

PERKIN-ELMER

**SPL/32 SPOOLER
SYSTEM ADMINISTRATION**

Reference Manual

48-056 F00 R00

The information in this document is subject to change without notice and should not be construed as a commitment by the Perkin-Elmer Corporation. The Perkin-Elmer Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license, and it can be used or copied only in a manner permitted by that license. Any copy of the described software must include the Perkin-Elmer copyright notice. Title to and ownership of the described software and any copies thereof shall remain in The Perkin-Elmer Corporation.

The Perkin-Elmer Corporation assumes no responsibility for the use or reliability of its software on equipment that is not supplied by Perkin-Elmer.

The Perkin-Elmer Corporation, Data Systems Group, 2 Crescent Place, Oceanport, New Jersey 07757

© 1983 by The Perkin-Elmer Corporation

Printed in the United States of America

TABLE OF CONTENTS

PREFACE		xi
CHAPTERS		
1	INTRODUCTION TO SPL/32	
1.1	INTRODUCTION	1-1
1.2	COMPONENTS OF THE SPOOLING PROCESS	1-3
1.2.1	Input Spooling	1-3
1.2.2	Output Spooling	1-3
1.2.3	The Spool Queue	1-3
1.2.4	The Pseudo Device	1-3
1.2.5	The Spool File	1-3
1.2.6	The Uniform Spooler Control Interface (USCI)	1-3
1.3	REVIEW	1-4
1.4	COMPONENTS OF SPOOLER CONTROL	1-5
1.4.1	Control and Subcontrol Tasks	1-5
1.4.2	Levels of Authorization	1-5
1.5	REQUIREMENTS FOR SPL/32	1-7
1.5.1	Hardware	1-7
1.5.2	Memory Requirements	1-7
1.5.3	Software Requirements	1-7
1.6	INITIALIZING SPL/32	1-8
1.6.1	Loading SPL/32	1-8
1.6.2	Linking SPL/32	1-9
1.6.3	Starting the Spooler	1-9
1.6.3.1	Creating a START Command File	1-10
1.6.4	SPL/32 System Console Interface	1-11
1.6.5	Ending the Spooler Task	1-12
2	INPUT SPOOLING	
2.1	INTRODUCTION	2-1
2.2	ESTABLISHING INPUT SPOOLING	2-1
2.3	INPUT SPOOLING CONTROL STATEMENTS	2-1
2.3.1	The /@INPUT Control Statement	2-1

CHAPTERS (Continued)

2.3.2	The /@SUBMIT Control Statement	2-3
2.3.3	Ending Input Spooling of a Card File	2-4
2.4	REMOVING INPUT SPOOLING DEVICES	2-5
3	THE SPOOL FILE AND THE SPOOL QUEUE	
3.1	INTRODUCTION	3-1
3.2	THE SPOOL FILE	3-1
3.2.1	Naming Spool Files	3-2
3.3	THE SPOOL QUEUE	3-4
3.3.1	Creating the Spool Queue	3-4
3.3.2	Displaying the Spool Queue File Name	3-4
3.3.3	Respecifying the Spool Queue	3-5
3.3.4	Respecifying the Spool Queue While the Spooler is Active	3-5
3.3.5	Interrogating the Spool Queue	3-5
3.3.5.1	Displaying Entries in the Spool Queue	3-5
3.3.5.2	Displaying Queue Entries According to Preprinted Form Requests	3-7
3.3.5.3	Displaying Files by Assigned Pseudo Device	3-7
3.3.5.4	Displaying Information About a Particular File	3-8
3.3.5.5	Displaying the Spooler Environment Configuration	3-8
3.3.5.6	Displaying All Spool Queue and Spooler Configuration Information	3-9
3.3.5.7	Functional Discussion of the INTERROGATE Command	3-10
4	OUTPUT SPOOLING	
4.1	INTRODUCTION	4-1
4.2	THE PSEUDO DEVICE	4-2
4.2.1	The Global Pseudo Device	4-2
4.2.2	Creating Pseudo Devices in the Spooling Environment	4-3
4.2.3	Associating Pseudo Devices with Physical Devices	4-4
4.2.4	Directing Output to a Pseudo Device	4-5
4.2.5	Limiting Default Access to a Pseudo Device	4-6
4.2.6	Establishing Unique Default Print Location	4-6
4.2.7	Disassociating Pseudo Devices From Physical Devices	4-7

CHAPTERS (Continued)

4.2.8	Deleting Pseudo Devices From the Spooling Environment	4-7
4.3	PHYSICAL OUTPUT DEVICE CONTROL	4-8
4.3.1	Altering the Status of Physical Output Devices	4-8
4.3.2	Removing Files From the Spool Queue	4-9
4.3.3	Removing Physical Devices From the Spooling Environment	4-10
4.4	USING PREPRINTED FORMS	4-10
4.4.1	Naming and Characterizing Preprinted Forms	4-11
4.4.2	Associating a Form With a Physical Device	4-11
4.4.3	Removing Forms From the Spooling Environment	4-12
4.5	USING IDENTIFICATION PAGES	4-13
4.6	USING THE CHECKPOINT FEATURE	4-14
4.6.1	Turning Checkpointing OFF	4-15
4.7	OUTPUT PRIORITY	4-16
4.7.1	Changing the Global Default Priority	4-16
4.7.2	Altering the Output Priority of a Spool File	4-17
5	TASK AND LEVELS OF AUTHORIZATION	
5.1	INTRODUCTION	5-1
5.2	CONTROL TASKS AND THEIR COMMUNICATION WITH THE SPOOLER	5-2
5.2.1	Command Messages	5-2
5.2.2	Acceptance Messages	5-2
5.2.3	Request For Mnemonic Table Messages	5-2
5.3	SPOOLER COMMUNICATION WITH CONTROL TASKS	5-3
5.3.1	Service Response Messages	5-3
5.3.2	Exception Response Messages	5-3
5.3.3	Authority Messages	5-3
5.3.4	Attention Messages	5-3
5.3.5	Mnemonic Messages	5-3
5.4	INCLUDING CONTROL TASKS IN THE SPOOLING ENVIRONMENT	5-4
5.5	DISPLAYING CONTROL TASKS IN THE SPOOLING ENVIRONMENT	5-4
5.6	REMOVING CONTROL TASKS FROM THE SPOOLING ENVIRONMENT	5-4

CHAPTERS (Continued)

5.7	SUBCONTROL TASKS AND THEIR COMMUNICATION THE SPOOLER	5-5
5.7.1	Command Messages	5-5
5.8	SPOOLER COMMUNICATION WITH SUBCONTROL TASKS	5-5
5.8.1	Service Response Messages	5-6
5.8.2	Exception Response Messages	5-6
5.9	INCLUDING SUBCONTROL TASKS IN THE SPOOLING ENVIRONMENT	5-6
5.10	DISPLAYING SUBCONTROL TASKS IN THE SPOOLING ENVIRONMENT	5-6
5.11	REMOVING SUBCONTROL TASKS FROM THE SPOOLING ENVIRONMENT	5-7
5.12	USER-LEVEL TASKS AND THEIR COMMUNICATION WITH THE SPOOLER	5-7
5.13	LEVELS OF AUTHORIZATION	5-8
5.13.1	Authorization Identifiers (AID)	5-10
5.13.2	Origin Identifiers (OID)	5-10
6	SPL/32 COMMANDS	
6.1	INTRODUCTION	6-1
6.2	ATTACH COMMAND	6-2
6.3	BANNER COMMAND	6-4
6.4	BLOCK COMMAND	6-7
6.5	CHANGE COMMAND	6-8
6.6	CHECKPOINT COMMAND	6-10
6.7	CREATE COMMAND	6-12
6.8	CTASK COMMAND	6-13
6.9	DELETE COMMAND	6-15
6.10	DETACH COMMAND	6-16
6.11	DEVICE COMMAND	6-17
6.12	END COMMAND	6-20
6.13	FLUSH COMMAND	6-21

CHAPTERS (Continued)

6.14	FORM COMMAND	6-23
6.15	GLOBAL COMMAND	6-26
6.16	GROUP COMMAND	6-27
6.17	INPUT COMMAND	6-29
6.18	INTERROGATE COMMAND	6-31
6.19	PRIMARY COMMAND	6-33
6.20	PRINT COMMAND	6-35
6.21	PRIORITY COMMAND	6-37
6.22	PUNCH COMMAND	6-38
6.23	QUEUE COMMAND	6-40
6.24	SCTASK COMMAND	6-42
6.25	SECONDARY COMMAND	6-44
6.26	SPOOLFILE COMMAND	6-45
6.27	SPVOLUME COMMAND	6-48
6.28	TRAILER COMMAND	6-50

APPENDIXES

A	SPL/32 COMMAND SUMMARY	A-1
B	SPL/32 COMMAND MESSAGE SUMMARY	B-1
C	SPL/32 VS. SPOOLER COMMAND CORRELATION	C-1
C.1	INTRODUCTION	C-1
D	MESSAGE HANDLING	D-1
D.1	INTRODUCTION	D-1
D.1.1	Message Overview	D-1
D.2	THE UNIFORM SPOOLER CONTROL INTERFACE (USCI)	D-1
D.2.1	Establishing Communication With the Spooler	D-2
D.2.2	Losing or Terminating Communications With Task	D-2
D.2.2.1	Normal Termination of a Control Task	D-2

APPENDIXES (Continued)

