C.4.4. | Directing the Operation of an Output Writer (Messages) | C-32 |
| C.5. | SPOOL FILE PROCESSING AIDS | C-34 |
| C.5.1. | Saving the Accumulated Job and Workstation Log in SYSLOG (SL\$LOG Program) | C-34 |
| C.5.1.1. | Using Tape for the Job and Workstation Log SYSLOG File (RUN DUMPLOGT) | C-37 |
| C.5.1.2. | Using Disk for the Job and Workstation Log SYSLOG File (RUN DUMPLOG) | C-38 |
| C.5.1.3. | Adding Job and Workstation Log Subfiles to the SYSLOG File on Disk (RUN DUMPLOG) | C-38 |
| C.5.1.4. | Reinitializing the Job and Workstation Log SYSLOG File on Disk (RUN DUMPLOG) | C-39 |
| C.5.2. | Producing Job Accounting Reports (JOBLOG Program) | C-40 |
| C.5.2.1. | Running JOBLOG with Tape Input (RUN JBLOGT) | C-40 |
| C.5.2.2. | Running JOBLOG with Disk Input (RUN JBLOG) | C-41 |
| C.5.3. | Saving the Accumulated Console Log in SYSLOG (SL\$LOG Program) | C-43 |
| C.5.3.1. | Using Tape for the Console Log SYSLOG File (RUN DUMPLOGT) | C-43 |
| C.5.3.2. | Using Disk for the Console Log SYSLOG File (RUN DUMPLOG) | C-44 |
| C.5.3.3. | Adding Console Log Subfiles to the SYSLOG File on Disk (RUN DUMPLOG) | C-45 |
| C.5.3.4. | Reinitializing the Console Log SYSLOG File on Disk (RUN DUMPLOG) | C-45 |

C.1. OVERVIEW

OS/3 provides automatic spooling (simultaneous peripheral operations online) that buffers the input/output information to and from a spool file (SYSPOOL) on disk. This permits the input devices (card readers and diskette drives) and output devices (printers, card punches, and tape and diskette drives) to operate at their rated speed while the user jobs are processed, without regard to device contention. In OS/3, spooling requires no changes in any user programs.

The system spooling functions are accomplished with symbionts, programs, and files. You load and direct the spooling functions by entering the following commands at the console:

- General spooling
- Input reader
- Output writer

These commands are divided into two types:

1. Those that load and direct the spooling function via one command entry. The command and its parameters require no further direction.
2. Those that load the spooling function via one command entry followed by a series of solicited or unsolicited messages to further direct spooling.

Spool file processing aids provide special programs and job control streams that allow you to:

- save accumulated spool job and workstation log subfiles in the SYSLOG file;
- save accumulated spool console log subfiles in the SYSLOG file;
- produce a job accounting report; and
- use the subfiles for your own programs.

The following paragraphs describe general spooling commands, input reader commands, output writer commands, and spool file processing aids, including specific formats and message entries.

C.2. GENERAL SPOOLING COMMANDS AND MESSAGES

General spooling commands allow you to:

- display the status of spool subfiles (DISPLAY ACT/SPL/CNSLG);
- display the spooling mode of operation (DISPLAY SPL,STATUS);
- delete closed spool subfiles (DELETE);
- place spool subfiles on hold (HOLD);
- release spool subfiles (BEGIN);
- breakpoint active spool subfiles (BRKPT);
- change the system spooling criteria (SET SPL,BURST/NBURST);
- change the system accumulation of job and workstation log files (SET SPL,DUMP/ENDDUMP);
- change the spool file printout format (SET SPL,HEADER/NOHDR);
- change the job log file and workstation log file print specifications (SET SPL,NOACT/NOLOG/NOPRINT/PRINT);
- change the console log file and workstation log file specifications (SET SPL,CNSLG);
and
- suppress or display forms change message (SET SPL,TEST/NOTEST).

The formats for these commands may include:

- spool file directories (Table C-1) to identify one of the input or output directories that make up the spool file; and
- spool file modifiers (Table C-2) to further identify the subfiles that are being referenced.

Table C-1. Spool File Directories

| Directory | Subfile Function |
|----------------|--|
| DDPPR | Distributed data processing output is to printer. |
| DDPPU | Distributed data processing output is to card punch. |
| LOG | Job, workstation, and console log input and output is in designated log subfile. |
| <u>P</u> PRINT | Subfile output is to designated printer. |
| <u>P</u> PUNCH | Subfile output is to designated card punch or data set label diskette. |
| RBPIN | Remote batch processing input is from card reader. |
| RBPPR | Remote batch processing output is to printer. |
| RBPPU | Remote batch processing output is to card punch. |
| RDR | Subfile input is from designated card reader, tape, or data set label diskette. |

NOTE:

When ALL is entered in place of a directory, all directories are accessible.

Table C-2. Spool File Command Modifiers (Part 1 of 2)

| Modifier | Subfile Identification |
|---|---|
| <u>ACCT</u> = {acctno} * | Account number of up to four characters |
| <u>BNUMB</u> =binary jobno | Five-digit binary job-number. Used with BEGIN SPL, DELETE SPL, HOLD SPL, PR, PU, PD, and RP commands |
| <u>CART</u> = {cartridge-id} * | Print cartridge identification of up to eight characters |
| DDPID=host-id | DDP host (job initiator) file identification of up to four characters (BX function for currently loaded output writer only) |
| DEV= {770 776 789} | Any device of the type designated that is available |
| [, <u>COPIES</u> =nnn] | Modifiers for change spool only. |
| [, <u>DVC</u> = {770 776 PPC ANY CLASS1 CLASS2 CLASS3}] | Any device of the type designated that is available |
| [, <u>DVC</u> =AUX, <u>ID</u> = {* user-id}] | |
| [, <u>ID</u> = {* user-id}] | |



Table C-2. Spool File Command Modifiers (Part 2 of 2)

| Modifier | Subfile Identification |
|-------------------------------|---|
| <u>FILE</u> = {filename} * | File name of up to eight characters |
| <u>FORM</u> = {formname} * | Form name of up to eight characters |
| <u>ID</u> =remote-id | Remote batch file identification of up to six characters (output writer BX function only) |
| <u>JOB</u> = {jobname} * | Job name of up to eight characters |
| <u>LBL</u> =labelname | Label name of up to 8 characters for data set label diskette, up to 17 characters for card reader |
| <u>STEP</u> =stepno | Job step number of three characters (left-justified with zeros) |
| <u>VOL</u> =volno | Volume number of up to six characters (for diskette only) |

NOTE:

The modifier with an asterisk (*) is used to modify spooling commands that control output file processing. It specifies that the output writer is to group the contents of the spool file according to the modifier type specified, and process the groups on a first-in/first-out basis. This provides a complete spool file listing, grouped according to the modifier specified.

C.2.1. Displaying the Status of Spool Subfiles (DISPLAY ACT/SPL/CNSLG)

Function:

To display all requested information for either currently active or complete subfiles on the console.

The DISPLAY command is also used to cause a console display of the number of lines that have accumulated in the current console log subfile.

Format 1:

```

DISPLAY ACT [ ( ALL ) ] [ ,modifier-1, ...,modifier-n ]
             {
             DDPPR
             DDPPU
             PRINT
             PUNCH
             RBPPR
             RBPPU
             }

```

Format 2:

→ DISPLAY SPL [ALL] [,modifier-1,...,modifier-n]
DDPPR
DDPPU
LOG
PRINT
PUNCH
RBPPR
RBPPU
RBPIN
RDR



Format 3:

DISPLAY CNSLG

Positional Parameter 1:

ACT

Indicates that only the currently active subfiles are being displayed.

SPL

Indicates that only the completed subfiles are being displayed.

CNSLG

Indicates that only the console log is being displayed.

Positional Parameter 2:

Identifies one of the spool file directories permitted, as described in Table C-1.

Unless otherwise specified, all of the directories are accessible to the command. A directory should not be used with format 3; if entered, the command is ignored.

Positional Parameter 3:

modifier-1, . . . , modifier-n

Further identifies the subfiles being referenced, as described in Table C-2. Modifiers should not be used with format 3. If any are entered, the command is ignored.

Solicited message entries are required to direct and terminate the display when ACT or SPL is specified with the DISPLAY command. (Solicited messages are those messages input by the operator in direct response to an output message that requires a reply.)

If DISPLAY SPL is used and no subfiles exist in the requested directory, the following message is displayed.

```
DI03 SPOOL FILE EMPTY
```

If one or more subfiles exist for the requested display, the following message is displayed to show the status of the subfiles.

```
DI01 status FILES=ffff PAGES=pppp CARDS=cccc
```

where:

status

Is QUEUED, HOLD, or IN PROGRESS.

ffff

Is the number of files.

pppp

Is the number of pages, excluding log files.

cccc

Is the number of punch images.

The following message is also displayed.

```
DIØ2 SPOOL FILE DETAILS? ***Y,N,Q,H,I,S,SQ,SH,SI***
```

The operator responds with a solicited message indicating the type of display:

- Y Displays all spool details
- N Does not continue display
- Q Displays all queued files
- H Displays all files in hold
- I Displays all files currently being processed by output writers. (If RDR is specified, displays all files currently being processed by programs.)
- s Abbreviated display of all files
- sq Abbreviated display of all queued files
- sh Abbreviated display of all files being held
- si Abbreviated display of all files in progress

Depending on the response, displays are produced as follows:

- If the response was Y, Q, H, or I and a directory other than RDR was specified, a display is produced in the following format:

```
DI04 JOB-NAME jobname FILE filename STATUS file-status
DI05 TOTAL- { PAGES } nnnnn REMOTE-ID xxxxxx COPIES nnn
             { CARDS }
             { LINES }
DI06 STEP-NUMBER nnn DEVICE-TYPE xxxxxx BREAKPOINT { Y }
                                                    { N }
DI07 BAND-NAME xxxxxxxxx FORM-NAME xxxxxxxxx ACCT xxxx
DI07A JOB-NUMBER nnnnn CURR-PAGE nnnnn
DI08 PROGRAM-NAME xxxxxxxxx CONTINUE? ***Y,N***
```

where:

CURR-PAGE nnnnn

Displays the page where printing begins when an output-writer is loaded. Interrupting a 200-page print with a 00 PR ST command closes and queues the file. If the printing stops at page 100, then CURR-PAGE is equal to 100, and when another output-writer is loaded, the printing begins at page 100. If the output writer has not been interrupted, CURR-PAGE will be 00000.

- If the response was Y, Q, H, or I and the RDR directory was specified, a display is produced in the following format:

```
DI09 RDR FILE nnnnn CARDS LBL xxxxxxxxxxxxxxxxxxxx VOL vvvvvv
DI10 CONTINUE? ***Y,N***
```

NOTE:

The VOL display appears only when the RDR directory file was spooled in from a diskette.

- If the response was S, SQ, SI, or SH, abbreviated displays in the following format are produced for up to five files at a time:

```
DI11 JOB=jobname PROG=programname FO=formname { PA } = nnnnn
                                                { CA }
                                                { LI }

ST=stepnumber
```

After five lines of DI11 messages have been produced, the following message appears on the console screen:

```
DI12 CONTINUE SUMMARY? ***Y,N***
```

- ↓
- If the response was S, SQ, SI, or SH and the RDR directory was specified, abbreviated displays in the following format are produced for up to five files at a time:

```
DI09 RDR FILE nnnnn CARDS LBL xxxxxxxxxxxxxxxxx VOL vvvvv
```

After five lines of DI09 messages have been produced, the following message appears on the console screen:



```
DI12 CONTINUE SUMMARY? ***Y,N***
```

NOTE:

The VOL display appears only when the RDR directory file was spooled in from a diskette.

↑

The operator responds to DI08, DI10, or DI12 with Y to continue the display or N to terminate the display.

If DISPLAY ACT is used and no modifiers are entered, the number of subfiles currently being created for the specified directory is displayed and the operator is given the option (Y or N) of displaying information about those files similar to that produced with the DISPLAY SPL command (when the reply to SPOOL FILE DETAILS? is Y, Q, H, or I). If modifiers are entered, the spool file details are displayed immediately.

Examples:**1. Operator keyin:**

```
DI SPL,ALL
```

Function requested:

Display all completed subfiles in the spool file.

2. Operator keyin:

```
DI SPL,PRINT
```

Function requested:

Display all completed subfiles in the print directory.

3. Operator keyin:

```
DI SPL,PUNCH,JOB=XYZ
```

Function requested:

Display all completed subfiles associated with job XYZ that are in the punch directory.

4. Operator keyin:

```
DI SPL,PR,FO=ABC,CART=48-BUS
```

Function requested:

Display all completed files in the print directory that require use of a form named ABC and a 48-character business print cartridge.

5. Operator keyin:

```
DI ACT
```

Function requested:

Display the number of active files (those being created) and question if details on subfiles in the directory are desired. If so, respond with Y to display the subdirectory entry.

6. Operator keyin:

```
DI CN
```

Function requested:

Display the number of lines that have accumulated in the current console log subfile.



C.2.2. Displaying the Spooling Mode of Operation (DISPLAY SPL,STATUS)

Function:

To display the spooling mode of operation currently in effect.

Format:

DISPLAY SPL,STATUS

Positional Parameter 1:

SPL

Indicates the DISPLAY command is directed to the spooling system.

Positional Parameter 2:

STATUS

Indicates that the current operating mode status is to be displayed.

When the DISPLAY SPL,STATUS command is entered, the system displays a message in the following format:

```
DI13 MODE={BU} CRITERIA: { JOB      } = { criteria-value }
                   {NB}      { FILE      } = { blank-screen  }
                               { CART     }
                               { ACCT     }
                               { DEVC     }
                               { FORM     }
                               { NONE     }
                               { blank-screen }
```

where:

BU

Is burst mode.

NB

Is nonburst mode.

CRITERIA

Is the spool file modifier providing selection criteria for the current operating mode. Selection criteria are not available in nonburst mode; the criteria display is always blank. When operating in burst mode without selection criteria specified, selection is by priority (the default condition) and NONE is displayed.

criteria-value

Is the specific value in effect for the criteria named. When operating in nonburst mode or in burst mode without criteria specified, the criteria-value display is blank.

The system then displays a message that indicates the percentage of the spool file that is still available. This message has the following format:

```
DI13A PERCENTAGE OF SPOOL FILE NOW AVAILABLE nnn%
```

C.2.3. Deleting Closed Spool Subfiles (DELETE)

Function:

To delete closed spool subfiles from the spool file. Only closed subfiles (queued or on hold) are deleted; active subfiles and subfiles currently being processed cannot be deleted.

Format:

```
DELETE SPL, ( ALL ) [,modifier-1,...,modifier-n]
             ( DDPPR )
             ( DDPPU )
             ( LOG )
             ( PRINT )
             ( PUNCH )
             ( RBPPR )
             ( RBPPU )
             ( RDR )
```

Positional Parameter 1:

SPL

Indicates that only closed subfiles in the directory named are being deleted.

Positional Parameter 2:

Identifies one of the spool file directories permitted, as described in Table C-1. Entering LOG does not affect logs destined for remote batch or distributed data processing jobs. If ALL is entered, all directories are affected.

Positional Parameter 3:

modifier-1,...,modifier-n

Further identifies the subfiles being referenced, as described in Table C-2.

When the DELETE SPL command is entered for a complete directory or ALL is specified, the system displays a message in the following format:

```
DE05 DELETING ENTIRE xxxxx QUEUE. CONTINUE (Y/N).
```

where xxxxx is the name of the queue that is to be deleted. This message acts as a precaution against deleting the entire spool file or a particular queue in the event that an error was made in the command keyin.

Enter Y if the command keyin is correct; enter N if it is incorrect. ↓

If Y is entered, the spool subfiles are deleted and the system displays a message in following format:

```
DE01 xxx SPOOL FILES DELETED
```

where xxx is the number of spool files deleted.

If N is entered, the spool subfiles are not deleted and the system displays a DE01 message that indicates no spool files were deleted. ↑

Examples:

1. Operator keyin:

```
DE SPL,ALL,JOB=ABC
```

Function requested:

Delete all closed subfiles associated with job ABC.



2. Operator keyin:

DE SPL,PRINT,JOB=XYZ

Function requested:

Delete all closed subfiles associated with job XYZ that are contained in the print directory.

C.2.4. Placing Spool Subfiles on Hold (HOLD)

Function:

To place active and closed spool subfiles in a hold condition. Subfiles in this condition are not available for processing by an output writer or job control symbiont until they are subsequently released by the operator using the BEGIN command (C.2.5).

Format 1:

HOLD SPL [ALL] [.modifier-1, ..., modifier-n]

| | |
|---|----------------|
| } | DDPPR |
| | DDPPU |
| | LOG |
| | <u>P</u> PRINT |
| | <u>P</u> PUNCH |
| | RBPPR |
| | RBPPU |
| | RDR |

Format 2:

HOLD { ACT } [ALL]

| | |
|---|----------------|
| } | LOG |
| | <u>P</u> PRINT |
| | <u>P</u> PUNCH |

Positional Parameter 1:

SPL

Indicates that only closed subfiles in the directory named will immediately be placed in a hold condition.

ACT

Indicates that only those subfiles that are currently active in the directory named will be placed in the hold condition when closed.

SPQ

Indicates that both currently active and closed subfiles are being referenced. If specified, all closed spool subfiles in the directory named will be immediately placed in the hold condition and all of the directory's active and future spool subfiles will be placed in the hold condition when closed.

Positional Parameter 2:

Identifies one of the spool file directories permitted, as described in Table C-1. Entering LOG does not affect logs destined for remote batch or distributed data processing jobs. If no parameter is specified or ALL is entered, all directories are affected.

Positional Parameter 3:

`modifier-1, . . . , modifier-n`

Further identifies the subfiles being referenced, as described in Table C-2. Modifiers should not be used with format 2. If any are entered, the command is ignored.

When the HOLD SPL or HOLD SPQ commands are entered, the system displays a message in the following format:

```
H001 xxx SPOOL FILES HELD
```

where xxx is the number of spool files held.

Examples:**1. Operator keyin:**

```
HO SPL,PRINT,CART=48-SCI
```

Function requested:

Hold all closed subfiles in the print directory that require the use of a 48-character scientific print cartridge.

2. Operator keyin:

```
HO SPL,RDR,LBL=TEST
```

Function requested:

Hold all closed subfiles in the reader directory that contain a label named TEST.

Format 3:

```

BEGIN {ACT} {ALL}
      {SPQ} {LOG}
           {PRINT}
           {PUNCH}

```

Positional Parameter 1:

SPL

Indicates that the hold status only will be removed from all closed spool subfiles in the directory named.

ACT

Indicates that only those subfiles that are currently active in the directory named will not be placed in the hold condition when closed.

SPQ

Indicates that both currently active and closed subfiles are being referenced. If specified, the hold status will be removed from all closed spool subfiles in the directory named, and the directory's active and future spool subfiles will not be placed in the hold condition when closed.

Positional Parameter 2:

Identifies one of the spool file directories permitted, as described in Table C-1. Entering LOG does not affect logs destined for remote batch or distributed data processing jobs. If no parameter is specified or ALL is entered, all directories are affected.

Positional Parameter 3:

modifier-1,...,modifier-n

Further identifies the subfiles being referenced, as described in Table C-2. Modifiers should not be used with format 3. If any are entered, the command is ignored.

Positional Parameter 4:

OUT=did

Specifies the designated device (did) that the output is to be printed on.

OUT=NO

Specifies that the output writer is not to be loaded. If this parameter is specified, the command keyin for format 2 can be up to 60 characters in length, including commas.

When the BEGIN SPL or BEGIN SPQ commands are entered, the system displays a message in the following format:

```
BE01 xxx SPOOL FILES RELEASED
```

where xxx is the number of spool files released.

Examples:

1. Operator keyin:

```
BE SPL,PRINT,CART=48-SCI
```

Function requested:

Release all the subfiles in the print directory that require the use of a 48-character scientific print cartridge and that are currently in the hold condition. This command countermands the HOLD command presented as example 1 in C.2.4.