D. 2. 3	Response Routing (Spooler to Task)	D-3
D. 2. 4	Dialogue Termination	D-3
D. 3	MESSAGE HEADER	D-4
D. 4	MESSAGE FORMATS	D-8
D. 4. 1	Task to SPL/32 Message Formats	D-8
D. 4. 1. 1	The Command Message	D-8
D. 4. 1. 2	Acceptance Response Message Format	D-10
D. 4. 1. 3	Request For Mnemonic Table Message	D-12
D. 4. 2	SPL/32 to Task Message Formats	D-14
D. 4. 2. 1	Service Response Message Format	D-14
D. 4. 2. 2	Buffer Set Format	D-15
D. 4. 3	Individual Command Service Response Messages	D-20
D. 4. 3. 1	ATTACH Command Service Record	D-20
D. 4. 3. 2	BLOCK Command Service Record	D-22
D. 4. 3. 3	CHECKPOINT Command Service Record	D-24
D. 4. 3. 4	CTASK Command Service Record	D-26
D. 4. 3. 5	FORM Command Service Record	D-28
D. 4. 3. 6	GLOBAL Command Service Record	D-30
D. 4. 3. 7	GROUP Command Service Record	D-32
D. 4. 3. 8	INPUT Command Service Record	D-34
D. 4. 3. 9	INTERROGATE Command Service Record	D-36
D. 4. 3. 10	PRIMARY Command Service Record	D-42
D. 4. 3. 11	PRIORITY Command Service Record	D-44
D. 4. 3. 12	QUEUE Command Service Record	D-46
D. 4. 3. 13	SCTASK Command Service Record	D-48
D. 4. 3. 14	SECONDARY Command Service Record	D-50
D. 4. 3. 15	SPVOLUME Command Service Record	D-52
D. 4. 4	Exception Response Messages	D-54
D. 4. 5	Attention Messages	D-59
D. 4. 6	Mnemonic Message Format	D-64
D. 4. 7	Authority Message Format	D-76
D. 5	MESSAGE RECOVERY PROCEDURES	D-68
E	SPL/32 COMMANDS PREPARED FORMAT	
E. 1	INTRODUCTION	E-1
E. 2	ATTACH COMMAND	E-2
E. 3	BANNER COMMAND	E-3
E. 4	BLOCK COMMAND	E-5
E. 5	CHANGE COMMAND	E-6
E. 6	CHECKPOINT COMMAND	E-8
E. 7	CREATE COMMAND	E-9
E. 8	CTASK COMMAND	E-10

APPENDIXES (Continued)

E. 9	DELETE COMMAND	E-11
E. 10	DETACH COMMAND	E-12
E. 11	DEVICE COMMAND	E-13
E. 12	END COMMAND	E-15
E. 13	FLUSH COMMAND	E-16
E. 14	FORM COMMAND (INCLUDE)	E-18
E. 15	FORM COMMAND (REMOVE)	E-20
E. 16	FORM COMMAND (MOUNT, MOUNT VERIFY)	E-21
E. 17	GLOBAL COMMAND	E-23
E. 18	GROUP COMMAND	E-24
E. 19	INPUT COMMAND (INCLUDE)	E-25
E. 20	INPUT COMMAND (REMOVE)	E-26
E. 21	INTERROGATE COMMAND	E-27
E. 22	PRIMARY COMMAND	E-29
E. 23	PRINT COMMAND	E-30
E. 24	PRIORITY COMMAND	E-32
E. 25	PUNCH COMMAND	E-33
E. 26	QUEUE COMMAND	E-35
E. 27	SCTASK COMMAND	E-37
E. 28	SECONDARY COMMAND	E-39
E. 29	SPVOLUME COMMAND	E-40
E. 30	TRAILER COMMAND	E-41
E. 31	SUMMARY	E-43

FIGURES

1-1	The Components of SPL/32	1-4
1-2	Control and Subcontrol Tasks Communications Flow	1-6

FIGURES (Continued)

2-1	Control Statement Sequences For Single and Stacked Card Files	2-5
3-1	Evolution of a Spool File	3-3
D-1	SPL/32 Message Header Record Format	D-5
D-2	Command Message Format	D-9
D-3	Acceptance Response Message Record Format	D-11
D-4	Request for Mnemonic Table Record Format	D-13
D-5	Service Response Message - Initial Buffer Format	D-16
D-6	Service Response Message - Continuation Buffer Format	D-18
D-7	ATTACH Command Service Message Record Format	D-21
D-8	BLOCK Command Service Message Record Format	D-23
D-9	CHECKPOINT Command Service Message Record Format	D-25
D-10	CTASK Command Service Message Record Format	D-27
D-11	FORM Command Service Message Record Format	D-29
D-12	GLOBAL Command Service Message Record Format	D-31
D-13	GROUP Command Service Message - Header Record and Data Record Formats	D-33
D-14	INPUT Command Service Message Record Format	D-35
D-15	INTERROGATE Command Service Message Header and Display Record Format	D-37
D-16	INTERROGATE Configuration Command Service Message Record Format	D-40
D-17	PRIMARY Command Service Message Record Format	D-43
D-18	PRIORITY Command Service Message Record Format	D-45
D-19	QUEUE Command Service Message Record Format	D-47
D-20	SCTASK Command Service Message Record Format	D-49
D-21	SECONDARY Command Service Message Record Format	D-51
D-22	SPVOLUME Command Service Message Record Format	D-53
D-23	EXCEPTION Response Message Format	D-55
D-24	SVC1 and 7 Attention Messages Record Format	D-60
D-25	General Attention Message Format	D-63
D-26	Mnemonic Message Format	D-65
D-27	Authority Message Format	D-67

TABLES

3-1	OPTION SYMBOLS DISPLAYED BY THE INTERROGATE COMMAND	3-6
5-1	AUTHORIZATION LEVELS AND ASSOCIATED SPOOLER COMMANDS	5-9
D-1	EXCEPTION MESSAGE ERROR CODES, MESSAGE TEXT, AND MESSAGE RELATED DATA SUMMARY	D-56
D-2	ERROR TEXT FIELD ENTRIES FOR SVC1 AND 7 ATTENTION MESSAGES	D-62
D-3	GENERAL ATTENTION MESSAGE FIELD ENTRIES	D-64

TABLES (Continued)

D-4	SPL/32 ERROR STATUS AND RECOVERY INFORMATION	D-69
E-1	SUMMARY OF COMMAND IDENTIFIERS (BYTE 54) AND COMMAND FUNCTION IDENTIFIERS (BYTE 55) FOR PREPARSED COMMANDS	E-43
INDEX		Ind-1

PREFACE

This manual describes the SPL/32 spooling system for Perkin-Elmer 32-bit computer systems. This information is useful to the system administrator and applications programmer for the initialization, administration, and internal function of the SPL/32 spooling process. The commands section may prove useful to the system operator. Chapter 1 presents the spooling concept; summarizes the basic components of SPL/32, outlines the hardware and software required to run SPL/32, and details the initialization procedures for starting the spooling process. Chapter 2 explains input spooling and related commands. Chapter 3 explains the function of the spool queue, spool files, and related commands. Chapter 4 presents output spooling and its associated commands. Chapter 5 presents the concept of task interfacing, the function of the uniform spooler control interface (USCI), and the concept of capabilities based on levels of authority and associated commands. Chapter 6 details each SPL/32 command. Complete format and optional parameter lists are presented for each command. Appendix A summarizes SPL/32 commands. Appendix B presents messages in a tabular format. Appendix C presents a comparison of SPL/32 commands and existing spooler commands in a tabular format. Appendix D presents message handling internals. Appendix E presents the SPL/32 commands in preparsed format.

For information on the contents of all Perkin-Elmer 32-bit manuals, see the 32-Bit Systems User Documentation Summary.

CHAPTER 1 INTRODUCTION TO SPL/32

1.1 INTRODUCTION

The SPL/32 Spooler is a component of the 32-bit operating system and is designed to improve the handling of input/output (I/O) processes to low speed devices. The spooler also provides advanced capabilities for configuring and dynamically altering the output spooling environment. SPL/32 is available to users of the following Perkin-Elmer processors:

- Memory access controller processors
 - Model 7/32
 - Model 8/32
 - Model 3220

- Memory address translator processors
 - Model 3210
 - Model 3230
 - Model 3240
 - Model 3250

Spooling (simultaneous peripheral operations online) is a process in which I/O to low speed devices, such as card readers, printers, and card punches, is sent directly to intermediate high speed devices (disks). This process releases the central processing unit (CPU) from having to execute at the speed at which low speed I/O devices can supply or receive information, resulting in improved system throughput.

The SPL/32 Spooler provides three major advantages to a user:

1. Increased Utilization of I/O Devices

I/O devices that are dedicated to the execution of a single process are frequently idle at some point during the execution of that process. Spooling disassociates the I/O device function from the program execution function of a process, thereby releasing the I/O device for use by some other process.

2. Increased System Performance

On all spooled systems, CPU input and output operations are performed to disk, a high speed I/O device, more compatible with the CPU's speed in I/O operations. Consequently, the CPU spends less time waiting. In addition, disk I/O operations can be blocked, thereby reducing the number of I/O operations performed per process executed.

3. Reduced Device Contention

The use of a particular output device, such as a line printer, is disassociated with the actual execution of a program. This enables many programs to generate output ultimately destined for the same output device without that device being dedicated to a particular task. The spooler directs program output to the appropriate device as it becomes available.

In addition to these generalized benefits, the SPL/32 Spooler provides commands that enable the administrator of a system to tailor the spooler to best suit the requirements of that particular system. These commands enable:

- the use of an unlimited number of output devices,
- the ability to add or drop devices dynamically,
- the ability to control output devices from designated terminals anywhere within the spooling environment,
- the use of special forms-handling capabilities,
- the use of security mechanisms,
- the ability to manipulate the position of files on the print queue and also alter the output requirements of these files,
- the ability to uniquely identify output files, and
- the ability to stop devices in the spooling environment, or the spooler itself, in an orderly manner.

1.2 COMPONENTS OF THE SPOOLING PROCESS

SPL/32 is composed of a number of interrelated components that together make up the spooling process. The following sections identify and briefly describe the basic components of the SPL/32 Spooler.

1.2.1 Input Spooling

Input spooling is a process in which a batch stream of cards is copied to a disk file prior to being accessed by the CPU for processing. The user specifies the name of the disk file. Once the card deck has been copied to disk, it can then be accessed, submitted for processing, or retained for future use.

1.2.2 Output Spooling

Output spooling is a process in which program output destined for a physical output device, such as a printer or card punch, is initially copied to a disk file, known as a spool file. Once a spool file is created, the output of that spool file is controlled by the spooler, which directs the file to a specified or default (if not otherwise specified) physical output device.

1.2.3 The Spool Queue

The spool queue is a disk file maintained by the spooler that contains an entry for each file destined for output via the spooler. Each entry contains information relative to the originator and output requirements.

1.2.4 The Pseudo Device

A pseudo device establishes a correlation between a user and a physical output device. Output is assigned to pseudo devices rather than directly to physical devices. The pseudo devices are logically associated with one or more physical output devices. The spooler is responsible for directing output assigned to a pseudo device to the appropriate physical device.

1.2.5 The Spool File

A spool file is a disk file that is allocated by the spooler when an assignment to a pseudo device is made via an ASSIGN or SPOOLFILE command. I/O operations are performed using spool files.

1.2.6 The Uniform Spooler Control Interface (USCI)

The USCI serves as the software interface between the spooler and all devices/tasks that communicate with the spooler.

1.3 REVIEW

Figure 1-1 is presented to show the relationships of the six basic components of the Perkin-Elmer Spooler (SPL/32).

4031

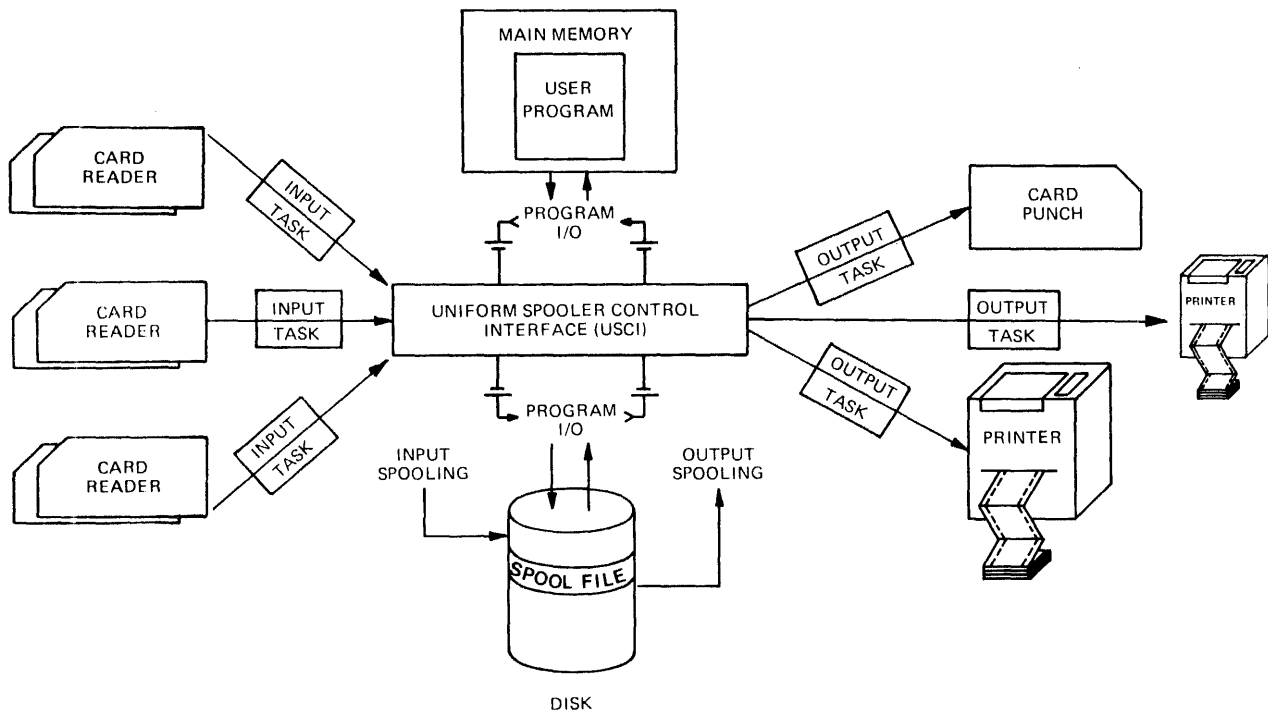


Figure 1-1 The Components of SPL/32

1.4 COMPONENTS OF SPOOLER CONTROL

The Perkin-Elmer Spooler provides a means of designating certain tasks within the spooling environment as tasks that can communicate with, and thus control, the use of the spooler. In addition, these tasks can be assigned levels of authority that further define the task's scope of control. The following sections introduce this concept.

1.4.1 Control and Subcontrol Tasks

Two types of tasks can communicate directly with the spooler:

- Control tasks
- Subcontrol tasks

Control tasks supervise spool devices via the use of spooler commands. These tasks are capable of having a direct request/direct response relationship with the spooler and are also designed to receive messages pertaining to the status of devices under their control. Control tasks can also grant tasks within their environment the privilege of issuing spooler commands directly to the spooler. These tasks are known as subcontrol tasks. Responses to commands issued from a subcontrol task can be selectively routed:

- they can be directed back to the originating subcontrol tasks, or
- they can be directed to the control task monitoring the originating subcontrol task.

This selective routing can change from request to request.

1.4.2 Levels of Authorization

The availability of the spooler commands used by control and subcontrol tasks to alter or control the spooling environment is limited through the use of levels of authorization. Each command has associated with it an authorization level identified via an authorization identifier (AID). This AID must accompany the command before the spooler will execute the command. The three levels of authorization are:

- Primary (highest)
- Secondary (selective)
- User (lowest)

The authorization level of a command reflects the command's ability to affect the spooling environment. Commands with primary authorization have the ability to alter the global spooling environment. Commands with secondary authorization have the ability to alter the environment of a logically defined group of tasks or terminals within the spooling environment. Commands with user authorization have the ability to alter the status of files placed on the spool queue by that user.

Figure 1-2 is presented to show the relationship between control and subcontrol tasks.

4058

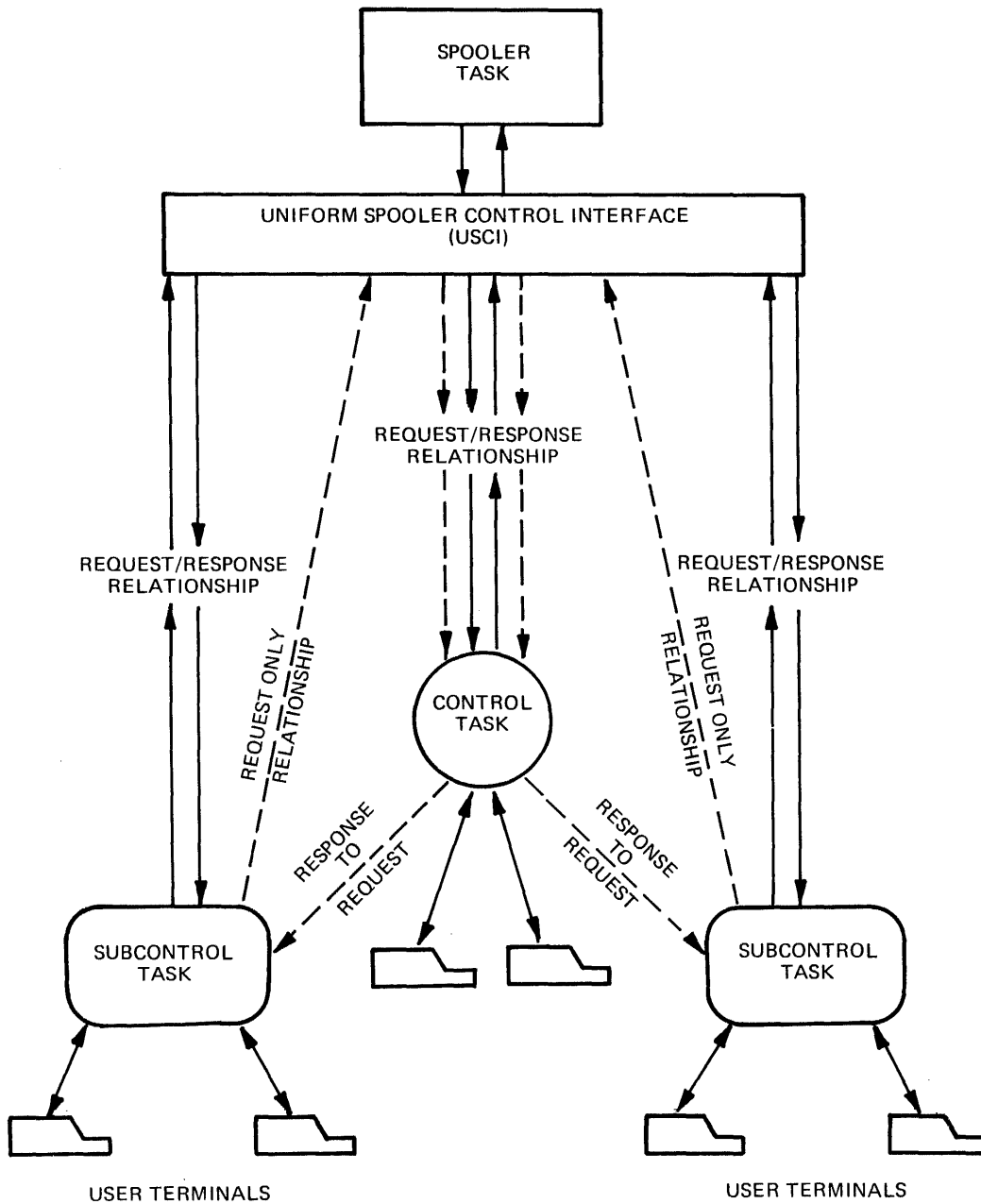


Figure 1-2 Control and Subcontrol Tasks Communications Flow

1.5 REQUIREMENTS FOR SPL/32

The following sections detail the hardware, memory, and software required to run SPL/32.

1.5.1 Hardware

The following minimum hardware requirements must be met in order to run SPL/32.

- One disk volume to store spool files and the spool queue.
- One terminal device to receive messages and enter SPL/32 commands.
- One card reader if input spooling is required.
- One printer for output of spooled print files.
- One card punch for output of spooled punch files.

SPL/32 will support peripheral devices listed in the Perkin-Elmer OS/32 System Generation (Sysgen) Reference Manual.

1.5.2 Memory Requirements

SPL/32 memory requirements are approximately 75.00kb plus:

- 32.50kb for each output device specified, and
- 25.00kb for each input device specified.

If no monitor control task is used (such as MTM), an additional 35kb are required for the SPL/32 monitor (SPLC).