2. Operator keyin:

```
BE SPL,RDR,LBL=TEST
```

Function requested:

Release all the subfiles in the reader directory that contain a label named TEST and are currently in the hold condition. This command countermands the HOLD command presented as example 2 in C.2.4.



C.2.6. Breakpointing Active Spool Subfiles (BRKPT)

Function:

To breakpoint subfiles currently being created by a job or by the console logging function. It effectively closes a subfile or subfiles so that the information created in the subfile or subfiles is available for processing by an output writer. A new subfile or subfiles are created that will contain the remainder of the information being created by the job.

The operator should use this function when warning messages indicating the spool file is nearly depleted are displayed on the console or in orderly shutdown at end of day.

NOTES:

1. *After the breakpoint has been taken for printer or punch subfiles, the operator should call in an output writer in burst mode.*
2. *After the breakpoint has been taken for the console log subfile, the output writer is brought in automatically. The output writer recognizes the console log subfile and prints it. The breakpointed console log subfile contains records collected from the last breakpoint or IPL of the system (provided you do not specify console log recovery during the IPL). Console log collection must be set to ON with option to permit printing, as specified via the SET SPL,CN command currently in effect.*
3. *Diskette subfiles may not be breakpointed.*
4. *After the BR command has been entered, a message telling the operator whether the breakpoint was successful will appear on the console.*

Format 1:

```
BRKPT {P} . {PRINT} . JOB=jobname[,modifier-1,...,modifier-n][,HOLD]
      {I} {PUNCH}
```

Format 2:

```
BRKPT CNSLG [ ,OUT={TAPE
                {DISK
                {DISKETTE} } ] [,HOLD]
```

Positional Parameter 1:

P

Indicates that the subfile is to be breakpointed at the end of the page.

I

Indicates that the subfile is to be breakpointed immediately.

CN S L G

Indicates that the console log subfile is to be breakpointed.

Positional Parameter 2:

PRINT and PUNCH

Identifies the type of subfile being breakpointed as a print (PR) or punch (PU) subfile.

OUT={ TAPE
 DISK
 DISKETTE }

Indicates that the console log is redirected to the output device specified: tape, disk, or diskette.

Positional Parameter 3:

JOB=jobname

Identifies the name of the job in the subfile to be breakpointed.

Positional Parameter 4:

modifier-1,...,modifier-n

Further identifies the subfiles being referenced, as described in Table C-2. Modifiers should not be used with format 2. If any are entered, the command is ignored.

Positional Parameter 5:

HOLD

Indicates that the subfile is to be held after the breakpoint. When specified for a print file, all subsequent subfiles for the breakpointed file will also be placed in a hold condition.

Examples:

1. Operator keyin:

BR I,PU,JOB=ABC

Function requested:

Breakpoint the punch subfile that is currently being created for the job named ABC.

2. Operator keyin:

BR P,PR,FILE=INVCONT,JOB=MYOWN

Function requested:

At the end of the page, breakpoint the print subfile named INVCONT that is currently being created for the job named MYOWN.

3. Operator keyin:

BR CN

Function requested:

Breakpoint the console log subfile.

C.2.6A. Changing the Device Type and/or the Number of Copies for Spooled Files
(CH SPL)

Function:

To change the device type and/or the number of copies printed for a spooled file. This command changes closed spool subfiles of either queued or held status, but not subfiles of active status. The parameters for changing the copies and/or devices are required, as there are no defaults.

Format:

```

CHANGE SPL, { ALL
              LOG
              PRINT } [,modifier-1 ... modifier-n]

[ ,COPIES=nnn]
[ [ [ ,DVC=( 770
          776
          PPC
          ANY
          CLASS1
          CLASS2
          CLASS3 ) ] ] ]
[ [ ,DVC=AUX, ID= { *
                  user-id } ] ]
[ [ ,ID= { *
         user-id } ] ]

```

↓

Positional Parameter 1:

ALL
LOG
PRINT

Specifies the directories in which to change the device type or number of copies. ALL includes LOG and PRINT queues only.

Positional Parameter 2:

DVC={ 770
776
PPC
ANY
CLASS1
CLASS2
CLASS3 }

Spool parameters are not hard assigned to a device specified on the DVC parameter, but logically assigned to a device type. Using the DVC parameter permits logically switching the files spooled to print on another device or class of device.

Positional Parameter 3:

DVC=AUX, ID={ *
user-id }

This parameter enables printing on an auxiliary printer of the user-id. The system displays an error message if the user-id is not specified with this parameter.

When you enter the CH SPL command, the following message is displayed:

CH01 nnn SPOOL FILES CHANGED

where:

nnn

Is the number of spooled files changed by your CH SPL command.

Examples:

1. Operator keyin:

CH SPL,ALL,DVC=PPC

This example sends all log and print files for the user to the device type PPC printer.

↑

2. Operator keyin: ↓

```
CH SPL,ALL,JOB=TEST,DEV=770,DVC=PPC
```

This example changes only the files originally destined for the 770 printer. (The original JCL indicated DEV 28.) It directs the spooled files to a PPC.

3. Operator keyin:

```
CH SPL,ALL,JOB=YOURJOB,DVC=CLASS2
```

This example redirects all spooled output to a class 2 PPC printer.

4. Operator keyin:

```
CH SPL,ALL,JOB=YOURJOB,DVC=ANY
```

This example enables printing of spooled files on any printer.

5. Operator keyin:

```
CH SPL,ALL,JOB=MYJOB,DVC=AUX,ID=*
```

This example redirects the spooled output to the aux-printer attached to the current workstation/terminal. The original spool file could have been created without using the // ROUTE in the // DVC 20 // LFD sequence for the printer.

6. Operator keyin:

```
CH SPL,ALL,JOB=MYJOB,DVC=AUX,ID=YOURID
```

This example redirects the spooled output to the aux-printer attached to the workstation/terminal of the user indicated by the parameter ID =. The original spool file could have been created without using the // ROUTE in the // DVC 20 // LFD sequence for the printer.

7. Operator keyin:

```
CH SPL,PR,JOB=MYJOB,COPIES=2
```

This example changes the number of copies requested for one or more spooled files currently queued or on hold. This overrides the number of copies specified in the JCL statement. ↑

C.2.7. Changing the System Spooling Criteria (SET SPL,BURST/NBURST)

Function:

To change the system operating mode for processing output files (burst or nonburst). Burst mode processing allows queued or breakpointed files from job steps to be processed by the output writer before the job they are associated with has terminated. Nonburst mode processing requires that a job be terminated before any of the files associated with the job can be output by the output writer. Operating in the burst mode, therefore, permits the output produced by a single job to be split between two or more like output devices. This tends to speed up throughput for a job, but also creates a confusion factor for the operator if more than one job is being run in the system. On the other hand, operating in the nonburst mode may slow down throughput, but ensures that all like output associated with one job is produced on one device (e.g., all print files are printed on the same printer).

When a SET SPL,BURST/NBURST command is issued, the system spooling operating mode is changed immediately and all future output writers loaded by the spooling system will function accordingly. Previously loaded output writers, however, continue to function using the operating mode established for them either when they were loaded or via operator command (C.4). This SET command should not be confused with the BURST and NBURST functions of the output writer commands.

Format:

```
SET SPL, { BURST } [ ,modifier ]  
         { NBURST }
```

Positional Parameter 1:

SPL

Indicates that the SET command is directed to the spooling system.

Positional Parameter 2:

BURST

Specifies that the spooling system is to operate in burst mode.

NBURST

Specifies that the spooling system is to operate in nonburst mode.

Positional Parameter 3:

modifier

Further identifies the subfiles being referenced, as described in Table C-2.

C.2.8. Changing the System Accumulation of Job Log and Workstation Log Files (SET SPL,DUMP/ENDDUMP)

Function:

To change the method of processing the job log and workstation log subfiles produced in the system. The initial method of processing is specified at SYSGEN time.

As each job in the system or each workstation session is terminated, its associated spool log subfile is closed and, according to the printing option currently in effect, automatically output to a high-speed printer as soon as an output writer becomes available. (For workstation logs, note that the SET SPL,CN,ON command must also be in effect.) The log subfile is then either deleted from the spool LOG subfile or accumulated for future processing, depending on the SYSGEN option selected. This command permits you to override the SYSGEN selected option.

Format:

```
SET SPL, { DUMP  
          { ENDDUMP }
```

Positional Parameter 1:

SPL

Indicates the SET command is directed to the spooling system.

Positional Parameter 2:

DUMP

Specifies that job log and workstation log files are to be accumulated for later routing to the SYSLOG file on tape or disk.

ENDDUMP

Specifies that job log and workstation log file accumulation is to end.

C.2.9. Changing the Spool File Printout Format (SET SPL,HEADER/NOHDR)

Function:

To enable or suppress the printing of a 3-page header that precedes each spooled print file. The spool file printout format is initially established at SYSGEN time.



Format:

```
SET SPL, {HEADER}
          {NOHDR}
```

Positional Parameter 1:

SPL

Indicates the SET command is directed to the spooling system.

Positional Parameter 2:

HEADER

Indicates that a 3-page header is to precede the printing of each spooled print file.

NOHDR

Indicates that spooled output files are to be printed without headers.

C.2.10. Changing the Job Log and Workstation Log File Print Specifications (SET SPL,NOACT/NOLOG/NOPRINT/PRINT)

Function:

To select what job log and workstation log records are to be printed. The specified job log records are printed at the end of each job. The initial job log print specification is established at SYSGEN time. When specified by the PRINT parameter and the workstation user LOGON command, workstation log records are printed when the workstation user breakpoints them or ends the current workstation session. (For workstation logs, note that the SET SPL,CN,ON command must currently be in effect.)

Format:

```
SET SPL, {NOACT}
          {NOLOG}
          {NOPRINT}
          {PRINT}
```

Positional Parameter 1:

SPL

Indicates the SET command is directed to the spooling system.

Positional Parameter 2:

NOACT

Indicates that accounting (A) records are not to be printed.

NOLOG

Indicates that log (L) records and workstation (W) records are not to be printed.

NOPRINT

Indicates that job accounting (A) and log (L) records and workstation (W) records are not to be printed.

PRINT

Indicates that job accounting (A) and log (L) records and workstation (W) records are to be printed.

NOTE:

User snap and cancel dumps are always printed at the end of each job.

**C.2.11. Changing the Console Log and Workstation Log File Specifications
(SET SPL,CNSLG)**

Function:

To allow the operator to override the SYSGEN-selected options to turn the console log and workstation log record collection function on or off, to delete or accumulate the console log for transfer to SYSLOG, and to permit or suppress printing the console log when it is breakpointed.

Format:

```
SET SPL,CNSLG [ {ON} ] [ {RETAIN} ] [ {PRINT} ]
               [ {OFF} ] [ {DELETE} ] [ {NOPRINT} ]
```

Positional Parameter 1:

SPL

Directs the SET command to the spooling system.

Positional Parameter 2:

CNSLG

Further directs the command to the console log subfile.

Positional Parameter 3:

ON

Indicates that the console log (C) records and workstation log (W) records are to be collected. ←

OFF

Indicates that the console log (C) records and workstation log (W) records are not to be collected. ←

Positional Parameter 4:

RETAIN

Indicates that the console log subfile is to be accumulated for transfer to SYSLOG.

DELETE

Indicates that the console log subfile is to be deleted after printing (not accumulated).

Positional Parameter 5:

PRINT

Indicates that the console log subfile is permitted to be printed when breakpointed.

NOPRINT

Indicates that the console log subfile is not to be printed when breakpointed.

NOTES:

1. *Positional parameter 3 must be set to ON to accumulate or print the console log or to accumulate or print the workstation log (via SET SPL,PRINT and SET SPL,DUMP commands respectively).*
2. *If positional parameter 3, 4, or 5 is omitted, the console log subfile remains in the same condition or mode for that parameter that it was in before the command.*
3. *A blinking marker symbol (▽) in the rightmost position of the console line indicates that the message was not written to the console log.*

←

C.2.12. Suppress or Display Forms Change Message (SET SPL,TEST/NOTEST)

Function:

To set the option directing the output writer to display or suppress the message that informs the operator of a forms change and to ask whether test lines on the first page of output should be printed. On the test lines page, all numeric characters are replaced with 9's and all alphabetic characters are replaced with Z's.

This command may be used to override the forms change message setting that was established during system generation (SPOOLTEST statement). If the forms change message setting was specified in the job control stream (// SPL statement), this command may be used to override the job control option that calls for the test lines message but may not be used to override the job control option that suppresses the forms change message.

Format:

```
SET SPL. {TEST }  
          {NOTEST }
```

Positional Parameter 1:

SPL

Indicates the SET command is directed to the spooling system.

Positional Parameter 2:

TEST

Indicates the test lines message should be displayed on the console when applicable.

NOTEST

Indicates the test lines message should not be displayed on the console.

C.3. INPUT READER COMMAND

The operator loads the input reader by the IN command entered at the console. No other commands or messages are required to support the input reader spooling function in OS/3.

Function:

To cause the operating system to load an input reader symbiont to transfer a card, tape, or data set label diskette file into the RDR spool directory.

Jobs or data files to be spooled from cards must be preceded by a // DATA job control statement. This provides the spool file label to be used later when the data is required or when the job is filed or run. The punched cards are read into the spool file until a // FIN or another // DATA card is detected.

Jobs or data files to be spooled from a single volume unlabeled tape must be preceded by a // DATA job control statement. This provides the spool file label to be used when the data is required or when the job is to be filed or run.

When jobs or data files are to be spooled from diskettes, the // DATA and // FIN job control statements should not be used. If any are entered, they are included as records in the data file. The data set label from the diskette provides the spool file label and the /* statement ends the data file. Note that when a job is input from diskette, it must be single volume to be filed or run intact later on.

A // RUN job control statement may appear anywhere within the user card, tape, or diskette file to cause the input reader to schedule the specified job. This job is scheduled, however, only when and if the spooled subfile is successfully entered and closed in the system spool file. Note that the // DATA job control statements may contain an IGNORE parameter (e.g., when converting job control streams from OS/4 to OS/3) to permit a // RUN statement to be treated as a normal input card.

Any number of input readers, up to the number of card readers, tapes, and diskettes in the system, can be executing simultaneously.

Format 1 (card reader):

```
IN((did)) [ { 51 } ]
           [ { 66 } ]
```

NOTE:

If no device (did) is specified, the first available card reader, as defined when the system was generated (SYSRDR), is expected to contain the card file to be spooled.

Format 2 (single volume unlabeled tape):

```
IN(did)
```

Format 3 (data set label diskette):

```
IN([did],label) [RETAI]N[.DELE]TE
```

NOTE:

If the data file is on a diskette, format 3 must be used.

Positional Parameter 1:

51

Indicates input reader is to accept a file that resides on 51-column cards.

66

Indicates input reader is to accept a file that resides on 66-column cards.

If omitted, 80-column (or 96-column, if configured in the system) punched cards are read.

RETAIN

Indicates that the spool file is to be retained after the file has been processed. This parameter is to be used only for single-volume input. The retained file remains in the spool file, available for additional processing, until deleted via the DELETE command or, if the DELETE parameter is used, in a subsequent introduction of a spooled subfile having the same file label.

Positional Parameter 2:**DELETE**

Indicates that spool subfiles with the same label name are to be deleted before creating the new subfile.

The RETAIN and DELETE parameters are both optional and may be entered in any order.

C.4. OUTPUT WRITER COMMANDS AND MESSAGES

Output writers are loaded under three conditions:

1. Automatically by the system, as required, up to the number of output devices available for their use. The output writers are set in either burst or nonburst mode, according to the mode established at SYSGEN time.
2. Automatically by the operator when the BEGIN spooling command is entered. The output writers are automatically set to burst mode by this command.
3. Manually by the operator under certain conditions, such as the system set in nonburst mode, in order to have files printed. Other conditions include:
 - Warm start (i.e., recovering files at IPL) if set to nonburst at SYSGEN
 - BR function if system is in nonburst mode
 - STOP or HALT function previously entered from console and printing of files for the job is incomplete
 - IN function when reintroducing redirected tape, disk, or format label diskette output

After the output writer is loaded, unsolicited and solicited messages are entered by the operator to change the current symbiont function.

C.4.1. Loading an Output Writer (PR/PU/PD)

Function:

To manually load a print, punch, or diskette output writer symbiont (PR, PU, PD). When entering this command, you may include a function as a parameter to direct the symbiont.

Format:

```
{ PR } [(did)] [function-code][.modifier-1,...,modifier-n]
{ PU }
{ PD }
```

Command Code:

PR

Indicates that the request is for an output writer to process print files.

PU

Indicates that the request is for an output writer to process punch files.

PD

Indicates that the request is for an output writer to process diskette files in data set label mode.

Positional Parameter 1:

(did)

Designates the address of the output device to be used by the PR or PU output writer.

To redirect print or punch output for temporary storage, you specify the address of a tape, disk, or diskette unit with a PR or PU output writer. All output files processed by the output writer being called are redirected to the specified device. The diskette is written in format label mode for later printing or punching; it cannot be used as input to the input reader.

If omitted, it is assumed that only one device is available for assignment in the system, or the system should select the first available device, and the output writer, by default, uses that device.

No did is needed if PD is used. Diskette files are written in data set label mode to the location on the diskette defined in the job control device assignment set for the file.

Positional Parameter 2:`function-code`

Identifies the mode of operation and processing criteria under which the output writer is to function. The various function codes that may be specified in this parameter location are listed and described in Table C-3. If omitted, the output writer is loaded in the mode (nonburst/burst) indicated by SYSGEN.

Positional Parameter 3:`modifier-1, ..., modifier-n`

Further identifies the subfiles being referenced, as described in Table C-2.

NOTE:

Function code keyins cannot exceed 28 characters in length, including commas.

**C.4.2. Loading an Output Writer for Nonauxiliary Destined Print Files
Created on Behalf of a Workstation User (PR,UID)****Function:**

To manually load a print output writer symbiont that will print nonauxiliary destined print files created on behalf of the workstation user identified by UID.

Format:

```
PR [function-code],UID=user-id[,modifier-1, ..., modifier-n]
```

Command Code:`PR`

Indicates that the request is for an output writer to process print files.

Positional Parameter 1:`function-code`

Identifies the mode of operation and processing criteria under which the output writer is to function. The various function codes are listed and described in Table C-3. If omitted, the output writer is loaded in the mode (nonburst/burst) indicated by SYSGEN.

Positional Parameter 2:`UID=user-id`

Identifies the workstation user on whose behalf the print files were created. The user-id is the 1- to 6-character, left-justified identification that was entered with the LOGON command at the workstation.

Positional Parameter 3:`modifier-1, . . . , modifier-n`

Further identifies the subfiles being referenced, as described in Table C-2.

C.4.3. Loading an Output Writer for Output Destined for a Workstation Auxiliary Printer (RP,UID)**Function:**

To manually load a print output writer symbiont that will print output, destined for a workstation auxiliary printer, at the central site rather than at the auxiliary printer.

Format:`RP BX,UID=user-id[modifier-1, . . . , modifier-n]`**Command Code:**`RP`

Indicates that the request is for an output writer to process print files destined for a workstation auxiliary printer.

Positional Parameter 1:`BX`

Indicates that the output writer is to function in the burst mode as described in Table C-3.

Positional Parameter 2:`UID=user-id`

Identifies the workstation user on whose behalf the print files were created. The user-id is a 1- to 6-character, left-justified identification that was entered with the LOGON command at the workstation.

Positional Parameter 3:`modifier-1, . . . , modifier-n`

Further identifies the subfiles being referenced, as described in Table C-2.