1.5.3 Software Requirements

SPL/32 will run on systems operating in an OS/32 Release 6.2 or higher environment with vertical forms control (VFC) and supervisor call (SVC) intercept options as required features.

SPL/32 and the original OS/32 spooler are mutually exclusive software products. Only one spooler can be active on a machine at a given time. The spooler to be used is selected at system generation (sysgen) time. Refer to the System Generation/32 (SYSGEN/32) Reference Manual for details.

1.6 INITIALIZING SPL/32

The following sections outline the procedures involved in preparing and starting SPL/32.

1.6.1 Loading SPL/32

SPL/32 is provided in object format with the OS/32 operating system package. SPL/32 must be built as a universal task and must be loaded with the task identifier (taskid).SPLR.

Format:

```
LOAD .SPLR,SPL32,segsz increment
```

Parameters:

SPLR	is the name of the taskid for the spooler.
SPL32	is the name of the load file on disk.
segsz increment	is a decimal number in kilobytes specifying the amount of get storage area in the task's impure memory segment. If specified, this value overrides the GET and EXPAND values used when the task is linked.

Example:

```
LOAD .SPLR,SPOOLER.TSK,20
```

where 20 indicates that the spooler should be loaded into a larger segment. Complete information regarding the LOAD command is presented in the Perkin-Elmer OS/32 Operator Reference Manual.

SPL/32 may also be loaded using an SVC6 load-function option. For information regarding this procedure, see the Perkin-Elmer OS/32 Supervisor Call (SVC) Reference Manual.

1.6.2 Linking SPL/32

The following sequence of LINK commands builds the spooler task as an image load module without overlays.

```
ESTABLISH TASK
OPTION,WORK=X4000,SYS=XFFFFFF,TEQSAVE=NO,UNIV,COM,CON,ACP,
LU=254,INTERCEPT
INCLUDE SPL
MAP PR:,ADDR,ALPHA
LIB PNEWIX
LIB PASRTL
BUILD SPL
END
```

For LINK instructions to build the spooler task with overlays, see the packaging document.

1.6.3 Starting the Spooler

Creating the desired initial spooler environment is accomplished through the use of the START command.

Format:

```
START [,COMMAND = fd]
```

Parameters:

fd	specifies a file descriptor for the location of the commands to configure the spooling environment.
----	---

Functional Details:

These commands may be in a disk file, card deck, magnetic tape, or may be entered from an interactive device. Any combination of valid SPL/32 commands (as listed in Chapter 6) may be used in the command file.

Examples:

START, COMMAND = M300:PARAM.DLK/118

File PARAM.DLK in account 118 on volume M300: contains the SPL/32 startup parameters.

START, COMMAND = CRD:

The card deck in card reader CRD: contains the SPL/32 startup parameters.

START, COMMAND = MAG1:

The tape mounted on tape drive MAG1: contains the SPL/32 startup parameters.

START, COMMAND = CON:

If the optional parameter [COMMAND = fd] is not included in the START command, the system console becomes, by default, the device where the initial SPL/32 commands for configuring the spooler environment will be entered.

1.6.3.1 Creating a START Command File

A START command file is simply a sequential text file that contains spooler commands. The commands in Chapter 6 are those available for use in the START command file. Commands must be presented one per line with the correct syntax. The end of a START command file is signified by the following for the media listed:

DEVICE	END ID
Cards	/* on the final card
Disk file	end of file (EOF) condition, or /*
Mag tape	EOF mark, or /*
Interactive device	Two consecutive carriage returns, or /*

A sample START command file follows:

```
SPVOLUME    INCLUDE M300, M301, M302
QUEUE       M301:SPLQUEUE.NEW
BLOCK       2/3
PRIORITY    220
INPUT       INCLUDE CRD1:, CRD2:, DRD3:
CREATE      pd1:, pd2:, pd3:
GLOBAL      pd1:
ATTACH      PDEVICE = pd1: RDEVICE = prt1:
ATTACH      PDEVICE = pd2: RDEVICE = prt2:, prt5:
ATTACH      PDEVICE = pd3: RDEVICE = prt9:
GROUP       PDEVICE = pd1:, INCLUDE CRT1:, CRT2:, CRT3:
GROUP       PDEVICE = pd2:, INCLUDE CRT7:, CRT8:, CRT9:
BANNER      prt1:, SINGLE
BANNER      prt9:, NONE
BANNER      prt2:, DOUBLE
FORM        INCLUDE REPORTS, CHECKS, TIMECARD
/*
```

By passing the name of the example file in a START command, the spooler environment will be configured according to the commands present in the command file.

Readers should familiarize themselves with the function of the various spooler commands before creating the START command file.

It is also possible to start the spooler without specifying a START command file. In this case, the spooling environment will be established using the default conditions for each environmental parameter. The spooling environment can then be altered dynamically as the need arises, without interrupting spooler performance.

1.6.4 SPL/32 System Console Interface

There are two general conventions available for interfacing with SPL/32 at the system console.

The initial procedure involves selecting the SPL/32 task as the current task at the system console before entering any spooler commands. Then spooler commands can be entered as long as they are preceded by a SEND instruction.

Example:

```
*TASK .SPL
*SEND INTERROGATE QUEUE } any SPL/32 command
*SEND CTASK
```

The second convention involves prefixing each SPL/32 command with the .SPL prefix or the .MTM prefix, depending upon which monitor control task is being used. Initially, .SPL should be used.

Example:

```
*.SPL INTERROGATE QUEUE
```

When the monitor task for the spooler is to be MTM, the following command sequence establishes MTM as the primary control task.

Example:

```
*LO .MTM,MTM  
*.SPL CTASK INCLUDE .MTM  
*.SPL PRIMARY .MTM/CON
```

Each subsequent SPL/32 command must be preceded by the .MTM prefix.

Example:

```
*.MTM INTERROGATE QUEUE
```

1.6.5 Ending the Spooler Task

The spooler can be stopped via the END command. This command provides the option of an immediate or delayed stop.

Examples:

```
END IMMEDIATE
```

All spooler functions cease. All pseudo devices and physical devices are removed.

```
END DELAYED
```

All spooler functions cease after the currently printing spool files are output. New assignments to the spooler are rejected. Section 6.22 further defines the END command.

CHAPTER 2 INPUT SPOOLING

2.1 INTRODUCTION

The SPL/32 input spooling feature provides facilities for copying a batch stream of cards such as source programs, operator commands, command substitution system (CSS) files, or other data to disk files for subsequent processing. This procedure is performed without user task intervention. Each card deck can be spooled to a user-specified file descriptor (fd), and the file can be submitted to the Perkin-Elmer Multi-Terminal Monitor (MTM) for batch processing. The following sections describe how the input spooling feature is established, functions, and is stopped.

2.2 ESTABLISHING INPUT SPOOLING

The input spooler task is a non-resident task that is implicitly loaded into memory when a card reader is added to the spooler environment. There is one input task for each card reader included.

Card readers are designated as spooled devices via the INCLUDE parameter of the INPUT command. See Section 6.15. Spooled card readers are exclusively assigned to the spooler function and cannot be used by any other task unless it is through the spooler.

2.3 INPUT SPOOLING CONTROL STATEMENTS

Each batch of cards to be spooled to disk must be preceded by a control card statement. This statement specifies the fd to which the input data is to be spooled. SPL/32 provides two such control statements:

- /@INPUT
- /@SUBMIT

2.3.1 The /@INPUT Control Statement

The /@INPUT control statement is used to copy a card file to a specified fd on disk.

Format:

`/@INPUT fd[,DELETE]`

Parameters:

`fd` specifies the file descriptor of the disk file in the form `voln:filename.ext/acctno.` An index file with a logical record length of 80 bytes and a blocksize of one 256 byte sector each for both data and index blocks is allocated with name `fd`.

`DELETE` if the file currently exists, it is deleted and reallocated when this parameter is specified.

Functional Details:

The default blocksize for data and index blocks can be changed via the `SPL/32 BLOCK` command. (See Section 6.4.) The blocksizes used by the `/@INPUT` command will be the default values in effect at the time the card reader is included in the spooler environment.

If a file currently exists and the `DELETE` option is omitted, the spooler will issue the following error message:

```
SVC7 ERROR VOLN:FILENAME.EXT
```

If a control statement card is syntactically incorrect, `SPL/32` issues one of the following messages based on the nature of the error:

```
MISSING PARAMETER: text
INVALID PARAMETER: text
INVALID FILENAME: text
INVALID KEYWORD: text
INVALID SEPARATOR: text
INVALID CONTROL CARD: text
```

The deck of cards containing the problem is read (the information is not copied to disk) until a new control statement is read. This process allows stacked input to be processed even if an intervening card deck contains an error. If a card reader returns a status X'A0' before an ending control statement is read, the spooler retries the read at timed intervals until more cards are placed in the reader. All other spooler functions, such as printing and punching to or reading from other spool devices, continue. If the volume name is omitted from the fd in a /@INPUT control statement, the file is allocated on one of the spool volumes specified in the SPVOLUME command.

2.3.2 The /@SUBMIT Control Statement

The /@SUBMIT control statement is used in an MTM environment to copy a card batch to a specified fd and to submit the fd to MTM for batch processing.

Format:

```
/@SUBMIT fd[,DELETE]
```

Parameters:

fd	specifies the file descriptor. An index file with a logical record length of 80 bytes and a blocksize of one 256 byte sector each for both data and index blocks is allocated with name fd.
DELETE	if the file currently exists, it is deleted and reallocated when this parameter is specified.

Functional Details:

The default blocksize for data and index blocks can be changed via the SPL/32 BLOCK command. (See Section 6.4.) The blocksizes used by the /@INPUT command will be the default values in effect when the card reader is included in the spooling environment.

If a file currently exists and the DELETE option is omitted, the spooler will issue the following error message:

```
SVC7 ERROR VOLN:FILENAME.EXT
```

If a control statement card is syntactically incorrect, SPL/32 issues one of the following messages based on the nature of the error:

MISSING PARAMETER: text
INVALID PARAMETER: text
INVALID FILENAME: text
INVALID KEYWORD: text
INVALID SEPARATOR: text
INVALID CONTROL CARD: text

The file containing the error is read (the information is not copied to disk) until a new control statement is read. The process of reading the remaining card files allows stacked input to be spooled even if an intervening card file contains an error.