Table C-3. Output Writer Function Codes (Part 1 of 2)

| Function | Meaning |
|--|--|
| <p>{ BURST } { BX } [modifier-1, . . . , modifier-3]</p> | <p>Output writer is to function in the burst mode, as optionally further qualified by modifiers 1 through 3. The modifiers that may be specified are listed and described in Table C-2. (STEP, LBL, and VOL are not used with BU and BX.) Burst mode processing is described in C.2.7. If BX is entered with modifiers, the output writer terminates after processing all files that satisfy the modifiers. If BU is entered, the output writer requests another function if more files exist that do not satisfy the modifiers.</p> |
| <p>BYPASS</p> | <p>Output writer is to discontinue processing the current subfile and continue processing on the next subfile. The bypassed subfile can be restarted later.</p> |
| <p>COPIES, nnn</p> | <p>Specifies the number of copies the output writer is to produce of each subfile it processes. The number of copies that may be specified is from 0 to 255. If zero is specified, the subfile is closed when the current processing of the file is completed. This function cannot be used with PD or for files being processed from a redirected disk, diskette, or tape.</p> |
| <p>DELETE</p> | <p>Delete subfile currently being processed from spool file and proceed with the next subfile ready to be processed.</p> |
| <p>DEVICE[.did]</p> | <p>Indicates that the output writer is to change the device it is currently using to print (punch) its output. If a new device (did) is identified in the function code, that device is allocated to the output writer, and the current device is deallocated. If a new device is not identified, a device having the same characteristics as the current device will replace the current device. If the output is to be redirected, specify the tape, disk, or diskette device (did) in the function code, and all subsequent output files are redirected to that device, as long as the same copy of the output writer remains in main storage.</p> |
| <p>DISPLAY</p> | <p>Output writer is to display the status of subfile being processed on the console.</p> |
| <p>HALT</p> | <p>Output writer to terminate after current subfile is processed. If multiple copies of the current subfile are requested, the remaining copies are produced when the output writer is reloaded.</p> |
| <p>HOLD</p> | <p>Output writer is to place the subfile it is processing in the hold state and process the next subfile. Subfiles in the hold state are not available for processing until they are released by a BEGIN command (C.2.5).</p> |
| <p>INPUT, did[.B] [{RET}] [{REL}]</p> | <p>Directs the output writer to accept input from the tape, disk, or diskette unit (did) identified in the function code. This function is used to reintroduce redirected output so that it may be printed or punched. If the input is from disk or format label diskette, the B option may be included to permit specific files to be selected. When B is entered with the function, a message is displayed requesting another function. Enter the BX function with modifiers to specify the particular files to be printed or punched. Redirected output (print or punch) can be retained after processing by including the RET option. After the file is processed (printed or punched), the spool subfile is placed in a hold condition on disk or diskette. The REL option allows you to release these retained subfiles. That is, the REL option causes the output writer to locate redirected output previously retained on disk or diskette and process them. These subfiles are deleted after they are processed. The INPUT function may not be used with PD.</p> |
| <p>NBURST</p> | <p>Output writer is to function in the nonburst mode. If this function code is received while the output writer is processing a subfile, it does not take effect until the subfile is completed. Nonburst mode processing is described in C.2.7.</p> |
| <p>RETAIN</p> | <p>Retains the currently active subfile in a HOLD condition in the spool file after it is processed. The retained file is unavailable for additional processing until released via the BEGIN command (C.2.5). (Otherwise, you delete the retained file via the DELETE command (C.2.3).)</p> |

Table C-3. Output Writer Function Codes (Part 2 of 2)

| Function | Meaning |
|---|---|
| <u>R</u> ESTART [{ nnn PAGE, nnnn } CARD, nnnn] | Restarts processing of the currently active subfile from a number of pages or cards. If no number is specified, the output writer is to restart processing from the beginning of the subfile. If only nnn is entered, the file processing is restarted nnn pages or cards back from the current position of the file. If PA or CA is entered with nnnn, the file is positioned back to the page or card identified by nnnn. This function may not be used with PD. |
| RESTART, { nnn LI, nnn } | Restarts processing at nnn lines from the current position or at the line number indicated by nnn in the log file being processed |
| SD | Used for debugging purposes only. If specified, the message ENTER SPOOL DEBUG COMMAND is displayed. This message requires a reply of ALL, LOG, PRINT, PUNCH, or RDR to print the directories of the various spool file queues. When printing is completed, the message ENTER SPOOL DEBUG COMMAND is displayed again. Enter another debug command or enter HALT to terminate the spool debug. |
| SKIP, { nnnn PAGE, nnnn } CARD, nnnn } | Directs the output writer to skip n number of pages or cards. If only nnnn is entered, the file is positioned nnnn pages or cards from the current position. If PA or CA is entered with nnnn, the file is positioned forward to the page or card identified by nnnn. After positioning, a request is made for an additional function. This function may not be used with PD. |
| SKIP, { nnn LI, nnn } | Directs the output writer to skip nnn lines from the current position or to skip to the line number indicated by nnn in the log file being processed |
| <u>S</u> TOP[, <u>P</u> AGE] | Directs the output writer to stop processing. If STOP is entered without PAGE, the output writer terminates immediately. If STOP is entered with PAGE, the output writer terminates after printing the complete current page. If a subfile is being processed when this function code is received, the subfile is closed but not deleted. When the subfile is accessed by another output writer, it is processed from the point at which it was closed by the previous output writer. |

NOTE:

After you enter a function, another request is usually made to enter another function after the current request is complete. If no additional functions are required, respond with a null entry (i.e., press the transmit key).



Examples:

1. Operator keyin:

```
PR BX,FO=*
```

Function requested:

The output writer is loaded in burst mode to group all spooled output files associated with the form name specified and then print the groups on a first-in, first-out basis. The output writer terminates when all form name file groups have been printed.

2. Operator keyin:

```
PR BX,ID=ABCDEF,JO=MYJOB
```

Function requested:

The currently loaded output writer is to begin printing all remote batch files associated with an ID of ABCDEF and created by job MYJOB. ID permits the output writer to print or punch all files (logs not processed) that have a remote ID equal to the ID entered by the operator. The output writer terminates when all remote batch MYJOB files with the ID of ABCDEF have been printed.

3. Operator keyin:

```
PR(100) BU,JO=MYJOB
```

Function requested:

The output writer is loaded to begin redirecting, in burst mode, all print files created by MYJOB, to the tape with a did of 100. Upon completion, the output writer requests another function if more print files exist in the spool file that were not created by MYJOB.

4. Operator keyin:

```
PR BX,DDPID=ABCD
```

Function requested:

The currently loaded output writer is to begin printing in burst mode all DDP files associated with a host-id of ABCD. All DDP output files created by jobs initiated with host-id ABCD are printed by the output writer, and then it terminates.

5. Operator keyin:

```
PR BX,UID=MYIDEN, JOB=PAYROLL
```

Function requested:

The output writer is loaded to print all nonauxiliary destined print files created by the job named PAYROLL on behalf of the workstation user identified by MYIDEN.

6. Operator keyin:

```
RP BX,UID=MYIDEN, JOB=PAYROLL
```

Function requested:

The output writer is loaded to print at the central site all auxiliary destined print files created by the job named PAYROLL on behalf of the workstation user identified by MYIDEN.

C.4.4. Directing the Operation of an Output Writer (Messages)

Function:

To change the operating mode or processing criteria under which an output writer is to function. The operator enters solicited and unsolicited messages. Unsolicited communication is not allowed if an output writer is waiting for a response to a solicited message.

Formats:

■ Unsolicited Messages

$$\theta\theta \left\{ \begin{array}{l} \text{PR} \\ \text{PU} \\ \text{PD} \end{array} \right\} [(did)] \text{ function-code}$$

where:

 $\theta\theta$

Is the job number and message-id used to communicate with an output writer through the supervisor in the unsolicited mode of transmission.

$$\left\{ \begin{array}{l} \text{PR} \\ \text{PU} \\ \text{PD} \end{array} \right\} [(did)] \text{ function-code}$$

Is the message text that is used to communicate with and identify the desired output writer. As shown, it is similar to the command format used to load an output writer (C.4.1). The difference between them is that the device identification, function code parameters, and modifiers are optional in the command format; but the device identification is optional here, the function code is required, and modifiers are not permitted. The did parameter here is used to identify the output writer that is to receive the instructions being sent in the function code portion of the message. (In this case, you use the DEV function code to redirect print or punch output.) A function code is required because the output writer being communicated with is already loaded. The function codes that can be specified here are the same as those that can be specified in the command format, and are described in Table C-3.

■ Solicited Messages

$$\theta i \text{ function-code}$$

where:

 θi

Is the job number and message-id used to reply directly to an output writer in the solicited mode of transmission. The message-id corresponds to the message-id of the output writer message that requested the reply.

$$\text{function-code}$$

Is the message text that is used to instruct the requesting output writer of the function to be performed. The function codes that may be specified are the same as those described in Table C-3.

Examples:**1. Operator keyin:**

```
00 PR(002) STOP
```

Function requested:

The output writer that is using the printer whose did is 002 is to stop all processing immediately.

2. Console display:

```
05? ENTER OUTPUT WRITER FUNCTION
```

Operator keyin:

```
05 BURST
```

Function requested:

The output writer that requested a new function is to begin processing in the burst mode.

C.5. SPOOL FILE PROCESSING AIDS

A special system service program (SL\$LOG) with a job control stream (DUMPLOG or DUMPLOGT) supplied by Sperry allows you to dump the accumulated job and workstation log from the spool LOG file to a SYSLOG file created on disk or tape. Once in SYSLOG, the accumulated log is available for processing by user programs and by a job log report program (JOBLOG) that produces a job accounting report.

The SL\$LOG program, using the DUMPLOG or DUMPLOGT job control stream, also allows you to dump the accumulated console log from the spool LOG file to the SYSLOG file on tape or disk. The accumulated console log is available for processing by a user program once it is dumped.

The following paragraphs describe how the accumulated job and workstation log is transferred to the SYSLOG file, how the job log accounting reports are produced, and how the accumulated console log is transferred to SYSLOG.

C.5.1. Saving the Accumulated Job and Workstation Log in SYSLOG (SL\$LOG Program)

OS/3 spooling maintains a spool log subfile for each job processed in your system. Each job log subfile contains a copy of:

- the system messages transmitted by the system on behalf of the job (log records);

- the job control statements processed for the job (log records);
- any user snap and cancel dumps processed for the job; and
- any accounting records maintained for the job (accounting records are available only if the system was generated with the accounting option specified (JOBACCT=YES)).

OS/3 spooling also maintains a spool log subfile for each workstation session when the CONSOLOG parameter is included during SYSGEN. The workstation log subfile contains a record of all system messages written to and read from the workstation.

As each job or workstation session terminates, its associated spool log subfile is closed and normally output to a high-speed printer as soon as the device becomes available. If the system is operating with the accumulate system job and workstation log files option in effect (the system was generated with the SYSLOG=YES parameter specified, or the SET SPL,DUMP command was issued by the operator after the system was initialized), the job and workstation log subfiles are marked as having been printed, and then accumulated for future use.

If job log and workstation log subfiles are being accumulated, the SL\$LOG program is used to transfer the accumulated job and workstation log to the SYSLOG file on tape or disk. The SL\$LOG program allows for dumping log (L) and workstation (W and R) records only, accounting (A) records only, or all records in the accumulated log. Snap and cancel dump records are not dumped. After the transfer operation takes place, SL\$LOG effectively erases the closed accumulated log from the spool LOG file, returning the space to the spooling system for its reuse.

If a disk is being used for the output SYSLOG file, the accumulated job and workstation log may be transferred to a previously created SYSLOG. If tape is being used, a new tape must be mounted each time the SL\$LOG program is run in order to transfer the accumulated log. If the same tape is used, the accumulated log file previously written on the tape will be overwritten by the accumulated log currently being transferred to it.

The SL\$LOG program is executed by using the RUN command with a job control stream supplied by Sperry, as described in the paragraphs that follow. A typical job log subfile printout is shown in Figure C-1.

```

// LOAD CASEY,F2
//PRNTR) SPL ,,,,P,NOHDR
// RUN
025 TEST CARD FOR OCL PROG
// LOAD CASEY,F2
//PLNCH SPL ,S,,,MYFORMS
//PLNCH SPL HOLD
// RUN
055 TEST CARD FOR OCL PROG
// FIN
AC01 JOB OCL002 ACCT. NO. ASSIGNED MEMORY=00008192 BYTES (PLUS 003072 BYTE PROLOGUE) 00/00/00
JCO7 USING DEV=FFF TYPE=PRNTR DEV=FFF TYPE=PRNTR
JCO8 USING DEV=FFF TYPE=PLNCH
JCO1 JOB OCL002 EXECUTING JOB STEP CASEY001 #001 00:01:43
AC10 LFD - PRNTR , FORM NAME - STAND1 , COPIES - (001), PAGES - (0000000), STEP #001
AC11 STEP #001 (CASEY001) USED 00004866 BYTES ELAPSED WALL CLOCK TIME=00:00:04.494 TOTAL SVC CALLS=00000213
AC12 TERM CODE=000 SWITCH-PRIORITY=10 CPU TIME USED =00:00:00.610 TRANSIENT CALLS=00000010
AC19 DEVICE EXCP'IS 303=000JL08J PRT=C0C0C02
JCO7 USING DEV=FFF TYPE=PRNTR DEV=FFF TYPE=PRNTR
JCO8 USING DEV=FFF TYPE=PLNCH
JCO1 JOB OCL002 EXECUTING JOB STEP CASEY001 #002 00:01:51
AC10 LFD - PRNTR , FORM NAME - STAND1 , COPIES - (001), PAGES - (0000000), STEP #002
AC11 STEP #002 (CASEY001) USED 00004866 BYTES ELAPSED WALL CLOCK TIME=00:00:04.665 TOTAL SVC CALLS=00000212
AC12 TERM CODE=000 SWITCH-PRIORITY=10 CPU TIME USED =00:00:00.593 TRANSIENT CALLS=00000010
AC19 DEVICE EXCP'IS 303=000JL08J PRT=C0C0C02
JCO7 USING DEV=FFF TYPE=PRNTR DEV=FFF TYPE=PRNTR
JCO8 USING DEV=FFF TYPE=PLNCH
AC21 JOB TOTALS USED 00004866 BYTES TOTAL ELAPSED WALL CLOCK TIME=00:00:20.605 TOTAL JOB SVC CALLS=00000425
AC22 WALL CLOCK TIME OF ALL STEPS =00:00:09.159 JOB TRANSIENT CALLS=00000020
AC23 TOTAL CPU TIME OF ALL STEPS =00:00:01.203 TOTAL JOB EXCP'IS =00000164
JCO2 JOB OCL002 TERMINATED NORMALLY 00:02:01

```

Figure C-1. Typical Job Log Subfile Printout

C.5.1.1. Using Tape for the Job and Workstation Log SYSLOG File (RUN DUMPLOGT)

Function:

To establish a SYSLOG file on tape and transfer the accumulated job and workstation log records from the system spool LOG file into this file. After the transfer operation is completed, the output tape will be rewound to its load point.

Format:

```

RUN DUMPLOGT [ , [ D={ACT} ] [ , V={vsn} ] [ , C={Y} ] ]
              [ , [ D={LOG} ] [ , V={SYSLOG} ] [ , C={N} ] ]

```

Positional Parameter 1:

DUMPLOGT

Indicates that the SYSLOG file is to be established on tape.

Positional Parameter 2:

Omitted from this use of the RUN command; however, a comma must be entered in this position.

Keyword Parameters:

D=**ACT**

Indicates that accounting (A) records only are to be transferred.

D=**LOG**

Indicates that log (L) and workstation (W and R) records only are to be transferred.

If the D keyword parameter is omitted, all job and workstation log records are transferred.

V=**vsn**

Identifies the volume serial number of the tape you want to use.

V=**SYSLOG**

Identifies SYSLOG as the volume serial number to be used.

C=**Y**

Indicates that checkpoint records are desired.

C=**N**

Indicates that no checkpoint records are desired.

C.5.1.2. Using Disk for the Job and Workstation Log SYSLOG File (RUN DUMPLOG)

Function:

To establish a SYSLOG file on disk and transfer the accumulated job and workstation log records from the system pool LOG file into this file.

Format:

```
RN DUMPLOG , F=ALLOC [ , D={  
                                  } ]
```

Positional Parameter 1:

DUMPLOG

Indicates that the SYSLOG file is to be established on disk.

Positional Parameter 2:

Omitted from this use of the RUN command; however, a comma must be entered in this position.

Keyword Parameters:

F=ALLOC

Indicates that SYSLOG file space is to be allocated on disk.

D=~~ACC~~

Indicates that accounting (A) records only are to be transferred.

D=~~LOG~~

Indicates that log (L) and workstation (W and R) records only are to be transferred.

If the D keyword parameter is omitted, all job and workstation log records are transferred.

C.5.1.3. Adding Job and Workstation Log Subfiles to the SYSLOG File on Disk (RUN DUMPLOG)

Function:

To add additional accumulated job and workstation log subfiles to an existing SYSLOG file.

Format:

```
RUN DUMPLOG [ . , D = { ACT }
              { LOG } ]
```

Positional Parameter 1:

DUMPLOG

Indicates that the SYSLOG file is to be established on disk.

Positional Parameter 2:

Omitted from this use of the RUN command; however, a comma must be entered in this position.

Keyword Parameters:

D=ACT

Indicates that additional accounting (A) records only are to be transferred.

D=LOG

Indicates that additional log (L) and workstation (W and R) records only are to be transferred.

If the D keyword parameter is omitted, all additional job and workstation log records are both transferred.

C.5.1.4. Reinitializing the Job and Workstation Log SYSLOG File on Disk (RUN DUMPLOG)

Function:

To reinitialize (effectively erase) the SYSLOG file and transfer new accumulated job and workstation log subfiles from the spool LOG file to the SYSLOG file. Thus, the current contents of the SYSLOG file are overwritten with new spool LOG file job and workstation log records.

Format:

```
RUN DUMPLOG . , F=INIT [ . , D = { ACT }
                       { LOG } ]
```

Positional Parameter 1:

DUMPLOG

Indicates that the SYSLOG file is to be established on disk.

Positional Parameter 2:

Omitted from this use of the RUN command; however, a comma must be entered in this position.

Keyword Parameters:

F=INIT

Indicates that SYSLOG file space is to be reinitialized on disk.

D=ACT

Indicates that accounting (A) records only are to be transferred.

D=LOG

Indicates that log (L) and workstation (W and R) records only are to be transferred.

If the D keyword parameter is omitted, all accounting and workstation log records are transferred.

C.5.2. Producing Job Accounting Reports (JOBLOG Program)

Once your job log SYSLOG file has been created by SL\$LOG, you can execute JOBLOG to produce a job accounting report. JOBLOG uses the OS/3 independent sort/merge routine to sort the SYSLOG file. When the SYSLOG file is on disk, the sort/merge routine writes the sorted SYSLOG file to the same volume serial number you specify for the unsorted SYSLOG input file. When the SYSLOG file is on tape, the routine writes the sorted SYSLOG file to a different tape from the volume serial number you specify for the unsorted SYSLOG input. Thus, you must mount two tapes when you run JOBLOG with tape input – the first is the SYSLOG input file with the volume serial number you specify, and the second is a "work" tape for the sorted SYSLOG output file. You do not specify a volume serial number for the work tape. You can execute JOBLOG by using the RUN command with a job control stream supplied by Sperry.

C.5.2.1. Running JOBLOG with Tape Input (RUN JBLOGT)**Function:**

To sort tape file input to produce a job accounting report.

Format:

$$\left\{ \begin{array}{l} \text{RUN} \\ \text{RV} \end{array} \right\} \text{JBLOGT} \left[\begin{array}{l} \cdot, V = \left\{ \begin{array}{l} \text{vsn} \\ \text{SYSTEM} \end{array} \right\}, S = \left(\begin{array}{l} \text{A} \\ \text{B} \\ \text{C} \end{array} \right) \end{array} \right]$$

Positional Parameter 1:

JBLOGT

Indicates that the job accounting file input to JOBLOG is on tape.

Positional Parameter 2:

Omitted from this use of the RUN command; however, a comma must be entered in this position.

Keyword Parameters:

V=vs n

Identifies the volume serial number of the tape containing the job accounting file.

V=SYSLOG

Identifies SYSLOG as the volume serial number of the tape containing the job accounting file.

S=A

Indicates sort option A. Jobs are sorted in the order submitted from the volume identified by the V parameter.

S=B

Indicates sort option B. Jobs are sorted in accounting number and job name order from the volume identified by the V parameter. Subtotals are taken whenever the accounting number and job name sequence changes.

S=C

Indicates sort option C. Job are sorted in accounting number and job name order from the volume identified by the V parameter. Subtotals are taken whenever the accounting number field changes.

Example:

The job accounting file is named ACCTG and is residing on tape. The file is to be sorted in the order that the jobs were submitted. You use the following command:

RUN JBLOGT, ,V=ACCTG,S=A

C.5.2.2. Running JOBLOG with Disk Input (RUN JBLOG)**Function:**

To sort disk file input to produce a job accounting report.

Format:

$$\left[\begin{array}{l} \text{RUN} \\ \text{RV} \end{array} \right] \text{JBLOG} \left[, , V = \left[\begin{array}{l} \text{vsn} \\ \text{RES} \end{array} \right] , L = \left[\begin{array}{l} \text{file-identifier} \\ \text{SYSLOG} \end{array} \right] , S = \left[\begin{array}{l} \text{A} \\ \text{B} \\ \text{C} \end{array} \right] \right]$$

Positional Parameter 1:**J B L O G**

Indicates that the job accounting file input to JOBLOG is on disk.

Positional Parameter 2:

Omitted from this use of the RUN command; however, a comma must be entered in this position.

Keyword Parameters:**V=v s n**

Identifies the volume serial number of the disk containing the job accounting file.

V=RES

Identifies SYSRES as the volume containing the job accounting file.

L=file-identifier

Identifies the file name of the job accounting file.

L=SYSLOG

Identifies SYSLOG as the name of the job accounting file.

S=A

Indicates sort option A. Jobs are sorted in the order submitted from the volume identified by the V parameter.

S=B

Indicates sort option B. Jobs are sorted in accounting number and job name order from the volume identified by the V parameter. Subtotals are taken whenever the accounting number and job name sequence changes.

S=C

Indicates sort option C. Jobs are sorted in accounting number and job name order from the volume identified by the V parameter. Subtotals are taken whenever the accounting number field changes.

Example:

The job accounting file is named SYSLOG and is residing on the SYSRES volume. You need the report by accounting number and job name. You use the following command:

```
RUN JOBLOG
```

C.5.3. Saving the Accumulated Console Log in SYSLOG (SL\$LOG Program)

OS/3 spooling maintains a record of all messages written to and read from the console when the CONSOLOG parameter is included during SYSGEN. This console log subfile is part of the spool LOG file and can be accessed by the operator and by the SL\$LOG program in a manner similar to that for the job and workstation logs. Console logs are accumulated when either the SYSGEN parameter RETAINLOG has been specified or when the operator enters the SET SPL,CNSLG command with the RETAIN parameter included. You can also use the SET SPL,CNSLG command to turn the console log collection on or off and to print or not print the console logs (C.2.11). When the console logs are accumulated, the SL\$LOG program allows you to dump them to the SYSLOG file created on tape or disk.

Usually you execute SL\$LOG to dump the accumulated job and workstation log so it can be used as input to the JOBLOG program. However, when you also want to dump the accumulated console logs, perform the following steps in the sequence shown to ensure the integrity of the output:

1. Execute the SL\$LOG program to dump the accumulated job and workstation log records.
2. Execute the JOBLOG program to produce a job accounting report.
3. Execute the SL\$LOG program to dump the accumulated console log records.
4. Execute a user-written program to process the console log output.

The SL\$LOG program is executed by using the RUN command with a job control stream supplied by Sperry, as described in the paragraphs that follow.

C.5.3.1. Using Tape for the Console Log SYSLOG File (RUN DUMPLOGT)

Function:

To establish a SYSLOG file on tape and transfer the accumulated console log records from the system spool LOG file into this file. After the transfer operation is completed, the output tape will be rewound to its load point.

Format:

```
RUN DUMPLOGT, .D=CON [ .V={ v s n } ] [ .C={ N } ]
```

Positional Parameter 1:

DUMPLOGT

Indicates that the SYSLOG file is to be established on tape.

Positional Parameter 2:

Omitted from this use of the RUN command; however, a comma must be entered in this position.

Keyword Parameters:**D=CON**

Indicates that the accumulated console log (C) records only are to be transferred.

V=vs n

Identifies the volume serial number of the tape you want to use.

V=SYSLOG

Identifies SYSLOG as the volume serial number to be used.

C=Y

Indicates that checkpoint records are desired.

C=N

Indicates that no checkpoint records are desired.

C.5.3.2. Using Disk for the Console Log SYSLOG File (RUN DUMPLOG)**Function:**

To establish a SYSLOG file on disk and transfer the accumulated console log records from the system spool LOG file into this file.

Format:

```
RUN DUMPLOG, , F=ALLOC, D=CON
```

Positional Parameter 1:**DUMPLOG**

Indicates that the SYSLOG file is to be established on disk.

Positional Parameter 2:

Omitted from this use of the RUN command; however, a comma must be entered in this position.

Keyword Parameters:**F=ALLOC**

Indicates that SYSLOG file space is to be allocated on disk.

D=CON

Indicates that the accumulated console log (C) records only are to be transferred.

C.5.3.3. Adding Console Log Subfiles to the SYSLOG File on Disk (RUN DUMPLOG)

Function:

To add additional accumulated console log subfiles to an existing SYSLOG file.

Format:

```
RUN DUMPLOG, , D=CON
```

Positional Parameter 1:

```
DUMPLOG
```

Indicates that the SYSLOG file is to be established on disk.

Positional Parameter 2:

Omitted from this use of the RUN command; however, a comma must be entered in this position.

Keyword Parameter:

```
D=CON
```

Indicates that additional console log (C) records only are to be added.

C.5.3.4. Reinitializing the Console Log SYSLOG File on Disk (RUN DUMPLOG)

Function:

To reinitialize (effectively erase) the SYSLOG file and transfer a new accumulated console log subfile from the spool LOG file to the SYSLOG file. Thus, the current contents of the SYSLOG file are overwritten with new spool LOG file console log records.

Format:

```
RUN DUMPLOG, , F=INIT, D=CON
```

Positional Parameter 1:

```
DUMPLOG
```

Indicates that the SYSLOG file is to be established on disk.

Positional Parameter 2:

Omitted from this use of the RUN command; however, a comma must be entered in this position.

Keyword Parameters:**F=INIT**

Indicates that SYSLOG file space is to be reinitialized on disk.

D=CON

Indicates that the accumulated console log (C) records only are to be transferred.

Appendix D. Spooling Summary

D.1. GENERATING THE SPOOLING SYSTEM

Spooling is installed in the operating system at system generation time. During supervisor generation (SUPGEN), spooling is established at one of four levels:

1. Output spooling to local devices
2. Input and output spooling to local devices
3. Input and output spooling local and remote batch devices (remote batch processing is included), with the option to suppress input spooling
4. Input and output spooling to local and remote batch devices and to distributed data processing, with the option to suppress input spooling

In the case of level 1, output is spooled to printers, card punches, and data set label diskettes; or output is redirected to magnetic tapes, disks, and format label diskettes for later printing and punching. In the case of level 2, input is spooled from card readers and diskettes, and output is spooled in the same manner as for level 1. In level 3, the remote batch devices (card readers, printers, and card punches) are added to the level 2 devices. And in level 4, distributed data processing disk devices are added to the level 3 devices.

Once this spooling level is entered, additional SUPGEN parameters establish the spooling system, excluding real and virtual devices, as the operator initializes it during the initial program load (IPL) procedure.

During input/output generation (I/OGEN), spooling options are also entered. I/OGEN parameters allocate the number of virtual input and output devices, by device type, to be used by the supervisor. A similar type of real device must be identified for each type of virtual device specified. Jobs are scheduled based on the number of virtual devices rather than the number of real devices.

I/OGEN parameters also allocate the number of locally or remotely connected workstations and associated auxiliary printers.

All user jobs run automatically according to the spooling parameter options selected during SYSGEN. No program or job control changes are required. (If you have specified spooling to workstation auxiliary printers, you must include an // OPTION OUT or // ROUTE job control statement in your job control stream to direct the print output files

to your own or another user's auxiliary printer.) With a spooling system, a user may override spooling for a particular device assignment by specifying a physical device address or REAL on the DVC job control statement. All references within the job to this DVC will access the device directly.

The spooling system generates job logs that encompass:

- system messages directed on behalf of the job;
- job control used for the job;
- accounting information (optional); and
- user dumps.

Spooling also maintains a record of all console activity (the console log) and all workstation activity during each workstation session (workstation logs). All system messages and commands displayed on the console or workstation screen are recorded in the console log or workstation log, respectively.

The physical volume that contains the spool file (SYSPOOL) is generally allocated during the IPL sequence. This volume may be specified:

- a. at IPL time;
- b. at SYSGEN; or
- c. by default to the system resident device.

Here a overrides b, and b overrides c. SUPGEN spooling parameters may be entered to establish multivolume spooling with up to eight disk volumes. The spool file size for each volume is specified in cylinders; the default value is 50.

The IPL procedure allows the operator to use the SYSGEN-established spool file volumes or to change the SYSGEN configuration by mounting fewer volumes or volumes other than those specified.

The spool file is limited by the spooling system to three physical extents per volume. The file may be allocated (or scratched) by the user. (The file-id is SYSPOOL.) This should be done only under a nonspooling supervisor.

During IPL, if the supervisor finds that the designated volume already contains a spool file, it utilizes this file and does not do any additional allocation.

D.1.1. SUPGEN Spooling Parameters

SUPGEN spooling parameters are used to establish the spooling level and configure the supervisor for spooling. Each spooling parameter option selected is initialized during the IPL procedure.

SPOOLING={ INPUT
NO
OUTPUT
REMOTE
DDP }

Establishes spooling and specifies spooling level. If no spooling capability is specified, all other spooling parameters are ignored. Also, no virtual devices in I/OGEN.

INPUT

Both input readers and output writers used for spooling.

NO

No spooling capability.

OUTPUT

Only output writers used for spooling.

REMOTE

Remote batch processors, input readers, and output writers used for spooling. If SPOOLING=REMOTE, also specify SPOOLICAM.

DDP

Distributed data processing, remote batch processors, input readers, and output writers used for spooling. If SPOOLING=DDP, also specify SPOOLICAM.

SPOOLNOINPUT={ NO
YES }

Indicates whether or not input spooling for remote batch processing is suppressed. Cannot load the input reader. Used when SPOOLING=REMOTE or DDP is specified and input spooling is not required.

YES

Specifies input spooling is to be suppressed.

If omitted, input spooling is not suppressed.

SPOOLVSN={ SYSRES
vsn }

Volume serial number of primary disk that supervisor uses as spool file.

SPOOLVSNn={ SYSRES
vsn }

Multivolume spooling supported. May allocate up to eight disk volumes for spooling. Identifies the nth sequential volume of spool file, where n is decimal number from 2 to 8. May identify each spool volume through multiple calls of keyword (e.g., SPOOLVSN2=vsn, SPOOLVSN3=vsn, etc.) or may denote only last sequential volume. If identifying only last volume, all other volumes default to *, permitting you to specify device addresses of disk units containing spooling volumes at IPL. All volumes in a multivolume spool file must be the same type of disk. For optimum performance, allocate one cylinder of the file for the spooling directory on the first volume, and contiguous data file space across the remaining volumes.

SPOOLCYL={ ALL
1-1000
50 }

Number of cylinders initially allocated for the spool file on primary spooling volume. Default 50 recommended. If request exceeds default value of virtual devices in I/OGEN, may require more than default number of spool cylinders. ALL allocates all available cylinders on primary spooling volume. Option should be used only for dedicated spooling volume.

$$\left[\text{SPOOLCYL}_n = \begin{matrix} \text{ALL} \\ 1-1000 \\ 50 \end{matrix} \right]$$

For multivolume spooling, specifies number of cylinders initially allocated for spool file on nth sequential spooling volume, where n is decimal number from 2 to 8.

$$\left[\text{SPOOLBUFR} = \begin{matrix} 1 \\ 2 \\ 4 \\ 8 \\ 16 \\ 32 \end{matrix} \right]$$

Specifies the size of the spooler word buffer allocated to each job preamble in terms of the number of 256-byte blocks of main storage.

$$\left[\text{SPOOLLOWBUFR} = \begin{matrix} 2 \\ 4 \\ 8 \\ 16 \\ 32 \end{matrix} \right]$$

Specifies the number of 256-byte blocks of main storage to be used by the print and punch output writers for spool buffers. Specifying larger values for this parameter increases system performance by reducing number of disk accesses to spool file.

$$\left[\text{SPOOLMAP} = \begin{matrix} 1-32767 \\ 64 \end{matrix} \right]$$

Specifies the number of full words (four bytes) of main storage reserved for the resident spool file bit map. Spool file suballocation is controlled by the bit map. If multivolume spooling is being used, the spool file bit map size should be calculated on the total number of cylinders being reserved for the spool file.

$$\left[\text{SPOOLICAM} = \begin{matrix} C1 \\ Cn \\ Mn \\ C? \end{matrix} \right]$$

Specifies the name of ICAM symbiont load module (C1-C9, M1-M9, or C?) called by the spooler to service remote batch and distributed data processing spooling. If C? is specified, operator is asked at load time for a valid ICAM name to be loaded. Must specify parameter SPOOLING=REMOTE or SPOOLING=DDP and must configure the ICAM symbiont as described in the COMMCT section of system generation. If omitted, jobs creating remote batch or distributed data processing output encounter error at end of job and require ICAM to be manually loaded to process output.

$$\left[\text{SPOOLCOMP} = \begin{matrix} \text{NO} \\ \text{YES} \end{matrix} \right]$$

Specifies whether spool file output is compressed where three or more consecutive blanks are present. Do not specify SPOOLCOMP=NO when most files created contain large amount of blanks or use block sizes greater than 120.

NO

Specifies output image in spool file is not to be compressed.

If omitted, spool file output image is compressed.

$$\left[\text{SPOOLTEST} = \begin{matrix} \text{NO} \\ \text{YES} \end{matrix} \right]$$

Specifies whether console message requesting operator decision to print test line is displayed when change of form is required.

NO

Specifies no test lines message is displayed.

If omitted, test lines message is displayed.

[SPOOLUPDATE={NO
YES}]

Specifies whether spool directory entry is updated when logical track is crossed or when file is closed.

NO

Specifies that spooler updates the spool directory entry only when a program file closes. Should a program cancel, you lose any output generated by the program before cancellation.

If omitted, spool directory entry is updated each time logical track is crossed in a program file. Should a program cancel, you are able to print any output that the program generated before cancellation.

[SPOOLBURST={NO
YES}]

Indicates whether or not output spooling functions in burst mode to enable output files to be written before termination of associated job. Operating in burst mode requires output processing criterion configured to control output writer's mode of processing for available output files.

YES

Specifies output spooling functions in burst mode of operation.

If omitted, output file processing by output writers is not done until job terminates (nonburst mode).

[SPOOLMODE=criterion[,criterion value]]

Establishes output file processing criterion for spooler when operating in burst mode. Output files processed according to criterion specified; processing stops when criterion satisfied. If operating in burst mode and no criterion specified, SPOOLMODE=PRI and output files are processed under job priority basis.

The criterion and associated criterion value that may be specified are:

ACCTNO,account-number

One to four alphanumeric characters indicating account number is criterion used for determining file processing. All files created by jobs whose account number same as criterion configured (as specified on the JOB control statement) are processed without operator intervention.

CARTNAME,cartridge-name

One to eight alphanumeric characters indicating cartridge name is criterion for determining file processing. All files whose cartridge name same as criterion configured (as specified on LCB job control statement) are processed without operator intervention.

DEVICE,device-type-code

Device type code as 770, 776, or 789. Indicates that device type number is criterion for determining file processing. All files available for processing whose device type is same as criterion configured are processed without operator intervention.

(continued)

[SPOOLMODE=criterion[, criterion value]]

(cont)

FILE,filename

One to eight alphanumeric characters indicating file name is criterion used for determining file processing. All files available for processing whose file name is same as criterion configured are processed without operator intervention.

FORM,formname

One to eight alphanumeric characters indicating form-name criterion for determining file processing. All files whose form name same as criterion configured (on VFB or SPL job control statements) are processed without operator intervention.

JOB,jobname

One to eight alphanumeric characters indicating job name is criterion for determining file processing. All files whose job name is same as criterion configured (as specified on JOB control statement) are processed without operator intervention.

PR

Indicates that job priority is criterion for determining file processing. Files are processed on first-in, first-out basis without operator intervention.

[SPOOLHDR={ NO
YES }]

Specifies whether spooled output files are printed with headers.

NO

Specifies spooled output files printed without headers.

If omitted, a 3-page header precedes printing of each spooled file.

[SPOOLPRT={ ACT
ALL
LOG
NO }]

Specifies log printing criteria for terminated job; includes log types for printing (L and A) and whether or not to print log records. Must specify JOBACCT=YES to have A records.

ACT

Specifies accounting records printed when job terminates.

ALL

Specifies both job log and accounting records printed when job terminates.

LOG

Specifies only log records printed when job terminates.

NO

Specifies job log and accounting records not printed.

If omitted, both log (L) and accounting (A) records are printed when job terminates.

[JOBACCT={**NO**
YES}]

Indicates whether or not the resident job accounting routines maintain a record of CPU time used by job and job step facilities, number of I/O requests made per device, number of supervisor requests generated, main storage usage, and transient function usage. May specify only if spooling configured.

YES

Specifies log accounting (A) records maintained.

If omitted, accounting records are not maintained.

[SYSLOG={**NO**
YES}]

Indicates whether or not job and workstation log files (L, A, W, and R records) are accumulated for future transfer to user disk or tape file. If not configured at SYSGEN, can be requested by operator from console. Must specify JOBACCT=YES to have A records.

YES

Specifies log files are accumulated.

If omitted, log files are not accumulated.

[CONSOLOG={MAX
MIN
NO
NORM}]

Indicates whether or not console log (C) records and workstation log (W and R) records are collected and specifies size of main storage buffer area. Record of all communication between system and console (console log) and system and workstation (workstation log) collected in buffer area in main storage, then copied onto spool file when buffer filled.

MAX

Specifies a 1072-byte buffer.

MIN

Specifies a 304-byte buffer.

NO

No console log function requested.

NORM

Specifies a 560-byte buffer.

If omitted, console and workstation log records are not collected in spool file.

[CONPRINT={**NO**
YES}]

Specifies whether the console log is printed when file is breakpointed.

NO

Specifies the console log (C) records are not to be printed.

If omitted, console log is printed.

[RETAINLOG={**NO**
YES}]

Indicates whether or not console log is retained in spool file after printing and accumulated for future use.

YES

Specifies console log (C) records are accumulated for future use.

If omitted, console log is deleted from spool file after printing.

[SPOOLRECV = { ALL
CLOSED
LOG
NONE }]

Specifies level of recovery for spool file when supervisor is initialized during IPL procedure.

ALL
Specifies all files (complete and incomplete) are recovered.

CLOSED
Specifies only completed files are recovered.

LOG
Specifies only completed log files are recovered.

~~NONE~~
Specifies spool file not to be recovered.

If omitted, spool file is presumed empty and not recovered.

[SPOOLFARSI = { ~~NO~~
YES }]

Indicates whether or not FARSI (Iranian language) translation is executed for all spooled files containing *FARSI or @FARSI as first six characters of // LBL job control statement.

YES
Specifies FARSI translation is to be executed. Output writer must be loaded with additional 400₁₆ bytes of main storage and input reader must be loaded with additional 300₁₆ bytes of main storage.

If omitted, FARSI translation is not executed.

[SPOOLMAXLINE = { 1-255 }]

Specifies maximum number of I/O records processed (in thousands) before system halts job and sends warning message to operator.

[DDPSPOOL = { 1-128 }]

Specifies number of concurrently running tasks that can return spoolout of DDP jobs. Each task returning DDP spoolout requires a task control block and at least 8K of dynamic buffer space.