If a card reader returns a status X'A0' before an ending control statement is read, the spooler assumes that some of the card deck is missing and retries the read at timed intervals until more cards are placed in the reader. All other spooling functions, such as printing and punching to or reading from spooled devices, continue.

If the volume name is omitted from the fd in a /@SUBMIT control statement, the file is allocated on one of the spool volumes specified in the SPVOLUME command.

2.3.3 Ending Input Spooling of a Card File

The following control statement denotes the end of an input card file:

/e

These characters must be entered in columns 1 and 2 with blanks in all remaining columns. This control statement should be located on the last card in the file.

When a number of card files are consecutively stacked in a single card reader, the halt control statement notifies the spooler of the end of one card file. The following card must contain the /@INPUT or /@SUBMIT control statement for the next card file to begin spooling. If the halt control statement is omitted, a new file will be allocated for the next card file, spooling will continue for that file, and the following message will be sent to the monitor control task:

MISSING CONTROL CARD: FILEVOLN:FILENAME.EXT

Figure 2-1 details the proper sequences for single and stacked card files.

4059

SINGLE CARD FILE

STACKED CARD FILES

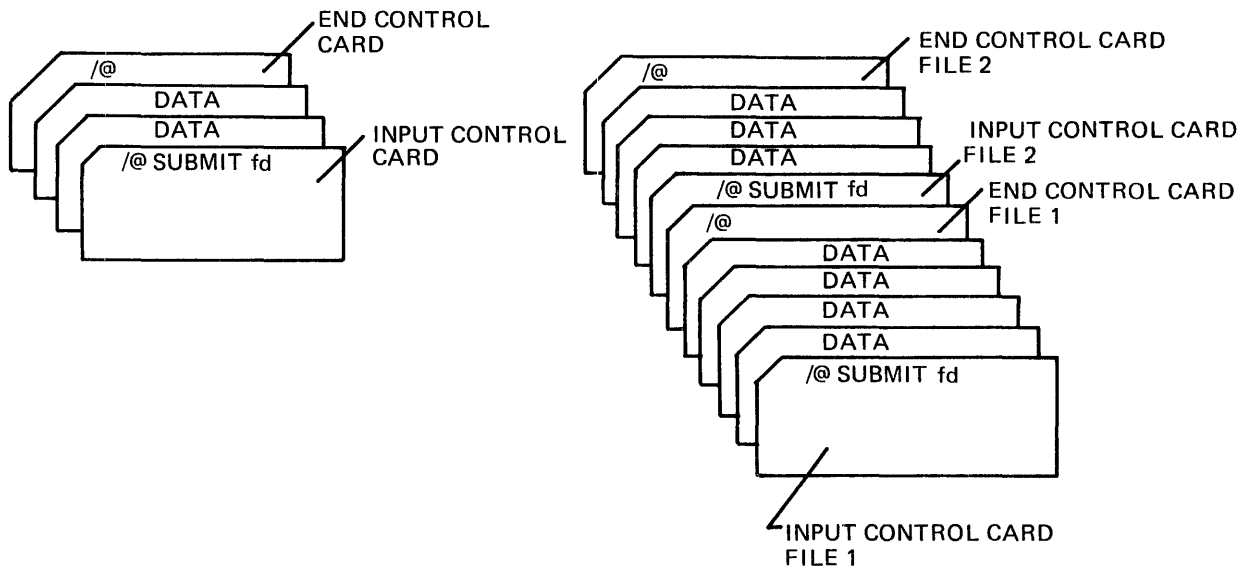


Figure 2-1 Control Statement Sequences For Single and Stacked Card Files

2.4 REMOVING INPUT SPOOLING DEVICES

Card readers are removed from the spooling environment via the REMOVE parameter of the INPUT command. SPL/32 allocates one input task for each card reader in the spooling environment. When a card reader is removed from the spooling environment, its associated input task is cancelled.

CHAPTER 3 THE SPOOL FILE AND THE SPOOL QUEUE

3.1 INTRODUCTION

A spool file is a disk file created by SPL/32 to receive output data that is subsequently to be sent to a physical output device. Spool files are allocated by the spooler in response to assignments to pseudo devices via an ASSIGN or SPOOLFILE command.

The spool queue is a disk file that contains status information regarding each file to be output via the spooler. These may be spool files (files allocated by the spooler) or user files (already existent files) temporarily assigned to the spooler for output via a PRINT, PUNCH, or CHANGE command. The following sections explain the structure, function, and use of these two components of SPL/32.

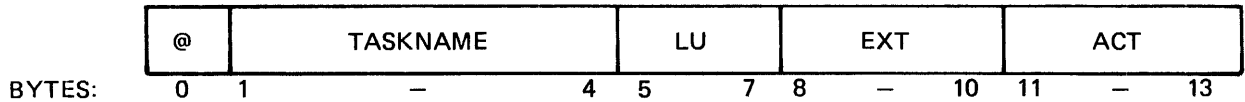
3.2 THE SPOOL FILE

In a spooled environment, output data assigned to a pseudo device via an ASSIGN or SPOOLFILE command is written to a disk file called a spool file. Each time an assignment is made to a pseudo device, OS/32 intercepts the supervisor call 7 (SVC7) generated and routes it to SPL/32. SPL/32 translates the assign and allocates a spool file. Spool files are allocated on multiple spool volumes (if present) in a rotating fashion so that no volume receives a disproportionate number of spool files. The index and data blocksize of spool files can be specified using the BLOCK command. The default value for both data and index blocksize is one 256 byte sector each. When changing the default blocksize, the value specified cannot exceed the values specified at system generation (sysgen) time. The record length of spool files is determined by the spooler when an assignment is made to a pseudo device, the spooler reads the device attributes of the physical device associated with the pseudo device. Spool files allocated as a result of the assignment to that pseudo device have the record length of the physical device. If multiple physical devices are associated with a single pseudo device, the record length of the device with the largest record length is used. Therefore, similar output devices should be grouped under one pseudo device. If a pseudo device has not been associated with any physical devices, the spooler assigns a default record length of 137 bytes.

3.2.1 Naming Spool Files

SPL/32 uses the following format when naming spool files:

3622



Fields:

- @ is the 1-character at sign that precedes the filename and identifies the file as a spool file.
- TASKNAME is the first four characters of the taskname on whose behalf the assignment is being made.
- LU is a 3-digit decimal number representing the logical unit being assigned.
- EXT is a 3-character extension used to distinguish between spool files if naming conflicts arise in the first eight characters of the spool file name. This field can contain any combination of the letters A through Z and the hexadecimal numbers 0 through F.
- ACT is a 1- to 3-digit decimal number specifying the user account from which the assignment originated.

Example:

@TEST002.000/119

The preceding example designates a spool file with the following information:

- assigned from a task called TESTRUN,
- assigned lu 002,
- has the extension 000, and
- originated from account number 119.

When the lu assigned to a spool file is closed, the SVC7 is intercepted by OS/32 and routed to the spooler, which places the spool file name on the spool queue. Spool files are placed on the queue in order of task priority and are subsequently output in that order depending on the availability of the required output device. Figure 3-1 illustrates the steps followed in the evolution of a spool file.

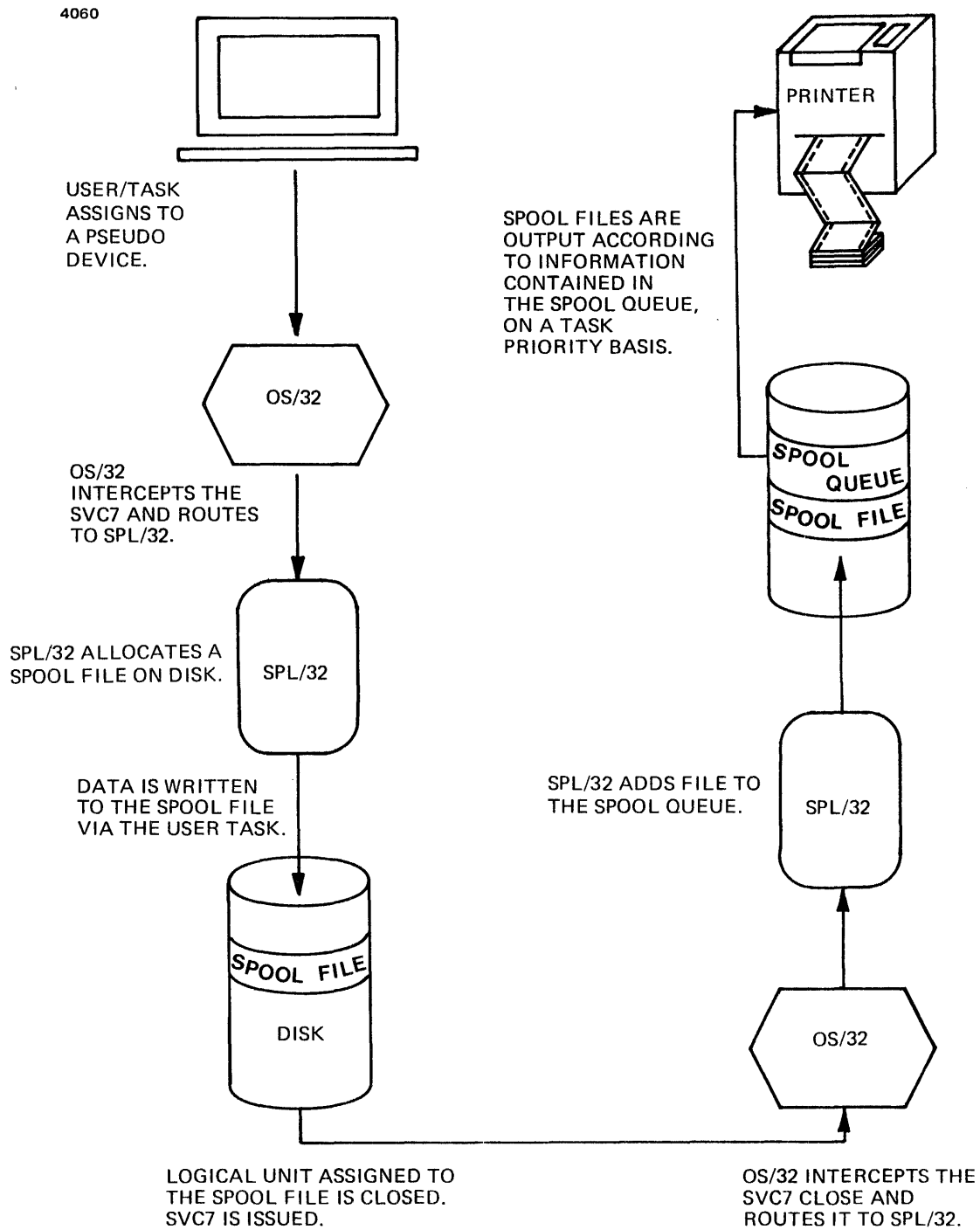


Figure 3-1 Evolution of a Spool File

3.3 THE SPOOL QUEUE

The spool queue is a disk file that contains an entry for each spool file on the system. Each entry contains origin and output requirements for each spool file.