D.1.2. I/OGEN Spooling Parameters for Virtual Devices

If the supervisor is configured for spooling, I/OGEN parameters allocate the number of virtual printers, readers, and punches to be used by the supervisor, provided real devices are identified in separate sets of I/OGEN parameters. If virtual devices are configured, the supervisor schedules jobs based on the number of virtual devices rather than the number of real devices. A maximum of 99 virtual devices of each type may be specified. If spooling was not configured during SUPGEN or if real devices are not identified, virtual printers, readers, and punches cannot be created.

PRINTER VIRTUAL= $\left. \begin{array}{l} \text{number} \\ \text{of job} \\ \text{slots} \\ \theta \\ 1-99 \end{array} \right\}$

Specifies number of virtual printers if spooling configured. To calculate optimum number, multiply number of jobs that can run concurrently by average number of printer files used for each job. Each virtual printer uses 32 bytes of main storage.

If omitted, the number of virtual printers created is twice the number of job slots specified during SUPGEN.

READER VIRTUAL= $\left. \begin{array}{l} \text{number} \\ \text{of job} \\ \text{slots} \\ \theta \\ 1-99 \end{array} \right\}$

Specifies number of virtual readers, if spooling configured. To calculate optimum number, multiply number of jobs that can run concurrently by average number of reader files used for each job. Each virtual printer uses 32 bytes of main storage.

If omitted, the number of virtual readers created is equal to the number of job slots specified during SUPGEN.



[PUNCH VIRTUAL = { $\left. \begin{array}{l} \text{1 x number} \\ \text{of job} \\ \text{slots} \\ \hline \text{0} \\ \hline \text{1-99} \end{array} \right\} \right]$

Specifies number of virtual punches, if spooling configured. To calculate optimum number, multiply number of jobs that can run concurrently by average number of punch files used for each job. Each virtual punch uses 32 bytes.

If omitted, the number of virtual punches created is equal to the number of job slots specified during SUPGEN.

D.1.3. I/OGEN Spooling Parameters for Workstation Auxiliary Printers

If the supervisor is configured for spooling and you intend to direct print output files to workstation auxiliary printers that are locally or remotely connected to your system, you must include certain I/OGEN parameters when the system is generated. If you want to direct output files to a locally connected auxiliary printer, you must include the WORKSTATION and AUXPRINTER parameters. If you want to direct output files to a remotely connected auxiliary printer, you must include the REMWORKSTATION and REMPRINTER parameters.

WORKSTATION

[CHAN = { C
E }]

Must specify workstations whether system supports interactivity from local workstations or remote terminals.

However, do not specify parameters for the console/workstation; your system generates that device automatically.

[ADDR = { address
 (for 3560)
 (for 3561) }]

Hexidecimal number identifying physical I/O channel with access to workstation. Default specification is device-dependent (determined as function of TYPE).

2-digit hexadecimal number identifying address and number of workstations defined. If more than one address, must be separated by commas, or hyphens may indicate consecutive addresses. You can specify from one to eight workstations for each controller. Valid addresses must end with digits 1-8.

NOTE:

If you want to change console/workstation characteristics, you must specify the console workstation at ADDR=11.

[TYPE = {
 3561
 3562 }]

Type of workstation or group of workstations defined. Type 3562 not applicable to System 80 model 8.

[DVCNO = lun]

Changes logical unit numbers (lun) for specific workstations.

[KATAKANA = {
 YES }]

Workstation recognizes input in Katakana (Japanese language).

[DESPACE = {
 YES }]

Specifies whether the space key protects or destroys the character the cursor is positioned on. This option is available for UTS 20D and UTS 40D workstations.

YES Character is replaced with a blank when space key is pressed.

NO Character is not changed when space key is pressed.

(continued)

WORKSTATION
(cont)

[SCRENMEM={ }-2]

Specifies single or dual screen support for UTS 20, UTS 40, and UTS 40D workstations.

- 1 Indicates single screen support.
- 2 Indicates dual screen support. This allows full screen system messages on screen 1 when the workstation is not allocated to a job. Screen 2 is used for auxiliary output only. Can be specified for UTS 20 workstation only if it has the screen bypass option.

[AUXPRINTER]

You can specify from one to four auxiliary printers for each local workstation you define. You must repeat the AUXPRINTER parameters for each auxiliary printer you want configured. If you define a range of local workstations, the auxiliary printers you specify will be defined for each workstation in that range.

[TYPE={ }
0797
0791
0789
0774
0425]

Identifies the type of auxiliary printer or group of auxiliary printers defined.

[VFB={ }
STAND
OWNVF1]

Printer vertical format buffer as default buffer for printer subsystem being defined. OWNVF1 references user-generated vertical format buffers.

[LOOP={ }-4]

Specifies whether a 3-channel or 4-channel forms control loop is used to control vertical form spacing.

[AUX={ 1-8 }]

Specifies single digit device identification number (channel number) of auxiliary printer defined. Each printer is identified by a 2-digit device identification number; for example, 1 and 2, 3 and 4, and so on. The actual channel number of the printer is usually the odd number of the pair. Therefore, you would specify device identification numbers of 1, 3, 5, and 7 to identify the printer.

[REMWORK
STATION]

AMOUNT={ }-99]

Number of communications terminals that support interactivity. Value is maximum number of interactive sessions system recognizes from remote terminals at one time. If communications network supports interactivity, must specify how many terminals function as remote workstations. No greater than number of terminals in system.

[DESPACE={ }
YES]

Specifies whether the space key protects or destroys the character the cursor is positioned on. This option is available for UTS 20 and UTS 40 workstations.

- YES Character is replaced with a blank when space key is pressed.
- NO Character is not changed when space key is pressed.

(continued)

REWORK
STATION (cont)

[SCRENMEM= { 1-2 }]

Specifies single or dual screen support for UTS 20 and UTS 40 workstations.

- 1 Indicates single screen support.
- 2 Indicates dual screen support. This allows full screen system messages on screen 1 when the workstation is not allocated to a job. Screen 2 is used for auxiliary output only. Can be specified for UTS 20 workstations only if it has the screen bypass option.

[KATAKANA= { YES }]

Remote workstation recognizes input in Katakana (Japanese language).

[REMPRINTER AMOUNT= { 1-99 }]

Number of remotely connected workstation auxiliary printers.

D.2. CONTROLLING THE SPOOLING SYSTEM

After the spooling system is established with system generation parameters, several levels of control are available. Spooling is controlled in a batch environment with job control statements and a macroinstruction, as well as with the capability of sending the operator a message from a remote location. With restrictions, some spooling functions may be controlled with workstation commands. And the system operator can control all spooling functions throughout the system with operator commands.

D.2.1. Job Control Spooling Statements

Used to gain some control over the spooling environment. The DATA, DST, DVC, JOB, OPTION, SPOOL jproc, and ROUTE statements provide spooling control options, as described in the following paragraphs. Considerations for selecting spooled file buffers are also provided.

// DATA FILEID=file-identifier[,RETAIN]
[.IGNORE][DELETE]

Provides means of spooling card or tape input. Precedes and identifies each input data file. Permits multiple input files separated by // DATA statement in a card deck or on tape. Includes option to retain input file after processing.

FILEID=file-identifier

Specifies file identifier. Agrees with either the LBL statement file identifier or concatenation of JOB statement job name and LFD statement file name.

RETAIN

Retains input file in spool file after processing. Remains in spool file available for additional processing until operator command deletes it.

IGNORE

Specifies that // RUN statements in spooled file are processed as data. Normally used when job control stream is converted.

DELETE

Indicates that spool subfiles with the same label name are to be deleted before creating the new subfile.

//[symbol] DST dest-1[,dest-2,...,dest-n]

Identifies remote batch destination for spooled output. Used in place of VOL statement.

dest

Specifies destination used by remote batch processor. One to six alphanumeric characters. Identifies multiple destinations for output.

//[symbol] DVC { nnn [(n)] } { addr
RES OPT
RUN IGNORE
ALT
I
O
REQ[(n)]
REAL

Requests assignment of peripheral device to job. For spooling, specifies output file written to diskette in data set label mode, or specifies data set label diskette input file. Only the following parameters are supported for spooling with diskettes.

nnn

Specifies logical unit number (130—133) for diskette.

{ 1
0 }

Specifies diskette file is input for job (1) or output created by job (0). This parameter is required for diskette spooling.

[,HOST=host-id]

//[symbol] JOB jobname [{ P }] [,min] [,max]

{ P }
H
█
L

Identifies job and indicates beginning of control information for the job. For spooling, specifies space in main storage for job log records, job header printing, and log printing criteria for terminated job.

[, { tasks }] [, { max-time }]

jobname

Specifies 1- to 8-alphanumeric-character name of job used for reference after job is filed.

[,print-option-list]

[,acct-no] [,nXm] [{ ACT }]

{ ACT }
LOG
NOACT
NOLOG
NONE
BOTH

{ P }
H
█
L

Specifies preemptive (P), high (H), normal (N), or low (L) scheduling priority for job.

[, { NOHDR }]

If omitted, normal priority is assigned.

min

Specifies minimum hexadecimal number of main storage bytes required to execute largest job step of job.

If omitted, value specified for max parameter is used.

max

Specifies maximum hexadecimal number of main storage bytes requested, but not required, to execute largest job step of job.

If omitted, value specified for min parameter is used.

If both min and max are omitted, length of largest load module needed by job is used.

// [symbol] JOB
(cont)

{ tasks }

Specifies maximum number of tasks that can be active simultaneously in any job step.

If omitted, one task may be active.

{ max-time }
{ SUP }

Specifies maximum number of minutes job should require to execute, or suppresses the maximum minutes limit entered at SYSGEN.

Max-time specifies execution limit, in minutes, for job. Displays message to operator with option to extend minutes or terminate job if limit is reached.

If omitted, SYSGEN default value is used. (Timer service must be included at SYSGEN to use max-time parameter.)

SUP suppresses max-time function completely for job.

print-option-list

Specifies which options are listed in the job log. With spooling, the default is B, indicating basic job control statements.

acct-no

Specifies 1- to 4-alphanumeric-character account number for job.

nXm

Specifies job log buffer space in main storage. The job log and any output files without reserved buffers use this space. The n is number of buffers, X is a constant, and m is buffer size in 256-byte blocks.

If omitted, one 256-byte buffer is assumed for job log only; if output files are to share space, two buffers are assigned.

{ ACT }
{ LOG }
{ NOACT }
{ NOLOG }
{ NONE }
{ BOTH }

Specifies how the log is printed. Option to print only accounting records (ACT) or only log records (LOG) regardless of what SET SPL operator command is in effect; suppress printing of accounting records (NOACT), log records (NOLOG), or both accounting and log records (NONE); or print both accounting and log records (BOTH) provided the SET SPL operator command does not override.

JOBACCT=YES must be specified at SYSGEN to control A record printing with this parameter.

(continued)

//[symbol] JOB
(cont)

In most systems, both log and accounting records are printed if this parameter is omitted.

The job log records are always printed if job control found to be in error or job terminates abnormally; this parameter has no effect in this case.

{ NOHDR }
{ HDR }

Specifies whether to suppress printing or to print header for job. System must be generated with SPOOLHDR=YES (the SYSGEN default condition).

If omitted, header is printed before job log and output file (nonburst mode).

//[symbol] OPTION p1[, ...,pn]

Specifies job log is printed with job output when specific printer is required; specifies job log destination (where printed); specifies identifier for a job log in accumulated LOG file; and specifies the destination of all job output (job log and any print or punch files specified in the job control stream after the OPTION statement). The parameter (p) is used to specify the identifiers that follow:

LOG=nn

Used in nonburst mode when specific printer is required for job output. The device assigned in // DVC for output is specified for the log (nn). Prevents separation of log and output on different printers and ensures the output is printed.

LOG=CENTRAL

Specifies job log will go to central printer at local site.

LOG=ORIGINATOR

Specifies job log will go to originator's printer.

MERGE=NO

Used when job logs are being accumulated for future use. Creates separate identifier for job log in spool LOG file. Permits user identification of specific job log before contents of LOG file are processed using SL\$LOG.

OUT=CENTRAL

Specifies that all the job output will go to the central printer or punch at the local site.

OUT=ORIGINATOR

Specifies that all the job print output will go to the originator's printer and all punch output will go to the central punch at the originator's site.

OUT=host-id:CENTRAL

Specifies that all the job output will go to the central printer or punch at the DDP site identified by the 1- to 4-character host-id.

(continued)

//[symbol] OPTION

(cont)

OUT=host-id:user-id

Specifies that all the job output will go to the DDP site identified by the 1- to 4-character host-id. The print output will go to a workstation auxiliary printer identified by the 1- to 6-character user-id and the punch output will go to the central punch at the DDP site.

OUT=user-id

Specifies that all the job print output will go to a workstation auxiliary printer (identified by the 1- to 6-character user-id) that is locally or remotely connected to your system.

This statement is supported for compatibility with programs that were written before the 8.0 software release. If you are writing a new program, use the // SPOOL jproc because it allows you to specify, in any order, the keyword parameters that control the spooling environment for output files.

```
//[symbol] SPL { HOLD
                RETAIN
                TAPE
                DISK
                DISKETTE } [,nXm] { ,no-cop }
                [ ,no-skpcode ] [ ,max-rec ]
                [ ,forms ] [ ,NOHDR ] [ ,NOTSTL ]
                [ ,brk-pge ] [ ,NOUPD ] [ ,NOCMP ]
                [ ,RETAIN ] [ ,HOLD ]
                [ ,SECURE ]
```

Controls spooling environment for output file. Specifies space in main storage for file. Option to compress output records; display test lines message; update directories in system spool control table; print file header; retain file after processing; breakpoint printer or punch file; hold queued file from processing; specify maximum number of output records; produce multiple copies of file; and redirect file to tape, disk, or format label diskette.

```
{ HOLD
  RETAIN
  TAPE
  DISK
  DISKETTE }
```

Provides processing options for print and punch output files. Option to hold queued file for later processing (HOLD); retain file after processing (RETAIN); or redirect file to tape (TAPE), disk (DISK), or format label diskette (DISKETTE) for later printing or punching.

nXm

Specifies output file buffer space in main storage. The n is number of buffers, X is a constant, and m is buffer size in 256-byte blocks.

If omitted, file shares job log buffer.

```
{ no-cop }
```

Specifies number of file copies (0 through 255) to be produced. Zero indicates no output.

If omitted, one copy is produced.

(continued)

// [symbol] SPL
(cont)

{no-skpcode}
{7}

Specifies number of lines in print file vertical format buffer that contain skip codes for forms control (including home paper and overflow). Zero indicates no forms control; default is seven skip codes.

{max-rec}
{5120}

Specifies maximum number of output records expected to be processed. If number is exceeded, system halts job and sends message to operator. For large job, specify up to 262,144 records to save processing time. Number is rounded to next higher multiple of 1024 records. Job halt and operator message occurs when max-rec number is reached; default is 5120 records.

forms

Specifies paper form name (for printers) or card type (for punches) of one to eight alphanumeric characters, to be issued in forms change message to operator. If VFB and SPL statements both specify a form name, the VFB name is used.

If omitted, system default is used.

{NOHDR}
{HDR}

Specifies output file header printing. System must be generated with SPOOLHDR=YES (the SYSGEN default condition). NOHDR suppresses printing of file header in burst mode. (Does not suppress printing job header in nonburst mode.)

If omitted, file header is printed under burst mode operation.

NOTSTL

Suppresses test lines message to operator when forms parameter detected.

If omitted, message is displayed asking if sample test page is to be printed on the form specified.

brk-pge

Breakpoints print or punch file after specific number of pages or cards is produced. Largest number that can be specified is 32,000.

If omitted, file is printed or punched according to nonburst or burst operating mode in effect.

NOUPD

Specifies spool control table is updated when file is closed.

If omitted, spool control table is updated after each logical track is spooled.

//[symbol]SPL
(cont)

NOCMP

Specifies output records are not compressed (blanks removed) before written to spool file. Do not use NOCMP when file contains large number of embedded blanks or block size larger than 120.

If omitted, blanks are removed before record is written to spool file and replaced before processed to output device.

RETAIN

Performs the same function as when specified as the first parameter. If you specify RETAIN as the twelfth parameter with TAPE, DUMP, DISK, or DISKETTE, the output is redirected to the appropriate device and a copy of the file is also retained (in the spool file) for later use.

HOLD

Performs the same function as when specified as the first parameter. If you specify HOLD as the last parameter with RETAIN, TAPE, DUMP, DISK, or DISKETTE; or with RETAIN (as the twelfth parameter) and TAPE, DUMP, DISK, or DISKETTE, the output file is first put on hold. The other parameters are acted upon accordingly when the file is released.

SECURE

Specifies that the print subfile destined for a workstation auxiliary printer is secured. The initiator must be logged on when the output is available for printing. Otherwise, the subfile will not be printed.

```
//[symbol] SPOOL [ REDIRECT={ DISK
                    { DISKETTE
                    { TAPE
                    ]
[ , BUF=nXm ]
[ , COPIES={ n
            { 1
            ]
[ , SKIPCODE={ n
              { 7
              ]
[ , RECORDS={ n
             { 5120
             ]
[ , FORMNAME=f o r m s ]
[ , HDR={ YES
        { NO
        ]
[ , TESTPAGE={ YES
              { NO
              ]
[ , PAGEBRK=n ]
[ , UPDATE={ YES
            { NO
            ]
[ , COMPRESS={ YES
              { NO
              ]
[ , RETAIN={ YES
            { NO
            ]
[ , HOLD={ YES
          { NO
          ]
[ , SECURE={ YES
            { NO
            ]
```

Controls spooling environment for output file. Specifies space in main storage for file. Option to compress output records; display test lines message; update directories in system spool control table; print file header; retain file after processing; breakpoint printer or punch file; hold queued files from processing; specify maximum number of output records; produce multiple copies of file; and redirect file to tape, disk, or format label diskette.

```
REDIRECT={ DISK
          { DISKETTE
          { TAPE
```

Redirects print or punch output file to disk (DISK), format label diskette (DISKETTE), or tape (TAPE) for later printing or punching.

BUF=nXm

Specifies output buffer space in main storage. The letter n is the number of buffers, X is a constant, and m is the buffer size in 256-byte blocks.

If omitted, file shares job log buffer.

```
COPIES={ n
        { 1
```

Specifies number of file copies (0 through 255) to be produced. Zero indicates no output.

If omitted, one copy is produced.

```
SKIPCODE={ n
           { 7
```

Specifies number of lines in print file vertical format buffer that contain skip codes for forms control (including home paper and overflow). Zero indicates no forms control.

If omitted, the default is seven skip codes.

```
RECORDS={ n
          { 5120
```

Specifies the maximum number of output records to be processed. If number is exceeded, system halts job and sends message to operator. For large job, specify up to 262,144 records to save processing time. Number is rounded up to next higher multiple of 1024 records. Job halt and operator message occurs when the n number is reached.

If omitted, the default is 5120 records.

FORMNAME=forms

Specifies paper form name (for printers) or card type (for punches) of one to eight alphanumeric characters, to be issued in forms change message to operator. If VFB and SPOOL jproc both specify a form name, the VFB name is used.

If omitted, the system default is used.

(continued)

// [symbol] SPOOL
(cont)

HDR={ YES }
{ NO }

Specifies output file header printing. System must be generated with SPOOLHDR=YES (the SYSGEN default condition). NO suppresses file header in burst mode. (Does not suppress printing job header in nonburst mode.)

If omitted, file header is printed under burst mode operation.

TESTPAGE={ YES }
{ NO }

Specifies if test lines message is to be sent to operator when a change of forms is detected. NO suppresses test lines message.

If omitted, system default is used.

PAGEBRK=n

Breakpoints print or punch file after specific number of pages or cards is produced. Largest number that can be specified is 32,000.

If omitted, file is printed according to nonburst or burst mode in effect.

UPDATE={ YES }
{ NO }

Specifies when spool control table is updated. NO specifies spool table is updated when file is closed.

If omitted, spool control table is updated after each logical track is spooled.