3.3.1 Creating the Spool Queue

The spool queue is created at SPL/32 startup time in one of two ways:

1. The spool queue can be explicitly created, named, and allocated on a specified disk volume using the QUEUE command.

Example:

```
QUEUE M67B:SPLQUEUE.NEW
```

SPLQUEUE.NEW is allocated as the spool queue on disk volume M67B. If the disk volume is omitted, the spool queue is allocated on the first volume specified in a previous SPVOLUME command, or on the default system volume if no SPVOLUME command was previously entered. If no spool volumes were previously specified, and the system volume cannot be written to, the TEMP volume must be available or else the spool queue cannot be allocated.

2. The spool queue is implicitly created by the spooler if the QUEUE command is not used. The default spool queue, SPL32.QUE, is allocated on the first volume specified in a previous SPVOLUME command or, by default, on the system volume if no previous SPVOLUME command was entered. If no spool volumes were previously specified, and the system volume cannot be written to, the TEMP volume must be available or else the spool queue cannot be allocated.

The spool queue is a dynamic file with a default block size of 1 and a record length of 136.

3.3.2 Displaying the Spool Queue File Name

The spool queue currently in use can be determined by entering the QUEUE command without parameters. The following example represents this with a resultant display.

Example:

```
* QUEUE  
*  
* QUEUEFILE = M301:SPL32.QUE/0
```

3.3.3 Respecifying the Spool Queue

If a system failure occurs while the spooler is active or the spooler task is cancelled, spool file entries listed on the spool queue at the time of the failure will be held. After the system is restarted and the spool queue name is specified, spooling resumes with the file that was being output at the time of the interruption. If no file was being output, spooling begins with the next entry on the queue. If the default spool queue, SPL32.QUE, was in use prior to the failure, it is not necessary to respecify the spool queue because the spooler will automatically reassign the default spool queue.

3.3.4 Respecifying the Spool Queue While the Spooler is Active

SPL/32 allows the user to respecify the spool queue while the spooler is active. This is accomplished via the QUEUE command. When a new spool queue is specified, those entries on the old queue that were being output at the time of the interruption are copied by SPL/32 into the new spool queue. Spooling then resumes with these entries.

3.3.5 Interrogating the Spool Queue

Entries on the spool queue and the status and configuration of the spooling environment can be displayed using the INTERROGATE command. This information can be selectively displayed according to the type of information desired. The following five sections detail the optional displays available.

3.3.5.1 Displaying Entries in the Spool Queue

The QUEUE parameter of the INTERROGATE command displays all entries in the spool queue relative to the user's authorization level.

Example:

```
INTERROGATE QUEUE
```

Resultant Display:

PSEUDO DEVICE	FILE NAME	FORMNAME	COP	PRI	OPTIONS
PD1:	M67B:@COPY003.000/118	STD	10	128	H I D
PD2:	M300:@TEST005.000/119	CHECKS	95	129	H I
PD3:	M300:@TEST004.001/120	STD	1	130	R I D
PD4:	M67B:TEST.PRT/100	FORMA	12	128	R I

The letters (or blanks) displayed in the options columns represent options specified in the PRINT/PUNCH request when each file was placed on the queue. These option symbols (or blanks) are explained in Table 3-1.

Note that the NOIMAGE, NODELETE, and NOVFC options are represented in the display by blanks in columns 2, 3, and 4 under the options header.

TABLE 3-1 OPTION SYMBOLS DISPLAYED BY THE INTERROGATE COMMAND

OPTION SYMBOL	DEFINITION	CHARACTER POSITION IN OPTION COLUMN	RESULT
H	Hold	1	Output request is held on the queue until released.
R	Release		Allows a held output request to become eligible for output.
I	Image	2	File is to be output as is.
	Noimage		File is to be output in formatted mode.
D	Delete	3	File is deleted after output.
	Nodelete		File is not deleted after output.
V	VFC	4	File is to be output with vertical forms control (VFC).
	NOVFC		File is to be output without VFC.

3.3.5.2 Displaying Queue Entries According to Preprinted Form Requests

The FORM parameter of the INTERROGATE command causes all entries on the spool queue that require the specified form to be displayed.

Example:

```
INTERROGATE FORM CHECKS
```

Resultant Display:

PSEUDO DEVICE	FILE NAME	FORMNAME	COP	PRI	OPTIONS
PD2:	M300:@TEST005.000/119	CHECKS	2	128	H I D
PD5:	M300:@TEST006.000/119	CHECKS	90	129	H I
PD6:	M67B:@RUNT008.000/118	CHECKS	120	129	H I D

This example displays information about all spool queue entries that requested the special form CHECKS. The characters in the option column are defined in Table 3-1.

3.3.5.3 Displaying Files by Assigned Pseudo Device

The PD: parameter of the INTERROGATE command displays all spool files on the queue that are assigned to the specified pseudo device.

Example:

```
INTERROGATE PD PD1:
```

Resultant Display:

PSEUDO DEVICE	FILE NAME	FORMNAME	COP	PRI	OPTIONS
PD1:	M300:@PROG001.000/118	STD	5	128	H I D
PD1:	M300:@PROG002.001/118	STD	6	128	R I
PD1:	M300:@PROG002.000/118	STD	1	128	R I D

The preceding example displays all spool files that are assigned to pseudo device PD1:. The characters in the OPTIONS column are defined in Table 3-1.

3.3.5.4 Displaying Information About a Particular File

The FD parameter of the INTERROGATE command displays information about a specified file.

Example:

```
INTERROGATE FD DATAFILE
```

Resultant Display:

PSEUDO DEVICE	FILE NAME	FORMNAME	COP	PRI	OPTIONS
PD2:	M300:@DATA001.001/119	STD	1	128	H D V

This example displays information about the specified file DATAFILE. The characters in the options column are defined in Table 3-1.

3.3.5.5 Displaying the Spooler Environment Configuration

The CONFIGURATION parameter of the INTERROGATE command displays the status of all physical devices in the spooler environment.

Example:

```
INTERROGATE CONFIGURATION
```


Resultant Display:

REAL DEVICE	DEVICE STATUS	FORM MOUNTED	ACTIVE FILE	ASSOCIATED PSEUDO DEVICES
PRT1:	I	STD	NONE	PD1:, PD2:, PD3:
PRT2:	A	CHECKS	M67B:@DAVI003.001/119	PD6:
PRT3:	P	STD	M67B:TEST.PRT/100	PD1:, PD7:

LEGEND

I = Inactive

A = Active

P = Paused

This example displays all physical devices in the spooler environment, the status of the device, the file (if any) currently printing on the device, the form mounted on the device, and the pseudo devices associated with the physical device.

3.3.5.6 Displaying All Spool Queue and Spooler Configuration Information

If the INTERROGATE command is entered with no parameters, all queue information and the spooling environment configuration are displayed. The queue information displayed is dependent upon the requestor's authorization level.

Example:

INTERROGATE

Resultant Display:

PSEUDO DEVICE	FILE NAME	FORMNAME	COP	PRI	OPTIONS
PD1:	M67B:@COPY003.000/119	STD	2	128	H D V
PD2:	M300:@TEST004.001/120	STD	10	129	H I

REAL DEVICE	DEVICE STATUS	FORM MOUNTED	ACTIVE FILE	ASSOCIATED PSEUDO DEVICES
PRT1:	I	STD	NONE	PD1:
PRT2:	A	STD	M67B:@COPY003.000/119	PD1:
PRT3:	P	STD	M300:@TEST004.000/120	PD2:

LEGEND

I = Inactive

A = Active

P = Paused

3.3.5.7 Functional Discussion of the INTERROGATE Command

Information displayed in response to an INTERROGATE command is dependent upon the authorization level of the device issuing the request. If a request is issued with primary authorization, the information displayed pertains to all entries currently on the spool queue. If the request is issued with secondary authorization, the information displayed pertains to the spool queue entries placed on the queue by members of the group that the secondary level of authorization applies to. If the request is issued with user authorization, only the spool queue entries placed on the queue by that user are displayed.

CHAPTER 4 OUTPUT SPOOLING

4.1 INTRODUCTION

The SPL/32 output spooling feature provides a variety of functions designed to handle the processing of output files. The printing or punching of a spool file is handled by the output spooler. The management of files on the spool queue waiting to be output is performed by the uniform spooler control interface (USCI). This interface is discussed in Appendix D.

The configuration of the output spooling environment can be tailored to optimize the benefits of spooling for each particular system. This can be done when the spooler is started or dynamically while the spooler is active.

The following features are provided as a function of output spooling:

- The number of output devices is dependent only on the amount of available memory.
- Capability of retaining a spooled output file after it is sent to a device.
- Capability of holding spooled files from output processing.
- The option to backspace, forward space, or rewind a file that is currently being output by the spooler, and then resume output.
- The option to produce up to 255 copies of an output file.
- The option to print informative header and trailer pages to identify output files.
- The capability of using preprinted forms and testing for form alignment before output.
- The capability to alter the output requirements of a file waiting to be output.
- The capability to alter the order in which files are output.
- The capability to control devices within the output spooling environment.
- The capability to quiesce the entire output spooling function or individual devices in an orderly fashion.