COMPRESS={ YES }
{ NO }

Specifies if output records are compressed (blanks removed) before written to the spool file. NO specifies that the records are not compressed. Do not specify NO when file contains large number of embedded blanks or block size larger than 120.

If omitted, blanks are removed before record is written to spool file and replaced before processed to output device.

RETAIN={ YES }
{ NO }

Specifies if a copy of the file is to be retained in the spool after processing.

If RETAIN=YES is specified in conjunction with REDIRECT, the output file is redirected to the appropriate device and a copy is retained in the spool file after processing.

If omitted, a copy of the file is not retained after processing.

(continued)

```
// [symbol] SPOOL
(cont)
```

```
HOLD={YES}
      {NO }
```

Specifies if a queued file is to be held for later processing. If HOLD=YES is specified in conjunction with RETAIN=YES and/or REDIRECT, the output file is put on hold. When it is released, the other appropriate actions will result.

If omitted, the file is not held.

```
SECURE={YES}
        {NO }
```

Specifies that a print subfile destined for a workstation auxiliary printer is either secured or not secured.

If YES is specified, the file is secured. The initiator must be logged on when the output is available for printing. Otherwise, the subfile will not be printed.

If NO is specified, the file is not secured. The initiator does not have to be logged on when the output is available for printing.

Buffers may be specified for each file by using the nXm parameter in the SPOOL jproc. These buffers are used only for that specific file and are available any time the file is logically opened. All files not having dedicated buffers share a common buffer with the log. The size of the common buffer may be explicitly given using the JOB statement.

When buffers are selected, the following considerations apply:

1. In general, if only one or two spooled files are active concurrently, more efficient operation should be realized by allowing all files to share the common buffer. The system keeps the currently active files in the available buffers.
2. Larger buffers reduce the number of accesses to the spool file. Even though a slight delay may occur when a single large buffer is transferred, the overall efficiency as compared to multiple small buffers is greater.
3. For a given file, more than two buffers are useful only if the data is transferred in short, separated bursts.
4. Preoperations are always issued on transfer of the last record of a buffer for reads to the spool file. They are issued, if possible, for the last user write to a buffer. "If possible" implies either:
 - another buffer exists associated with this file; or
 - the current record fills the current buffer to 96 percent.

Preoperations do not suspend the requesting user.

//[symbol] ROUTE d1, . . . , dn

Specifies up to eight destinations for individual print and punch files. These destinations are the central printer or punch at a local or DDP site, a workstation auxiliary printer at a DDP site, or an auxiliary printer that is locally or remotely connected to your system. The destinations are as follows:

CENTRAL

Destination is the central printer or punch at the local site.

host-id:CENTRAL

Destination is the central printer or punch at the DDP site identified by the 1- to 4-character host-id.

host-id:user-id

Destination is a DDP site (identified by the 1- to 4-character host-id) at which the printer output will be routed to an auxiliary workstation printer (identified by the 1- to 6-character user-id).

user-id

Destination is an auxiliary printer on a workstation (identified by the 1- to 6-character user-id) that is locally or remotely connected to your system.

\$Y\$MAS

Destination is the auxiliary printer if the master at the local site is a workstation. Otherwise, destination is the central printer at the local site.

D.2.2. Spooling Macroinstruction

Used to create a breakpoint in an active subfile in the spool file. Breakpoint closes and reopens the subfile as it is being generated by the job. Each segment created by the breakpoint is considered a logical subfile so that output to the physical device can be started before job step termination. This macroinstruction can be used only in assembler programs.

■ Breakpointing an Active File (BRKPT Macroinstruction)

[symbol] BRKPT { filename
 { CCB-name
 (1)

Breakpoints printer or punch file currently being created that is associated with file name or CCB-name specified. For programs using data management macroinstructions to define and access the file, use filename to specify symbolic address or DTF macroinstruction associated with file to be breakpointed. For programs using physical IOCS macroinstructions, use CCB-name to specify symbolic address of command control block (CCB) associated with file to be breakpointed. Specify (1) when the address of the appropriate DTF or CCB is already loaded into register 1.

D.2.3. Workstation User Spooling Commands

Used to display and manipulate subfiles in the spool file that are associated with the user-id of the workstation user logged on. Only those spooled files created by jobs initiated or running under the same workstation user-id, and the workstation log created by the current workstation user, can be accessed.

The formats for these commands may include:

- Spool File Directories

`directory`

Identifies the specific spool file directory acted upon. When ALL is entered in place of a directory, all directories accessible to the command are acted upon. Allowable entries are:

`LOG`

Job and workstation log input and output is in designated log subfile.

`PRINT`

Subfile output is to the designated printer.

`PUNCH`

Subfile output is to the designated card punch or data set label diskette.

`RDR`

Subfile input is from designated card reader or data set label diskette.

- Spool File Command Modifiers

`modifier-1,...,modifier-n`

Optionally used to further identify, within a specified directory, the subfiles being referenced. Any number of modifiers may be specified. Allowable modifiers are:

`ACCT=acctno`

One to four characters

`BNUM=binary job-number`

Five-digit binary job number. Used with BEGIN SPL, DELETE SPL, DISPLAY SPL, DISPLAY ACT, HOLD SPL, PR, PU, PD, and RP commands

`CART=cartridge-id`

Print cartridge identification, one to eight characters



$$\text{DEV} \left\{ \begin{array}{l} 770 \\ 776 \\ 789 \end{array} \right\}$$

Any device of the type designated that is available.

FILE=filename

One to eight characters

FORM=formname

One to eight characters

JOB=jobname

One to eight characters

SIEP=stepno

Three characters (left-justified with zeros)

■ Output Writer Function Codes

All of the output writer function codes listed in D.2.6 with the exception of INPUT or DEVICE can be used with the RP, PR, and PU workstation commands.

NOTE:

Command keyins cannot exceed 28 characters in length, including commas. This applies to all commands except DELETE SPL and HOLD SPL, which can have a keyin of up to 60 characters in length, including commas. If OUT=NO is specified for the BEGIN SPL command, you may enter a keyin of up to 60 characters.

■ Holding Output Writer at the Mount Message

The MOUNT message reply provides a subfile hold option. The format of the MOUNT message is:

MOUNT ON DEV=___ FORM=_____ LPI=_ BAND=_____ RHI?

The H reply to the MOUNT message closes the open spool subfile and places it in a hold state on the spool queue. The currently active output-writer remains active and continues processing according to the criteria specified at load time.

NOTE:

Respond with R after you make the change, or respond with I if the form or cartridge cannot be used.

■ Removing Closed Files from Hold Condition (BEGIN SPL)

```

BEGIN SPL [ { ALL } ] [ , modifier-1 , . . . ,
           { PRINT }
           { PUNCH }
           { LOG }
           { RDR }
modifier-n ] [ { OUT=did }
              { NO } ]

```

Removes hold status from closed files in directory named, according to modifiers specified. Also loads output writer automatically in burst mode to process files when printer becomes available (does not interrupt active printer when command is from workstation).

If a job is initiated from a workstation and the spool subfile is destined for an auxiliary printer connected to that workstation, this command loads the RP output writer in the burst mode according to the parameters entered in the BEGIN SPL command when the spool subfile is released from the held state.

If RDR is entered without LBL=label modifier, all reader spooled files associated with workstation user-id are removed from hold condition.

The OUT parameter cannot be used if ALL or RDR is specified.

■ Breakpointing an Active File (BRKPT)

```

BRKPT { P } , { PRINT }
      { I } , { PUNCH }

[ , modifier-1 , . . . , modifier-n ]
, JOB=jobname [ , HOLD ]

```

Breakpoints printer or punch file currently being created by job name and modifiers specified. P breakpoints file at end of page; I breakpoints file immediately. Diskette files cannot be breakpointed.

This command closes files and makes them available to output writer. New file is created that contains remainder of file.

After breakpoint is taken, an output writer in burst mode should be loaded to process the file.

A breakpoint with the HOLD option for a print file places all subsequent subfiles of the file in a hold condition.

■ Breakpointing the Workstation Log File (BRKPT LOG)

```

BRKPT LOG [ , OUT= { TAPE
                   { DISK
                   { DISKETTE } ] ] [ , HOLD ]

```

Breakpoints workstation log file to print it or redirect it to specified output device. New workstation log file starts with first message or command after breakpoint command.

After BRKPT is issued for workstation log, output writer is loaded automatically to print log.

The HOLD parameter holds the breakpointed LOG subfile, but not the newly opened LOG subfile until another hold command is issued.





■ Changing the Device Type and/or the Number of Copies for Spooled Files

```

CHANGE SPL, { ALL
             LOG
             PRINT }
[ ,modifier-1 ... modifier-n]
[ ,COPIES=nnn]
[ [ ,DVC={ 770
          776
          PPC
          ANY
          CLASS1
          CLASS2
          CLASS3 } ] ]
[ [ ,DVC=AUX, ID= { *
                  user-id } ] ]
[ [ ,ID= { *
        user-id } ] ]
    
```

Alters the device type and/or the number of copies for a spooled file.

There are no parameter defaults. To change the number of copies and/or the device, the parameter must be entered.

Changes the closed spool files of either queued or held status, but not files of active status.

■ Deleting Closed Files (DELETE SPL)

```

DELETE SPL, { ALL
             PRINT
             PUNCH
             LOG
             RDR } [ ,modifier-1,....
modifier-n]
    
```

Deletes closed files in directory named, according to modifiers specified. Includes queued files (waiting for output writer processing) and files in hold condition. Active files and files in progress (being processed) cannot be deleted.

If RDR is entered without LBL=label modifier, all reader spooled files associated with workstation user-id are deleted.

■ Displaying the Status of Active Files (DISPLAY ACT)

```

DISPLAY ACT [ { ALL
              PRINT
              PUNCH } ] [ ,modifier-1,....
modifier-n]
    
```

Displays the number of files currently being created in directory named, according to modifiers specified.

Solicited messages are used to direct and terminate display. Includes option to display further information about some or all of files specified, in either detailed or abbreviated format.

■ Displaying the Status of Completed Files (DISPLAY SPL)

```

DISPLAY SPL [ { ALL
              PRINT
              PUNCH
              LOG
              RDR } ] [ ,modifier-1,....
modifier-n]
    
```

Displays the number of completed files (queued for processing, on hold, and in progress) in directory named, according to modifiers specified.

Solicited messages are used to direct and terminate display. Includes option to display further information about some or all of files specified, in either detailed or abbreviated format.

- **Displaying the Workstation Log Status (DISPLAY LOG)**

DISPLAY LOG

Displays the number of lines accumulated in current workstation log file.

- **Placing Closed Files in Hold Condition (HOLD SPL)**

HOLD SPL [**ALL**] [,modifier-1,....

| |
|-------|
| PRINT |
| PUNCH |
| LOG |
| RDR |

Places on hold closed files in directory named, according to modifiers specified.

modifier-n]

- **Manually Loading an Output Writer**

{PR} [(did)] [function-code]

[modifier-1,....,modifier-n]

Loads output writer for nonauxiliary destined printer (PR) or punch (PU) at central site on behalf of current logged-on user. If device address (did) is omitted, system assumes only one device is available and selects the first available device. When the device address is tape, disk, or diskette unit, output is directed to that device.

Function code specifies output writer mode of operation and processing criteria. (See D.2.3.) If omitted, output writer is loaded in mode (nonburst or burst) assigned at SYSGEN. After the function is entered and the request is completed, the system usually requests entry of another function. Enter the function using solicited message reply format (described in D.2.6) or press transmit key if no additional functions are required.

Modifiers further identify the file to be processed. (See D.2.3.)

- **Manually Loading Output Writer for Workstation Auxiliary Printer**

RP [function-code]

[modifier-1,....,modifier-n]

Loads output writer for the auxiliary printer connected to a local or remote workstation. All output directed to the auxiliary printer is printed on behalf of the current logged-on user at the workstation auxiliary printer.

Function code specifies output writer mode of operation and processing criteria. (See D.2.3.) If omitted, output writer is loaded in mode (nonburst or burst) assigned at SYSGEN. After the function is entered, the system usually requests entry of another function. Enter the function using solicited message reply format (described in D.2.6) or press transmit key if no additional functions are required.

Modifiers further identify the file to be processed. (See D.2.3.)

D.2.4. Operator General Spooling Commands

Used to display and manipulate subfiles in the spool file; also used to change the operating mode of the spooling function.

The formats for these commands may include:

- Spool File Directories



directory

Identifies the specific spool file directory acted upon. When ALL is entered in place of a directory, all directories accessible to the command are acted upon. Allowable entries are:

DDPPR

Distributed data processing output is to printer.

DDPPU

Distributed data processing output is to card punch.

LOG

Job, workstation, and console log input and output is in designated log subfile.

PRINT

Subfile output is to the designated printer.

PUNCH

Subfile output is to the designated card punch or data set label diskette.

RBPIN

Remote batch processing input is from card reader.

RBPPR

Remote batch processing output is to printer.

RBPPU

Remote batch processing output is to card punch.

RDR

Subfile input is from designated card reader or data set label diskette.

■ Spool File Command Modifiers

modifier-1, ..., modifier-n

Optionally used to further identify, within a specified directory, the subfiles being referenced. Any number of modifiers may be specified. Allowable modifiers are:

ACCT={acctno}

One to four characters

BNUMB=binary jobno

Five-digit binary job number. Used with BEGIN SPL, DELETE SPL, HOLD SPL, PR, PU, and RP commands.



$$\underline{\text{CART}} = \left\{ \begin{array}{l} \text{cartridge-id} \\ \cdot \end{array} \right\}$$

Print cartridge identification, one to eight characters

$\underline{\text{DDPID}} = \text{host-id}$

One to four characters (output writer BX function only)

$$\underline{\text{DEV}} = \left\{ \begin{array}{l} 770 \\ 776 \\ 789 \\ * \end{array} \right\}$$

Any device of the type designated that is available.

$$\underline{\text{FILE}} = \left\{ \begin{array}{l} \text{filename} \\ \cdot \end{array} \right\}$$

One to eight characters

$$\underline{\text{FORM}} = \left\{ \begin{array}{l} \text{formname} \\ \cdot \end{array} \right\}$$

One to eight characters

$\underline{\text{ID}} = \text{remote-id}$

One to six characters (output writer BX function only)

$$\underline{\text{JOB}} = \left\{ \begin{array}{l} \text{jobname} \\ \cdot \end{array} \right\}$$

One to eight characters

$\underline{\text{LBL}} = \text{labelname}$

One to eight characters for diskette; 1-17 characters for card reader

$\underline{\text{STEP}} = \text{stepno}$

Three characters (left justified with zeros)

$\underline{\text{VOL}} = \text{volno}$

One to six characters (diskette only)

NOTES:

1. Command keyins cannot exceed 28 characters in length, including commas. This applies to all commands except DELETE SPL and HOLD SPL, which can have a keyin of up to 60 characters in length, including commas. If OUT=NO is specified for the BEGIN SPL command, you may enter a keyin of up to 60 characters.
2. Enter * and a modifier type to group spool file contents for processing on a first-in, first-out basis. Provides a complete spool file listing, grouped according to the modifier specified.

■ Removing Active Files from Hold Condition (BEGIN ACT)

```
BEGIN ACT [ ( ALL
             ( PRINT
             ( PUNCH
             ( LOG ) ]
```

Removes hold status from files currently being created in directory named.

LOG does not affect remote batch or distributed data processing job logs.

Command is ignored if modifiers entered.

■ Removing Closed Files from Hold Condition (BEGIN SPL)

```
BEGIN SPL [ ( ALL
             ( DDPPR
             ( DDPPU
             ( PRINT
             ( PUNCH
             ( LOG
             ( RBPPR
             ( RBPPU
             ( RDR ) ] [ ,modifier-1, . . . ,
modifier-n ]
```

Removes hold status from closed files in directory named, according to modifiers specified. Also loads output writer automatically in burst mode to process files.

LOG does not affect remote batch or distributed data processing job logs.

If RDR is entered without LBL=label modifier, all reader spooled files are removed from hold condition.

If ALL is entered or implied by default, all directories are affected.

■ Removing Closed Files from Hold Condition and Designating Output Device (BEGIN SPL)

```
BEGIN SPL ( LOG
           ( PRINT
           ( PUNCH
modifier-n ], OUT= { did
                  ( NO )
```

Removes hold status from closed files in directory named, according to modifiers specified, and specifies the designated device (did) that the output is to be printed or punched on. Also loads output writer automatically in burst mode to process files.

LOG does not affect remote batch or distributed data processing job logs.

If OUT=NO is specified, the output writer is not loaded. When this parameter is specified, the command keyin can be up to 60 characters in length, including commas.

■ Removing Active and Closed Files from Hold Condition (BEGIN SPQ)

```
BEGIN SPQ [ ( ALL
            ( PRINT
            ( PUNCH
            ( LOG ) ]
```

Combines BE ACT and BE SPL commands. Removes hold status from both currently active and closed files in directory named. Also loads output writer automatically in burst mode to process files.

LOG does not affect remote batch or distributed data processing job logs.

Command is ignored if modifiers entered.

■ Breakpointing an Active File (BRKPT)

→ BRKPT { P } . { PRINT } . JOB=jobname [, HOLD]
 { I } { PUNCH }
 [, modifier-1 , . . . , modifier-n]

Breakpoints printer or punch file currently being created by job name and modifiers specified. P breakpoints file at end of page; I breakpoints file immediately. Diskette files cannot be breakpointed.

This command closes files and makes them available to output writer. New file is created that contains remainder of file. BRKPT is used whenever warning messages indicating spool file is nearly depleted appear on console screen.

After breakpoint is taken, operator should load output writer in burst mode to process the file.

Breakpointing a print file with the HOLD option places all subsequent subfiles of the breakpointed file in a HOLD condition also.

■ Breakpointing the Console Log File (BRKPT CNSLG)

→ BRKPT CNSLG [, OUT= { TAPE }] [, HOLD]
 { DISK }
 { DISKETTE }

Breakpoints console log file to print it or redirect it to specified output device. New console log file starts with first message or command after breakpoint command.

After BRKPT is issued for console log file, output writer is loaded automatically. Output writer recognizes the console log file and prints it.

Console log record collection must be set to ON with option to permit printing (specified via SET SPL,CN command currently in effect).

Breakpointing a print file with the HOLD option places all subsequent subfiles of the breakpointed file in a HOLD condition also.

■ Changing the Device Type and/or the Number of Copies for Spooled Files

→ CHANGE SPL , { ALL }
 { LOG }
 { PRINT }
 [, modifier-1 . . . modifier-n]
 [, COPIES=nnn]
 { [, DVC= { 770 }]
 { 776 }
 { PPC }
 { ANY }
 { CLASS1 }
 { CLASS2 }
 { CLASS3 }]
 [, DVC=AUX , ID= { * }
 { user-id }]]
 [, ID= { * }
 { user-id }]]

Alters the device type and/or the number of copies for a spooled file.

There are no parameter defaults. To change the number of copies and/or the device, the parameter must be entered.

Changes the closed spool files of either queued or held status, but not files of active status.

■ Deleting Closed Files (DELETE SPL)

```

DELETE SPL, ( ALL ) [ ,modifier-1, ...,
             ( DDPPR )
             ( DDPPU )
             ( PRINT )
             ( PUNCH )
             ( LOG )
             ( RBPPR )
             ( RBPPU )
             ( RDR )
modifier-n ]

```

Deletes closed files in directory named, according to modifiers specified. Includes queued files (waiting for output writer processing) and files in hold condition. Active files and files in progress (being processed) cannot be deleted.

LOG does not affect remote batch or distributed data processing job logs.

IF RDR is entered without LBL=label modifier, all reader spooled files are deleted.

If ALL is entered, all directories are affected.

■ Displaying the Status of Active Files (DISPLAY ACT)

```

DISPLAY ACT ( ALL ) [ ,modifier-1, ...,
            ( DDPPR )
            ( DDPPU )
            ( PRINT )
            ( PUNCH )
            ( RBPPR )
            ( RBPPU )
modifier-n ]

```

Displays the number of files currently being created in directory named, according to modifiers specified.

Solicited messages are used to direct and terminate display. Includes option to display further information about some or all of files specified, in either detailed or abbreviated format.

■ Displaying the Status of Completed Files (DISPLAY SPL)

```

DISPLAY SPL ( ALL ) [ ,modifier-1, ...,
            ( DDPPR )
            ( DDPPU )
            ( PRINT )
            ( PUNCH )
            ( LOG )
            ( RBPIN )
            ( RBPPR )
            ( RBPPU )
            ( RDR )
modifier-n ]

```

Displays the number of completed files (queued for processing, on hold, and in progress) in directory named, according to modifiers specified.

Solicited messages are used to direct and terminate display. Includes option to display further information about some or all of files specified, in either detailed or abbreviated format.

LOG does not affect remote batch or distributed data processing job logs.

■ Displaying the Console Log Status (DISPLAY CNSLG)

DISPLAY CNSLG

Displays the number of lines accumulated in current console log file.

■ Displaying the Spooling Mode of Operation (DISPLAY SPL,STATUS)

DISPLAY SPL,STATUS

Displays burst or nonburst mode of operation currently in effect. For burst mode, includes selection criteria when specified.

Also displays the percentage of spool file space available.



- **Placing Active Files in Hold Condition (HOLD ACT)**

```
HOLD ACT [ . ( ALL )
           ( PRINT )
           ( PUNCH )
           ( LOG ) ]
```

Places on hold files currently being created in directory named.

LOG does not affect remote batch or distributed data processing job logs.

Job logs for jobs found in error by run processor are not affected.

Command is ignored if modifiers entered.

- **Placing Closed Files in Hold Condition (HOLD SPL)**

```
HOLD SPL [ . ( ALL ) [ .modifier-1, . . . ,
              ( DDPFR
              ( DDPFU
              ( PRINT
              ( PUNCH
              ( LOG
              ( RBPRR
              ( RBPFU
              ( RDR
modifier-n ]
```

Places on hold closed files in directory named, according to modifiers specified.

LOG does not affect remote batch or distributed data processing job logs.

If ALL is entered or implied by default, all directories are affected.

- **Placing Active and Closed Files in Hold Condition (HOLD SPQ)**

```
HOLD SPQ [ . ( ALL )
           ( PRINT )
           ( PUNCH )
           ( LOG ) ]
```

Combines HO ACT and HO SPL commands. All closed files in directory named are immediately placed in hold condition; all active and future files in directory are placed on hold when closed.

LOG does not affect remote batch or distributed data processing job logs.

Job logs for active jobs found in error by run processor are not affected; job logs for closed jobs that were found in error are held.

Command is ignored if modifiers entered.

- **Setting Spooling System Operating Mode (SET SPL,BURST/NBURST)**

```
SET SPL, { BURST [ ,modifier ]
          ( NBURST ) }
```

Sets spooling system operating mode for processing output files to burst or nonburst (NBURST). All subsequently loaded output writers run in mode selected. Previously loaded output writers are not affected.

A modifier tailors file selection for burst mode processing.

■ **Setting Console Log and Workstation Log File Specifications (SET SPL,CNSLG)**

```
SET SPL,CNSLG [ { ON } ] [ { RETAIN } ]
               [ { OFF } ] [ { DELETE } ]
               [ { PRINT } ]
               [ { NOPRINT } ]
```

Turns console log and workstation log record collection function on and off, deletes or accumulates (RETAIN) console log for transfer to SYSLOG, and prints or does not print console log.

Must be set to ON to accumulate or print console log or to accumulate or print workstation log via SET SPL,PRINT and SET SPL,DUMP commands respectively.

If any optional parameters are omitted, console log file remains in same condition or mode for that parameter that it was in before the command.

Blinking marker symbol (█) in rightmost position of console line indicates message not written to console log.

■ **Setting Accumulation of Job Log and Workstation Log Files (SET SPL,DUMP/ENDDUMP)**

```
SET SPL, { DUMP }
         { ENDDUMP }
```

Deletes or accumulates job log and workstation log files after printing for transfer to SYSLOG. DUMP specifies files are accumulated for later routing to SYSLOG tape or disk file; ENDDUMP specifies no accumulation.

The SET SPL,CN,ON command must be in effect to accumulate workstation logs.

■ **Setting Job Log and Workstation Log Printing Specifications (SET SPL,NOACT/NOLOG/NOPRINT/PRINT)**

```
SET SPL, { NOACT }
         { NOLOG }
         { NOPRINT }
         { PRINT }
```

Specifies what job log and workstation log records are to be printed: job log (L and A) records at end of job and workstation log (W and R) records at breakpoint or end of session. NOACT specifies not to print accounting (A) records; NOLOG specifies not to print log (L) records and workstation (W and R) records; NOPRINT specifies not to print job accounting (A) and log (L) records, and workstation (W and R) records; and PRINT specifies printing all job log (A and L) and workstation log (W and R) records.

JOBACCT=YES must be specified at SYSGEN to control A record printing; SET SPL,CN,ON command must be in effect to control W record printing.

■ **Setting Spool File Printout Format (SET SPL,NOHDR/HEADER)**

```
SET SPL, { NOHDR }
         { HEADER }
```

Suppresses (NOHDR) or prints (HEADER) 3-page header preceding each spooled print file.

■ **Setting Forms Change Message (SET SPL,TEST/NOTEST)**

```
SET SPL, { TEST }
         { NOTEST }
```

Displays (TEST) or suppresses (NOTEST) console message indicating a forms change with option to print test lines.

D.2.5. Operator Input Reader Command

Used to load an input reader symbiont to transfer a card or data set label diskette file into the RDR spool directory.

■ Spooling Punched Card Input Files

```
IN[(did)] { 51 }
           { 66 }
```

Loads input reader for spooling card input. Specify either 51- or 66-column cards: If omitted, 80-column (or 96-column, if configured) cards are read.

A // DATA job control statement must precede jobs or data files to be spooled. Cards are read until a // FIN or another // DATA statement is detected.

If no device (did) is specified, first available card reader (SYSRDR) is expected to contain input file.

■ Spooling Tape Input Files

```
IN(did)
```

Loads input reader for spooling tape input.

A // DATA job control statement must precede jobs or data files to be spooled.

Use only with single volume input.

■ Spooling Data Set Label Diskette Input Files

```
IN([did],label) [RETAIN][,DELETE]
```

Loads input reader for spooling diskette input. Label must match LBL job control statement file name, with maximum of eight characters. RETAIN specifies spooled file is retained after job processing. Retained file is available for additional processing until deleted via DELETE command or if the DELETE parameter is used in a subsequent introduction of a spooled subfile having the same file label.

Use only with single-volume input.

D.2.6. Operator Output Writer Commands

Used to load an output writer symbiont according to the operating mode and selection criteria specified. Solicited and unsolicited messages are used to direct the operation of output writers that are active in the system.

Output writers are loaded under three conditions:

1. Automatically by system when files require processing and appropriate device is available
2. Automatically when operator enters BEGIN spooling command
3. Manually by operator using output writer commands and messages. Function codes and modifiers are included to change operating mode, select criteria, and tailor processing.

Under certain conditions, such as system set in nonburst mode, the operator manually loads output writers to have files printed. Other conditions include:

1. Warm start (recover files at IPL) if set to nonburst at SYSGEN
2. BR function if system is in nonburst mode
3. STOP or HALT function previously entered from console and printing of job's files is incomplete
4. IN function when reintroducing redirected tape, disk, or format label diskette output

■ Manually Loading Output Writer

```
{ PD } [(did)] [function-code]
{ PR }
{ PU }
```

```
[,modifier-1,...,modifier-n]
```

Loads output writer for printer (PR), punch (PU), or data set label diskette (PD) file processing.

If device address (did) omitted, system assumes only one device is available and selects first available device. When device address is tape, disk, or diskette unit, output file is redirected to that device (not valid with PD). PD does not require device address; job control device assignment set provides diskette location to be used.

Function-code specifies output writer mode of operation and processing criteria. (Function codes are described in later paragraph.) If omitted, output writer is loaded in mode (burst or nonburst) assigned at SYSGEN. After function is entered and request is completed, system usually requests entry of another function. Enter function using solicited message reply format (described in following paragraphs) or press transmit key if no additional functions are required.

Command modifiers further identify files to be processed. (See D.2.4.)

■ Manually Loading Output Writer for Nonauxiliary Destined Print Files Created on Behalf of a Workstation User

```
PR [function-code],UID=user-id
[,modifier-1,...,modifier-n]
```

Loads output writer to print nonauxiliary destined print files created on behalf of the workstation user identified by UID. Function code specifies output writer mode of operation and processing criteria. (Function codes are described in a later paragraph.) If omitted, output writer is loaded in mode assigned at SYSGEN. After the function is entered and the request is completed, the system usually requests another function. Enter the function using solicited message reply format (described in the following paragraphs) or press the transmit key if no additional functions are required. UID is a 1- to 6-character, left-justified identification that was entered with the LOGON command at the workstation.

Modifiers further identify the file to be processed. (See D.2.4.)

■ Manually Loading an Output Writer for Output Destined for Workstation Auxiliary Printer

```
RP BX,UID=user-id
    [modifier-1,...,modifier-n]
```

Loads output writer to print output, specified as destined for the auxiliary printer (via // SPOOL jproc, // ROUTE job control statement, and OPTION job control statement), at the central site rather than at the auxiliary printer.

BX specifies that the output writer is to function in the burst mode.

After the function is entered, the system usually requests another function. Enter the function using solicited message format (described in the following paragraphs) or press the transmit key if no additional functions are required.

UID is a 1- to 6-character, left-justified identification that was entered with the LOGON command at the workstation.

Modifiers further identify the files to be processed. (See D.2.4.)

■ Directing Active Output Writer

Used to change operating mode or processing criteria of active output writer. Messages are either solicited (output writer requests function from operator) or unsolicited (operator interrupts output writer processing to enter a function). Unsolicited messages cannot be entered if output writer is waiting for response to solicited message.

- Solicited Message Reply Format

```
0i function-code
```

Used to respond to function request from output writer. Both job number (0) and message-id (i) must be included in reply. Message-id corresponds to the message-id transmitted by requesting output writer.

Allowable functions are described in list of output writer function codes.

- Unsolicited Message Reply Format

```
00 { PD } [(did)] function-code
    { PR }
    { PU }
```

Used to interrupt active output writer to issue a function change. The job number and message-id (00) are used to communicate with output writer (via the supervisor).

PR (printer), PU (punch), or PD (data set label diskette), identifies the output writer to be interrupted. If device address (did) omitted, system assumes output writer is active at first available device.

Allowable functions are described in list of output writer function codes. Use the DEV function code to redirect output currently being processed.

■ Output Writer Function Codes

BURST [,modifier-1, . . . ,modifier-3]
BX }

Places the output writer in burst mode. Functions can be further qualified by optional modifiers 1 through 3. Modifiers that may be specified are listed in the description of general operator commands (D.2.4). (STEP, LBL, and VOL are not used with BU and BX.) If BX is entered with modifiers, output writer terminates after processing all files that satisfy the modifiers. If BU is entered, output writer requests another function if more files exist that do not satisfy the modifiers.

BYPASS

Terminates processing of the current file. Current file is closed and output writer continues processing next file. Bypassed files can be restarted later.

COPIES, nnn

Sets the number of copies the output writer is to produce for each file it processes. From 1 to 255 copies (nnn) may be specified. If 0 is specified, 1 is assumed. File is closed when processing is completed. This function cannot be used with PD or for files being processed from a redirected disk, diskette, or tape. ←

DELETE

Deletes the file being processed, and proceeds with next file to be processed.

DEVICE[,did]

Indicates that output writer is to be change device it is currently using to print (punch) its output. If a new device address (did) is specified, it is assigned to the output writer and current device is deallocated. If the did is omitted, a device having same characteristics as current device replaces the current device. The new device remains in use as long as same copy of output writer remains in main storage.

If output is to be redirected, specify the tape, disk, or diskette device (did) in the function code, and all subsequent output files are redirected to that device, as long as that copy of output writer is active. Redirected output to diskette is recorded in format label mode.

If some form of restart is to be performed before switching of device, enter the RESTART function before entering DEVICE function.

The output writer tape format is shown in Appendix A. Output writer disk and format label diskette format is shown in Appendix B.

DISPLAY

Displays the status of current file on the console screen. The information displayed is:

- | | |
|--------------------|--------------------------------|
| 1. File name | 5. Current page (card) number |
| 2. Job name | 6. Total pages (cards) in file |
| 3. Program name | 7. Number of remaining copies |
| 4. Job step number | 8. Existence of a breakpoint |

Function is ignored if no file is open.

HALT

Terminates output writer after the current file (if any) is processed. If file being processed has multiple copies, remaining copies are produced when output writer is reloaded.

HOLD

Places the current file in a hold state and begins processing next file.

Files in hold state are not available for processing until released by BEGIN command.

INPUT, did[,B] [, {RET}]
[{REL}]

Directs output writer to accept input from tape, disk, or diskette unit (did) identified in function code. Used to reintroduce redirected output so that it can be printed or punched.

If input is from disk or format label diskette, the B option may be included to permit specific files to be selected. When B is entered with function, a message is displayed requesting another function. Enter BX function with modifiers to specify particular files to be printed or punched.

Retains (RET) redirected print and punch output on disk or diskette after processing. Also releases (REL) redirected output previously retained. After processing, released files are deleted.

INPUT function may not be used with PD.

NBURST

Places output writer in nonburst mode. If specified while output writer is processing a file, function does not take effect until file processing is completed.

RETAIN

Retains currently active file in a HOLD state in spool file after it is processed. The retained file is unavailable for additional processing until released via BEGIN command. (Otherwise, delete the retained file via DELETE command.)

RESTART [, nnn
[, PAGE , nnnn]
[, CARD , nnnn]

Restarts processing of currently active file from a number of pages or cards. If no number is specified, output writer restarts processing from beginning of file. If only nnn is entered, file processing is restarted nnn pages or cards back from current position of file. If PA or CA is entered with nnnn, file is positioned back to page or card identified by nnnn.

RESTART cannot be used with PD.

Enter RESTART,nnn first if used in conjunction with DEVICE.

RESTART, { nnn }
{ LI , nnn }

Restarts processing at nnn lines from the current position or at the line number indicated by nnn in the log file being processed.

SD

Used for debugging purposes only. If specified, the message ENTER SPOOL DEBUG COMMAND is displayed. This message requires a reply of ALL, LOG, PRINT, or RDR to print the directories of the various spool file queues. When printing is completed, the message ENTER SPOOL DEBUG COMMAND is displayed again. Enter another debug command or enter HALT to terminate the spool debug.

SKIP { ,nnn
 , PAGE,nnnn }
 , CARD,nnnn }

Directs output writer to skip forward a specific number (nnn) of pages or cards; or to skip forward to specific page number (PAGE,nnnn) or card number (CARD,nnnn). After positioning, request is made for another function.

SKIP cannot be used with PD.

SKIP, { nnn
 LI, nnn }

Directs the output writer to skip nnn lines from the current position or to skip to the line number indicated by nnn in the log file being processed. ←

SIOP [,PAGE]

Directs output writer to stop processing. If PAGE is omitted, output writer terminates immediately. If PAGE is included, output writer terminates after printing the complete current page.

File being processed is closed but not deleted. When accessed by another output writer, file is processed from point at which it was closed.



D.3. PROCESSING THE SPOOL LOG FILE

Systems that are generated with spooling maintain a log subfile in the spool LOG file for each job processed in the system. The job log subfile contains the job's messages and job control statements (L records) and, when JOBACCT=YES was included during SYSGEN, the job's accounting (A) records. Spooling also maintains a workstation log subfile in the spool LOG file, provided the CONSOLOG parameter was included during SYSGEN. The workstation log subfile contains a record of all system messages written to and read from the workstation.

As each job or workstation session terminates, its associated job log subfile is closed and normally output to a high-speed printer as soon as the device becomes available. If the system is generated with the accumulate system job and workstation log files option (SYSLOG=YES) or if the operator issues the SET SPL,DUMP command after system initialization, the job and workstation log subfiles are marked as having been printed and then accumulated to allow further processing by user job accounting and bookkeeping programs.

A record of all messages written to and read from the console is also maintained in the spool LOG file, provided console logs are specified with the CONSOLOG parameter during SYSGEN. Console logs can be printed at any time by the operator and can also be accumulated for future use. They are accumulated when either the SYSGEN parameter RETAINLOG has been specified or when the operator enters the SET SPL,CNSLG command with the RETAIN parameter included.

Two programs are provided to assist the spool LOG file processing: the system log accumulation program (SL\$LOG) and, for job logs, the JOBLOG program. SL\$LOG transfers selected portions of the accumulated job and workstation log or accumulated console log subfiles from the spool LOG file to a SYSLOG file on disk or tape. After S\$LOG execution, the SYSLOG file is available for processing.

Usually SL\$LOG is used to dump the accumulated job and workstation log so it can be used as input to the JOBLOG program. The JOBLOG program generates a job accounting report based on the accumulated job and workstation log in the SYSLOG file. Since the SL\$LOG program may also be used to dump the accumulated console log to SYSLOG, the following steps should be performed in the sequence shown to ensure the integrity of SYSLOG output:

1. Execute the SL\$LOG program to dump the accumulated job and workstation log records.
2. Execute the JOBLOG program to produce a job accounting report.
3. Execute the SL\$LOG program to dump the accumulated console log records.
4. Execute a user-written program to process the console log output.

The SL\$LOG program is executed by using the RUN command with a job control stream (named DUMPLOG for disk or DUMPLOGT for tape) supplied by Sperry. The default processing options established for SL\$LOG can be tailored to a user's special processing conditions by changing the DUMPLOG call statement to include desired options. The SL\$LOG program is described in Section 8.

The operator also uses the RUN command to execute the JOBLOG program. Using the SYSLOG file as input, JOBLOG calls on the OS/3 independent sort/merge routine to create a sorted SYSLOG output file. After the sorting process is complete, JOBLOG uses the sorted file to produce the accounting report. A description of the JOBLOG program and four sample job control streams for producing tailored accounting reports are provided in Section 8.

■ Using Tape for the Job and Workstation Log SYSLOG File (RUN DUMPLOGT)

```
RUN DUMPLOGT [ , [ D={ACT LOG} ] [ V={vs n SYSLOG} ] ]
[ , C={N} ] ]
```

Establishes SYSLOG file on tape and transfers accumulated job and workstation log records from spool LOG file into SYSLOG.

```
D={ACT LOG}
```

ACT specifies that accounting records (A) are transferred. LOG specifies that log (L) and workstation (W and R) records are transferred.

If omitted, both are transferred.

```
V={vs n SYSLOG}
```

Identifies volume serial number of tape to be used.

If omitted, SYSLOG is used.

```
C={N}
```

Specifies whether or not checkpoint records are desired.

If omitted, checkpoint records are specified.

■ Using Disk for the Job and Workstation Log SYSLOG File (RUN DUMPLOG)

```
RUN DUMPLOG , F=ALLOC [ , D={ACT LOG} ]
```

Establishes SYSLOG file on disk and transfers accumulated job and workstation log records from spool LOG file into SYSLOG.

```
F=ALLOC
```

Specifies SYSLOG file space is allocated on disk.

```
D={ACT LOG}
```

ACT specifies that accounting records (A) are transferred. LOG specifies that log (L) and workstation (W and R) records are transferred.

If omitted, both are transferred.

- Adding Job and Workstation Log Subfiles to the SYSLOG File on Disk (RUN DUMPLOG)

RUN DUMPLOG [, , D={ACT
LOG}]

Adds additional accumulated job and workstation log subfiles to existing SYSLOG file.

D={ACT
LOG}

ACT specifies that accounting records (A) are transferred.
LOG specifies that log (L) and workstation (W and R) records are transferred.

If omitted, both are transferred.