4.2 THE PSEUDO DEVICE

A pseudo device is a symbolic device reference whose function is to establish a correlation between a user/task and a physical output device. Within the output spooling environment, output files are directed to pseudo devices rather than directly to physical output devices. This provides two general benefits:

- The routing of output data is not limited to a single physical output device. A pseudo device can be associated with many physical output devices and output can be directed by the spooler to the first available of these many physical devices.
- The configuration and relationships between pseudo and physical devices can be altered while the spooler is active.

When a task or user assigns a file for output to a pseudo device, the spooler translates the assign as a request to use a physical output device associated with the pseudo device.

Pseudo devices can be created and deleted while the spooler is active and can be logically associated with one or more physical output devices. When a physical device is associated with a pseudo device, it becomes assigned exclusively to the spooler and cannot be used by any other task except through the spooler. Pseudo devices are represented by 1- to 4-character alphanumeric identifiers; the first character must be alphabetic, the remaining may be alphanumeric.

The following sections detail the process of creating, configuring, and using pseudo devices during output spooling.

4.2.1 The Global Pseudo Device

The global pseudo device is the default system pseudo device. If a pseudo device is not specified in a PRINT or PUNCH request, the file will automatically be assigned to the global pseudo device. This enables a user to request the output of a file even though the user is not aware of the specific pseudo device names present in the spooling environment or is not concerned with having a file output on a specific device.

When the spooler is started, a global pseudo device called PR: is the default pseudo device for the output spooling environment. All requests to output files that do not explicitly specify a pseudo device are assigned to the global pseudo device, which is associated with all physical output devices in the spooling environment. The global pseudo device name can be changed via the GLOBAL command. Primary authorization is required to change the global pseudo device.

Example:

```
GLOBAL PR9:
```

The global pseudo device name is changed from PR: to PR9:.

4.2.2 Creating Pseudo Devices In the Spooling Environment

The initial spooling environment will contain only the global pseudo device unless other pseudo devices are introduced into the environment. Pseudo devices can be created in the spooling environment in two ways:

- explicitly, via the CREATE command, or
- implicitly, via the ATTACH command.

The CREATE command is used to specify new pseudo devices in the spooling environment.

Example:

```
CREATE PD1:,PD2:,PD3:
```

PD1:, PD2:, and PD3: are to be recognized as valid pseudo device names. Up to 20 pseudo devices can be specified in a single CREATE command.

If the pseudo device name specified in an ATTACH command does not exist in the spooling environment, SPL/32 will create it and the following message is displayed:

```
IMPLICITLY CREATED PSEUDO DEVICE: pd:
```

Pseudo devices can be created when the spooler is started by including the CREATE command in the start command file, or they can be created while the spooler is active. Once a pseudo device has been created, it must then be associated with physical output devices. If a pseudo device is not associated with a physical output device, spool files assigned to that pseudo device will remain on the spool queue and will not be output.

4.2.3 Associating Pseudo Devices with Physical Devices

The process of associating a pseudo device with physical output devices is accomplished via the ATTACH command.

Example:

```
ATTACH PDEVICE = PD1: RDEVICE = PRT1:
```

The pseudo device PD1: is logically associated with the physical print device PRT1:. If pseudo device PD1: does not exist, SPL/32 will create it and then notify the initiator, as detailed in the previous section.

Multiple physical devices can also be associated with a single pseudo device.

Example:

```
ATTACH PDEVICE = PD1: RDEVICE = PRT2:,PRT3:,PRT4:
```

The pseudo device PD1: is logically associated with the physical print devices PRT2:, PRT3:, and PRT4:. When a pseudo device is associated with multiple physical devices, the output files assigned to that pseudo device can be sent to any one of the multiple physical devices based on immediate availability.

When a pseudo device is associated with multiple physical devices, the device attributes for the pseudo device are defined by the last physical device attached to the pseudo device. The logical record length for the pseudo device is determined by the physical device with the largest logical record length associated with the pseudo device via an ATTACH command. Therefore, insure that the physical devices associated with a single pseudo device have like device attributes.

NOTE

If a pseudo device is not associated with any physical output device, output assigned to that pseudo device will not be output and will remain on the spool queue until the pseudo device is associated with a physical device or the files are flushed.

4.2.4 Directing Output to a Pseudo Device

Users can direct output to a pseudo device by using one of the following spooler commands:

- PRINT
- PUNCH
- CHANGE

Examples:

```
PRINT TESTRUN.CBL/120,DEVICE=PD3:
```

```
PUNCH CALC32.CAL/119,DEVICE=PD4:
```

File TESTRUN.CBL in account 120 is directed to pseudo device PD3: for output. File CALC32.CAL in account 119 is directed to pseudo device PD4: for output. The CHANGE command can be used to change the pseudo device specification of a previous PRINT or PUNCH command while the file is on the spool queue.

Example:

```
CHANGE CALC32.CAL/119,DEVICE=PD3:
```

File CALC32.CAL in account 119, previously directed to pseudo device PD4: is now directed to pseudo device PD3:.

A user can specify the following optional output parameters when directing output to a pseudo device via the PRINT or PUNCH commands or can change these parameters via the CHANGE command:

- The number of copies of a file to be produced
- Whether a preprinted form is to be used
- The output priority of a file
- The output format of the file (IMAGE, VFC, NOIMAGE, NOVFC)
- Whether the file is to be deleted after output
- Whether the checkpointing feature is to be used
- Whether a file is to be held on the spool queue

4.2.5 Limiting Default Access to a Pseudo Device

Although files assigned to the global pseudo device can be output to any of the physical output devices present in the spooling environment, this type of access can be restricted for specified pseudo devices and consequently, the physical devices associated with them. This restriction is accomplished via the RESTRICT parameter of the DEVICE command.

Example:

```
DEVICE PD1:,RESTRICT
```

Pseudo device PD1: now has restricted access. This means only explicit assignments to PD1: in a PRINT, PUNCH, or CHANGE request will be accepted for output on PD1:. Assignments to the global pseudo device will not be output to the physical devices associated with PD1:. Removing the restricted access condition from a pseudo device is accomplished via the FREE parameter of the DEVICE command.

Example:

```
DEVICE PD1:,FREE
```

Pseudo device PD1: has no access restrictions. A file assigned to the global pseudo device can eventually be output on one of the physical devices associated with pseudo device PD1:.

4.2.6 Establishing Unique Default Print Locations

Unique default print locations can be established for a specified set of tasks or terminals. This is particularly useful for grouping units that perform a common function, such as word processing terminals and a high quality printer. The printer is established as the default print location for those terminals. Unique default print locations are established via the GROUP command.

Example:

```
GROUP PDEVICE=PD7: INCLUDE CRT1: ,CRT2: ,CRT3:
```


A group of terminals (CRT1:, CRT2:, CRT3:) are logically associated with the pseudo device PD7:. By associating PD7: with physical device PRT7:, output assigned to the global pseudo device from any of those three terminals is sent to PD7: and subsequently printed on PRT7:.

The default print location for a group differs from the global default print location; however, the manner in which a user assigns to either default location is the same.

A task or terminal can be associated with one default print location only, either the global default print location or a group default print location. A task or terminal is removed from a group via the REMOVE option of the GROUP command.

Example:

```
GROUP PDEVICE=PD2: REMOVE CRT2:
```

CRT2: is no longer associated with a group and will now use the global pseudo device to determine a default print location.

4.2.7 Disassociating Pseudo Devices From Physical Devices

The association between a pseudo device and a physical device (or devices) is broken via the DETACH command.

Example:

```
DETACH PDEVICE=PD2: RDEVICE=PRT4:
```

The physical device PRT4: is detached from pseudo device PD2:. If the physical device specified is not associated with any other pseudo device, the logical unit (lu) assigned to that physical device is closed. However, if it is associated with another pseudo device, that association remains unaffected. Up to 19 physical devices can be specified in a single DETACH command.

4.2.8 Deleting Pseudo Devices From the Spooling Environment

Pseudo devices are deleted from the spooling environment via the DELETE command.

Example:

```
DELETE PD1:,PD2:
```

Pseudo devices PD1: and PD2: are deleted from the spooling environment.

When a pseudo device is deleted, its association with any physical device is also deleted. Subsequent assignments to the deleted pseudo device are rejected; however, assignments already on the spool queue are allowed to complete.

4.3 PHYSICAL OUTPUT DEVICE CONTROL

Physical output devices are introduced in the spooling environment when they are associated with pseudo devices in the ATTACH command as previously shown in Section 4.2.3. The following sections detail how these devices can be controlled.

4.3.1 Altering the Status of Physical Output Devices

The status of physical output devices can be altered via the DEVICE command, which provides the system operator with the ability to:

- suspend a physical output device (delayed or immediately),
- continue a physical output device that was suspended,
- reposition a file being output on a specific physical device in four ways:
 1. The file may be forward spaced a specified number of lines.
 2. The file may be backspaced a specified number of lines.
 3. The file may be forward spaced a specified number of pages.
 4. The file may be backspaced a specified number of pages.
- Rewind the file being output on a physical device to the beginning of the file and resume output.

Examples of the use of the DEVICE command to perform these functions follow.

Example:

```
DEVICE PRT1:, SUSPEND IMMEDIATE
```

```
DEVICE PRT2:, SUSPEND DELAYED
```

The printer PRT1: is suspended immediately; printer PRT2: will be suspended when the currently printing file has completed.

Example:

```
DEVICE PRT1:, CONTINUE
```

This example demonstrates how to continue a physical device (PRT1:) that was previously paused. The file that was being printed on PRT1: when the PAUSE command was issued will resume output starting with the next record.

Example:

```
DEVICE PRT1:, SUSPEND IMMEDIATE
```

```
DEVICE PRT1:, FLINE = 25
```

This example suspends printer PRT1:, forward spaces the file being output on device PRT1: 25 lines, and resumes printing the file. The entry of a position parameter in the DEVICE command, such as FLINE, implicitly causes a suspended device to resume output. It is not necessary to enter a CONTINUE parameter. Section 6.10 provides complete details on the use of the DEVICE command.