- Reinitializing the Job and Workstation Log SYSLOG File on Disk (RUN DUMPLOG)

RUN DUMPLOG , , F=INIT [, D={ACT
LOG}]

Reinitializes (effectively erases) SYSLOG file and transfers new accumulated job and workstation log subfiles from spool LOG file to SYSLOG.

F=INIT

Specifies SYSLOG file space is reinitialized on disk.

D={ACT
LOG}

ACT specifies that accounting records (A) are transferred.
LOG specifies that log (L) and workstation (W and R) records are transferred.

If omitted, both are transferred.

- Running JOBLOG Using Tape Input (RUN JBLOGT)

{RUN
RV} JBLOGT [, , V={vs n
SYSLOG} , S={A
B
C}]

Sorts tape file input to produce job accounting report.

V={vs n
SYSLOG}

Identifies volume serial number of tape containing job accounting file.

If omitted, SYSLOG is used.

S={A
B
C}

Specifies sort option. Jobs sorted in order submitted (A); in accounting number and job name order (default condition), with subtotals taken whenever accounting number and job name sequence changes (B); or in accounting number and job name order, with subtotals taken whenever accounting number field changes (C).

■ Running JOBLOG using Disk Input (RUN JBLOG)

$$\left. \begin{array}{l} \{ \text{RUN} \} \\ \{ \text{RV} \} \end{array} \right\} \text{JBLOG} \left[\begin{array}{l} \cdot, V = \{ \text{vsn} \} \\ \cdot, \{ \text{RES} \} \\ L = \{ \text{file-identifier} \} \\ \{ \text{SYSLOG} \} \end{array} \right], S = \left. \begin{array}{l} \{ \text{A} \} \\ \{ \text{B} \} \\ \{ \text{C} \} \end{array} \right\}$$

Sorts disk file input to produce job accounting report.

$$V = \left. \begin{array}{l} \{ \text{vsn} \} \\ \{ \text{RES} \} \end{array} \right\}$$

Identifies volume serial number of tape containing job accounting file.

If omitted, SYSRES (RES) is used.

$$L = \left. \begin{array}{l} \{ \text{file-identifier} \} \\ \{ \text{SYSLOG} \} \end{array} \right\}$$

Identifies file name of job accounting file.

If omitted, SYSLOG is name used.

$$S = \left. \begin{array}{l} \{ \text{A} \} \\ \{ \text{B} \} \\ \{ \text{C} \} \end{array} \right\}$$

Specifies sort option. Jobs sorted in order submitted (A); in accounting number and job name order (default condition), with subtotals taken whenever accounting number and job name sequence changes (B); or in accounting number and job name order, with subtotals taken whenever accounting number field changes (C).

■ Using Tape for the Console Log SYSLOG File (RUN DUMPLOGT)

$$\text{RUN DUMPLOGT} \cdot, D = \text{CON} \left[\begin{array}{l} \cdot, V = \{ \text{vsn} \} \\ \{ \text{SYSLOG} \} \\ \cdot, C = \left. \begin{array}{l} \{ \text{Y} \} \\ \{ \text{N} \} \end{array} \right\} \end{array} \right]$$

Establishes SYSLOG file on tape and transfers accumulated console log records from spool LOG file into SYSLOG.

$$D = \text{CON}$$

Specifies console log (C) records only are transferred.

$$V = \left. \begin{array}{l} \{ \text{vsn} \} \\ \{ \text{SYSLOG} \} \end{array} \right\}$$

Identifies volume serial number of tape to be used.

If omitted, SYSLOG is used.

$$C = \left. \begin{array}{l} \{ \text{Y} \} \\ \{ \text{N} \} \end{array} \right\}$$

Specifies whether checkpoint records are desired.

If omitted, checkpoint records are specified.

■ Using Disk for the Console Log SYSLOG File (RUN DUMPLOG)**RUN DUMPLOG , , F=ALLOC , D=CON**

Establishes SYSLOG file on disk and transfers accumulated console log records from spool LOG file into SYSLOG.

F=ALLOC

Specifies SYSLOG file space is allocated on disk.

D=CON

Specifies console log records only are transferred.

■ Adding Console Log Subfiles to the SYSLOG File on Disk (RUN DUMPLOG)**RUN DUMPLOG , , D=CON**

Adds additional accumulated console log subfiles to existing SYSLOG file.

D=CON

Specifies console log records only are transferred.

■ Reinitializing the Console Log SYSLOG File on Disk (RUN DUMPLOG)**RUN DUMPLOG , , F=INIT , D=CON**

Reinitializes (effectively erases) SYSLOG file and transfers new accumulated console log subfiles from spool LOG file to SYSLOG.

F=INIT

Specifies SYSLOG file space is reinitialized on disk.

D=CON

Specifies console log records only are transferred.



Index

| Term | Reference | Page | Term | Reference | Page |
|---|-----------|------|---|-----------|------|
| A | | | B | | |
| Account summary listing, typical | Fig. 8-3 | 8-20 | Basic functions | | |
| Accounting record formats | | | input | 3.1 | 3-1 |
| accounting data (bytes 0-130) | 8.1.2.1.1 | 8-2 | output | 4.1 | 4-1 |
| accounting data sorting keys (bytes 133-255) | 8.1.2.1.2 | 8-12 | Breakpoint used to queue files from an incomplete job step | Fig. 4-4 | 4-10 |
| description | 8.1.2.1 | 8-2 | Breakpointed files written from the spool file | Fig. 4-5 | 4-11 |
| interactive services sorting keys (bytes 133-255) | 8.1.2.1.3 | 8-13 | Breakpointing | | |
| Accumulating and transferring | | | active spool subfiles (BRKPT) | C.2.6 | C-17 |
| console log | 4.6.1 | 4-23 | printer and punch files | 4.1.4.3.1 | 4-8 |
| job and workstation log | 4.6.3 | 4-29 | Burst mode | | |
| | 4.6.1 | 4-23 | output spooling | 4.1.4 | 4-3 |
| | 4.6.2 | 4-28 | using | 4.1.4.3 | 4-8 |
| Adding console log subfiles to SYSLOG file on disk (RUN DUMPLOG) | C.5.3.3 | C-45 | | | |
| Adding job and workstation log subfiles to SYSLOG file on disk (RUN DUMPLOG) | C.5.1.3 | C-38 | | | |
| Automatic job scheduling | 3.1.4 | 3-2 | | | |
| Auxiliary printer, workstation | | | | | |
| locally connected to system | 4.3 | 4-17 | | | |
| remotely connected to system | 4.3 | 4-17 | | | |

| Term | Reference | Page | Term | Reference | Page |
|--|--|------------------------------|---|------------------------|-------------------|
| C | | | D | | |
| Changing console log and workstation log file specifications (SET SPL,CNSLG) | C.2.11 | C-22 | Deleting closed spool subfiles (DELETE) | C.2.3 | C-12 |
| Changing job log and workstation log file print specifications (SET SPL,NOACT/NOLOG/NOPRINT/PRINT) | C.2.10 | C-21 | Device swapping | 4.2.8 | 4-17 |
| Changing spool file printout format (SET SPL,HEADER/NOHDR) | C.2.9 | C-20 | Directing operation of an output writer (messages) | C.4.4 | C-32 |
| Changing system accumulation of job log and workstation log files (SET SPL,DUMP/ENDDUMP) | C.2.8 | C-20 | Displaying spooling mode of operation (DISPLAY SPL,STATUS) | C.2.2 | C-11 |
| Changing system spooling criteria (SET SPL,BURST/NBURST) | C.2.7 | C-19 | Displaying status of spool subfiles (DISPLAY ACT/SPL/CNSLG) | C.2.1 | C-6 |
| Cold start | 7.2 | 7-2 | Distributed data processing | 6.1 6.3 Fig. 6-2 | 6-1 6-5 6-5 |
| Commands and messages, general spooling | | | DUMPLOG/DUMPLOGT job control stream | 4.6 | 4-22 |
| BEGIN | C.2.5 | C-15 | E | | |
| BRKPT | C.2.6 | C-17 | Establishing options during system generation | 9.3 | 9-2 |
| CH SPL | C.2.6A | C-19 | F | | |
| | D.2.3 | D-23 | File | | |
| | D.24 | D-26a | availability to output writer | Fig. 4-6 | 4-12 |
| DELETE | C.2.3 | C-12 | status at time of breakpoint | Fig. 4-3 | 4-9 |
| DISPLAY ACT/SPL/CNSLG | C.2.1 | C-6 | File retention | | |
| DISPLAY SPL,STATUS | C.2.2 | C-11 | input | 3.1.3 | 3-2 |
| HOLD | C.2.4 | C-13 | output | 4.1.3 | 4-3 |
| SET SPL,BURST/NBURST | C.2.7 | C-20b | Format (output writer) | | |
| SET SPL,CNSLG | C.2.11 | C-22 | disk | Appendix B | |
| SET SPL,DUMP/ENDDUMP | C.2.8 | C-20c | format label diskette | Appendix B | |
| SET SPL,HEADER/NOHDR | C.2.9 | C-20c | tape | Appendix A | |
| SET SPL,NOACT/NOLOG/NOPRINT/PRINT | C.2.10 | C-21 | | | |
| SET SPL,TEST/NOTEST | C.2.12 | C-24 | | | |
| Console log processing | 4.2.5 4.6.1 Fig. 4-10 Fig. 4-13 | 4-15 4-23 4-24 4-28 | | | |
| record format (SYSLOG) | 8.1.3 | 8-15 | | | |
| Control levels | 9.2 | 9-1 | | | |

| Term | Reference | Page | Term | Reference | Page |
|--|-----------------------------|------|---|-------------------------|----------------------|
| H | | | L | | |
| Headers to separate files | 4.2.2 | 4-13 | Loading an output writer (RP/PR/PU/PD) | C.4.1 C.4.2 C.4.3 | C-27 C-28 C-29 |
| I | | | M | | |
| Input | | | Log record format | 8.1.2.2 | 8-14 |
| diskette | 3.4 | 3-3 | | | |
| punched cards | 3.2 | 3-3 | | | |
| tape | 3.3 | 3-3 | Multiple | | |
| Input readers | | | copies of a file | 4.2.7 | 4-17 |
| command | C.3 | C-24 | destinations | 6.2 | 6-2 |
| description | 2.1.1 | 2-3 | | | |
| multiple | 3.1.1 | 3-2 | | | |
| operator commands | D.2.5 | D-34 | | | |
| Interactive services usage summary, typical | Fig. 8-4 | 8-21 | | | |
| J | | | N | | |
| JBLOG/JBLOGT job control stream | 8.2.1 | 8-19 | Nonburst mode | | |
| Job accounting | | | output spooling | 4.1.4 | 4-3 |
| concepts | Section 1 | | using | 4.1.4.3 | 4-8 |
| data format | Fig. 8-1 | 8-3 | Nonresident symbiont spooling functions | | |
| producing reports | Section 8 | | input | 3.1.2 | 3-2 |
| typical report | Fig. 8-2 | 8-20 | output | 4.1.2 | 4-3 |
| Job log | | | | | |
| ACT records | 4.2.3 | 4-14 | | | |
| LOG records | 4.2.3 | 4-14 | | | |
| printing | 4.2.3 | 4-14 | | | |
| processing | 4.2.3 | 4-14 | | | |
| | 4.6.1 | 4-23 | | | |
| | Fig. 4-9 | 4-23 | | | |
| | Fig. 4-12 | 4-26 | | | |
| record format (SYSLOG) | 8.1.2 | 8-2 | | | |
| sample | Fig. 4-7 | 4-14 | | | |
| subfile printout | Fig. C-1 | C-36 | | | |
| Job log report program (JOBLOG) | | | | | |
| accounting report types | 8.2.1 | 8-19 | | | |
| description | 8.2 | 8-18 | | | |
| executing JOBLOG | 8.2.2 | 8-22 | | | |
| sample job control streams | 8.2.3 | 8-23 | | | |
| JOBLOG program | See job log report program. | | | | |

| Term | Reference | Page | Term | Reference | Page |
|--|------------|------|--|-----------|------|
| O | | | R | | |
| Operator's spooling reference | Appendix C | | Reading input from cards and diskettes | Fig. 2-2 | 2-3 |
| Output | | | Record formats (SL\$LOG program) | | |
| diskette: data set label mode | 4.4 | 4-21 | accounting (A) | 8.1.2.1 | 8-2 |
| printer and card punch | 4.2 | 4-13 | console (C) | 8.1.3 | 8-15 |
| sample listing | Fig. 2-8 | 2-13 | log (L) | 8.1.2.2 | 8-14 |
| tape, disk, and format label | | | workstation (W and R) | 8.1.2.3 | 8-14 |
| diskette | 4.5 | 4-21 | Reinitializing SYSLOG file on disk (RUN DUMPLOG) | | |
| workstation auxiliary printer, locally or remotely connected | 4.3 | 4-17 | console log | C.5.3.4 | C-45 |
| 0791 correspondence quality type | 4.3 | 4-17 | job and workstation log | C.5.1.4 | C-39 |
| Output writers | | | Releasing spool subfiles (BEGIN) | C.2.5 | C-15 |
| burst mode | 4.1.4.2 | 4-6 | Remote batch processor | 6.1 | 6-1 |
| commands and messages | C.4 | C-26 | | 6.2 | 6-2 |
| description | 2.1.3 | 2-5 | | Fig. 6-1 | 6-2 |
| function codes | Table C-3 | C-30 | Remote spooling | Section 6 | |
| holding files from processing | 4.2.1 | 4-13 | Removing and replacing blanks in output records | Fig. 2-6 | 2-10 |
| multiple | 4.1.1 | 4-2 | Running JOBLOG using disk input (RUN JBLOG) | C.5.2.2 | C-41 |
| nonburst mode | 4.1.4.1 | 4-4 | Running JOBLOG using tape input (RUN JBLOGT) | C.5.2.1 | C-40 |
| operator commands | D.2.6 | D-34 | | | |
| page mode printing | 4.3 | 4-17 | | | |
| P | | | | | |
| Placing spool subfiles on hold (HOLD) | C.2.4 | C-13 | | | |
| Printing | | | | | |
| console log | 4.2.5 | 4-15 | | | |
| job log | 4.2.3 | 4-14 | | | |
| page mode | 4.3 | 4-17 | | | |
| workstation log | 4.2.6 | 4-16 | | | |
| Producing job accounting reports (JOBLOG program) | C.5.2 | C-40 | | | |
| Punched cards | | | | | |
| containing different column totals | 3.2.1 | 3-3 | | | |
| multiple input files | 3.2.2 | 3-3 | | | |

| Term | Reference | Page | Term | Reference | Page |
|---|------------|------|--|------------|------|
| S | | | | | |
| Saving accumulated subfiles in SYSLOG (SL\$LOG program) | | | Spooling element relationships | | |
| console log | C.5.3 | C-43 | basic | Fig. 2-1 | 2-2 |
| job and workstation log | C.5.1 | C-34 | distributed data processing | | |
| | | | devices included | Fig. 6-2 | 6-5 |
| | | | remote batch devices included | Fig. 6-1 | 6-2 |
| Spool file | | | Spooling environment, establishing | 2.2 | 2-7 |
| command modifiers | Table C-2 | C-5 | Spooling functions | | |
| description | 2.1.2 | 2-4 | input | Section 3 | |
| directories | Table C-1 | C-5 | output | Section 4 | |
| input and output during program execution | Fig. 2-3 | 2-5 | Spooling parameters | | |
| LOG | 4.6 | 4-22 | I/OGEN, virtual devices | D.1.2 | D-8 |
| processing aids | C.5 | C-34 | I/OGEN, workstation auxiliary printers | D.1.3 | D-9 |
| SYSPOOL | 1.2 | 1-1 | SUPGEN | D.1.1 | D-2 |
| | 2.1.2 | 2-4 | | | |
| | C.1 | C-3 | Spooling simplified | | |
| | D.1 | D-1 | after job run begins | 2.2.1.2 | 2-8 |
| Spool LOG file processing | D.3 | D-39 | before job is run | 2.2.1.1 | 2-8 |
| | | | SUPGEN spooling levels | 9.3 | 9-2 |
| Spooled files | | | Suppress or display forms change message (SET SPL,TEST/NOTEST) | C.2.12 | C-24 |
| deleting | 5.5 | 5-4 | SYSLOG file | 4.6 | 4-22 |
| displaying status | 5.2 | 5-1 | System log accumulation program (SL\$LOG) | | |
| holding | 5.3 | 5-2 | accumulating and transferring logs | 4.6 | 4-22 |
| releasing | 5.4 | 5-3 | console log output | 8.1.3 | 8-15 |
| Spooler | 2.1.2 | 2-4 | description | 8.1 | 8-1 |
| Spooling | | | job and workstation log output | 8.1.2 | 8-2 |
| basic purposes | 1.3 | 1-2 | output files | 8.1.1 | 8-1 |
| concepts | Section 1 | | transfer records to disk | 8.1.4 | 8-16 |
| control | Section 5 | | transfer records to tape | 8.1.5 | 8-17 |
| elements | Section 2 | | System operation | | |
| | Fig. 2-1 | 2-2 | with spooling | Fig. 1-2 | 1-4 |
| | Fig. 6-1 | 6-2 | without spooling | Fig. 1-1 | 1-3 |
| | Fig. 6-2 | 6-5 | \$SPOOL | 4.5 | 4-21 |
| establishing and controlling | Section 9 | | \$Y\$SPOOL | Appendix B | |
| general commands and messages | C.2 | C-4 | | 4.5 | 4-21 |
| initialization and file recovery | Section 7 | | | Appendix B | |
| job control statements | D.2.1 | D-11 | | Appendix B | |
| log processing | 4.6 | 4-22 | | 4.5 | 4-21 |
| | Fig. 4-8 | 4-22 | | | |
| macroinstruction | D.2.2 | D-22 | | | |
| operator general commands | D.2.3 | D-23 | | | |
| operator's reference | Appendix C | | | | |
| remote | Section 6 | | | | |
| summary | Appendix D | | | | |
| | Table 9-1 | 9-3 | | | |
| system control | D.2 | D-11 | | | |
| system generation | D.1 | D-1 | | | |
| workstation user commands | D.2.3 | D-23 | | | |



USER COMMENT SHEET

We will use your comments to improve subsequent editions.

NOTE: Please do not use this form as an order blank.

(Document Title)

(Document No.)

(Revision No.)

(Update No.)

Comments:

From:

(Name of User)

(Business Address)

CUT

FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 21 BLUE BELL, PA.

POSTAGE WILL BE PAID BY ADDRESSEE

SPERRY CORPORATION

ATTN.: SOFTWARE SYSTEMS PUBLICATIONS

P.O. BOX 500
BLUE BELL, PENNSYLVANIA 19424



FOLD



USER COMMENTS

We will use your comments to improve subsequent editions.

NOTE: Please do not use this form as an order blank.

(Document Title)

(Document No.)

(Revision No.)

(Update Level)

Comments:

From:

(Name of User)

(Business Address)

Fold on dotted lines, and mail. (No postage is necessary if mailed in the U.S.A.)
Thank you for your cooperation



FOLD



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

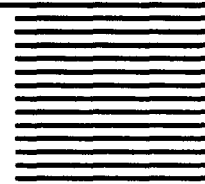
BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 21 BLUE BELL, PA.

POSTAGE WILL BE PAID BY ADDRESSEE

SPERRY CORPORATION

**ATTN: Documentation Quality Control Group
C/O SYSTEM PUBLICATIONS**



P.O. BOX 500
BLUE BELL, PENNSYLVANIA 19422-9990



FOLD

UNISYS

USER COMMENTS

We will use your comments to improve subsequent editions.

NOTE: Please do not use this form as an order blank.

(Document Title)

(Document No.)

(Revision No.)

(Update Level)

Comments:

From:

(Name of User)

(Business Address)

Fold on dotted lines, and mail. (No postage is necessary if mailed in the U.S.A.)
Thank you for your cooperation

CUT

FOLD



NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 21 BLUE BELL, PA.

POSTAGE WILL BE PAID BY ADDRESSEE

Unisys Corporation
E/MSG Product Information Development
PO Box 500 C1-NE6
Blue Bell, PA 19422-9990



FOLD