4.3.2 Removing Files From The Spool Queue

Files can be removed from the spool queue via the FLUSH command.

Example:

```
FLUSH TEXTFILE.STD/119
```

The specified file (TEXTFILE.STD/119) is removed from the spool queue.

Files can also be removed from the spool queue and also deleted from disk in the same operation via the DELETE option of the FLUSH command.

Example:

```
FLUSH TEXTFILE.NEW/118,DELETE
```

The specified file (TEXTFILE.NEW) is removed from the spool queue and deleted from disk (if it is not currently being printed). If the file is currently being printed, it will not be deleted from disk.

A file can also be deleted from the physical device on which the file is being output.

Example:

FLUSH PRT1:

I/O to the specified device PRT1: is immediately terminated and the file being output on the specified device is deleted from the spool queue. This option is useful for terminating I/O to a malfunctioning printer or a remote printer without knowing the name of the file being output on the printer.

4.3.3 Removing Physical Devices From the Spooling Environment

Physical devices are removed from the spooling environment via the DELETE command.

Example:

DELETE PRT7:

Physical device PRT7: is deleted from the spooling environment.

When a physical device is deleted, its association with any pseudo device is also deleted. If a file is being output on the device when it is deleted, the output is allowed to finish.

4.4 USING PREPRINTED FORMS

Another feature of SPL/32 is the ability to handle and manipulate special forms in the spooling environment. SPL/32 provides facilities to:

- name forms,
- characterize the format of a form,
- associate forms with specific printers, and
- verify alignment of a mounted form prior to use.

The following sections describe how these facilities can be used.

4.4.1 Naming and Characterizing Preprinted Forms

In order to use preprinted forms in the spooling environment, the forms must first be identified to the spooler. The physical print area of the form can be characterized, or the default print area values can be used. This is accomplished via the INCLUDE parameter of the FORM command.

Example:

```
FORM INCLUDE REPORT1 (PLENGTH=60 BMARGIN=10 TMARGIN=5 PWIDTH=132)
```

The form REPORT1 is included as a valid form name and is given a page length of 60 lines, a bottom margin of 10 lines, a top margin of 5 lines, and a page width of 132 columns.

Up to four forms may be included in a single FORM command. The spooler has a predefined form name standard (STD) that represents the paper in standard use. STD is the default print form, and at spooler start time it is assumed that STD is mounted on all print devices. The primary and secondary authorization levels may INCLUDE or REMOVE form names from the spooling environment. If a PRINT or PUNCH request specifies a form that has not been included in the spooling environment, the spooler will output the following message:

```
INVALID FORM: formname
```

and output the file on the default standard form STD.

4.4.2 Associating a Form With a Physical Device

Once a form has been identified as a valid form in the spooler environment, the user can specify what physical device that form is mounted on. This is accomplished via the MOUNT parameter of the FORM command.

Example:

```
FORM MOUNT TWOPART, PRT2:
```

The spooler is notified that form TWOPART is mounted on printer PRT2:.

Once a form has been associated with a device, the spooler retains this association information. Subsequent print requests that specify a particular form must be directed to the pseudo device that is associated with the printer holding the desired form. SPL/32 will direct the file to the desired physical device for output as long as the correct pseudo device is specified.

SPL/32 also provides the option to verify the alignment of a form that has been mounted on a printer.

Example:

```
FORM MOUNT TWOPART VERIFY=1, PRT1:
```

This notifies the spooler that the form TWOPART is mounted on device PRT1:. The device will be paused after the first page of the first file is printed, and the following message is output to the control task monitoring the printer:

```
VERIFY COMPLETE ON: device name
```

The operator can then realign the form (if necessary), rewind the input file and resume output.

If a file is placed on the spool queue that requires a form that is not currently mounted on any device, the following message is sent to the operator:

```
REQUEST IS PENDING FOR FORM form name
```

The spool file will remain on the queue until the specified form is mounted or until the file is flushed.

4.4.3 Removing Forms From the Spooling Environment

Forms are removed from the spooling environment via the REMOVE parameter of the FORM command.

Example:

```
FORM REMOVE REPORT1
```

Form REPORT1 is removed from the list of valid form names maintained by the spooler. Subsequent requests for the form REPORT1 are printed on the standard form STD and the warning message:

INVALID FORM: form name

is generated. Any entry on the spool queue that requires the removed form will remain on the queue until the required form is mounted.

4.5 USING IDENTIFICATION PAGES

SPL/32 provides the capability to uniquely identify printed output through the use of identification pages that precede the file to be output (BANNER PAGES) or follow the file to be output (TRAILER PAGES). The user has the option to use one or both of these identification pages. The user can specify the character to be used by the spooler to generate the BANNER or TRAILER pages. The default character is 0. The default BANNER and TRAILER pages contain the following information about each file:

- file descriptor,
- user account number,
- date and time the file was output, and
- taskname.

The user can specify any combination of these default parameters plus the following optional information:

- a 1- to 12-character user-specified message.

In addition to the above information, an information block is printed at the bottom of each BANNER or TRAILER page which consists of the following information:

- pseudo device
- real device
- full fd
- user account number
- queue time

The user has the option to specify whether single, double, or no identification pages are to be generated. Banner page and trailer page specifications are established via the BANNER and TRAILER commands, respectively.

Examples:

BANNER PDI:, DOUBLE

TRAILER PDI:, SINGLE

These examples establish a double banner page and a single trailer page for each file output on pseudo device PDI:. Each identification page will contain the default information set.

Banner and trailer page specifications can be directed at either a pseudo device or a physical device. Once a device has been associated with banner or trailer page specifications, all files subsequently output on that device will be identified according to that specification until a new BANNER or TRAILER command is issued.

4.6 USING THE CHECKPOINT FEATURE

The CHECKPOINT feature is a protective option of SPL/32 designed to eliminate the need to reprint entire files when the output of a file is interrupted due to a system failure or I/O error. If a failure occurs while the CHECKPOINT option is in effect, a file being output at the time of the failure is repositioned to the last record checkpointed, and output is resumed when the system is restarted. This is a particularly useful feature when very large files are being output because it eliminates the need to begin printing files from the beginning of file after failures.

To use the checkpoint feature, two requirements must be satisfied:

1. The global CHECKPOINT option must be on.
2. The CHECKPOINT option must be specified in the PRINT, PUNCH, ASSIGN, SPOOLFILE, or CHANGE command that submits a file for output.

If either of these conditions is not met, checkpointing is not performed. The global CHECKPOINT option is turned on via the CHECKPOINT command.

Example:

CHECKPOINT ON,80

This example turns the CHECKPOINT feature on and specifies that a checkpoint is to be performed after each 80 records output. The number specified represents the largest number of records that would have to be reprinted if a system failure occurs. Once the global CHECKPOINT feature has been turned on, as the above example shows, a user can request that a file be checkpointed by specifying the CHECKPOINT option in the PRINT, PUNCH, or CHANGE command.

Examples:

PRINT REPORT.NO1/119,DEVICE=PR2:,CHECKPOINT

PUNCH TEST32.CAL/120,DEVICE=PR3:,CHECKPOINT

CHANGE SCAN32.CAL/119,DEVICE=PD1:,CHECKPOINT

The files in these examples will be checkpointed when output, as long as the global checkpoint feature is on. If the global checkpoint feature is off, checkpointing will not be performed. Therefore, it is advisable to make sure the global checkpoint feature is on before attempting to use the CHECKPOINT feature. Entering the CHECKPOINT command without parameters will result in a display of the checkpoint value if global checkpointing is on. If no value is displayed, global checkpointing is off.

4.6.1 Turning Checkpointing Off

Global checkpointing is turned off via the OFF parameter of the CHECKPOINT command.

Example:

CHECKPOINT OFF

The global checkpoint feature is off and checkpointing cannot be performed. Checkpointing can also be turned off at the user-level if the global checkpoint feature is on. This is accomplished by specifying the NOCHECKPOINT parameter in a PRINT, PUNCH, ASSIGN, SPOOLFILE, or CHANGE command.

Example:

```
PRINT TESTPROG.CAL/119,DEVICE=PR:,NOCHECKPOINT
PUNCH STATTEST.CAL/120,DEVICE=PD2:,NOCHECKPOINT
CHANGE REPORT.TEX/118,DEVICE=PD7:,NOCHECKPOINT
```

4.7 OUTPUT PRIORITY

Spool files are output on a priority basis. Each file assigned for output is given a priority value, either implicitly or explicitly. This priority number is used by the spooler to determine the order in which files will be output; files with the lower priority number are output first. The priority of a spool file is explicitly assigned in the PRIORITY= parameter of a PRINT, PUNCH, or CHANGE request. When this priority is not specified, the file is given the default priority of the system. When a file is placed on the spool queue, it has two priorities associated with it:

- a global priority, and
- a user priority.

The global priority is the priority of the file with respect to all other spool files. The user priority is the priority of the file with respect to all spool files initiating from the same user and having the same global priority.

When a PRINT, PUNCH, or CHANGE command is issued from a unit with primary or secondary authorization, the PRIORITY= keyword refers to the global priority of the file. When a PRINT, PUNCH, or CHANGE command is issued from a unit with user authorization, the PRIORITY= keyword refers to the user priority of the file. See Chapter 5 for a discussion of authorization levels.

4.7.1 Changing the Global Default Priority

The global default priority value is changed via the PRIORITY command.

Example:

```
PRIORITY 132
```

This example sets the global default priority to 132.

If the PRIORITY command is omitted at spooler startup, the default priority is set at 128. The priority may be changed while the spooler is active, but only primary authorization can change the global default priority.

4.7.2 Altering the Output Priority of a Spool File

SPL/32 provides the capability to alter the output priority of spool files waiting on the spool queue. This is accomplished via the PRIORITY= keyword of the CHANGE command.

For example, if the file PAYROLL.CHK in account 119 was residing on the spool queue waiting to be printed, the file has a priority of 118. To alter that priority the user enters the following CHANGE command:

Example:

```
CHANGE PAYROLL.CHK/119,PRIORITY=120
```

The output priority of the file PAYROLL.CHK is altered to 120.

The degree to which the output priority of a file can be altered depends upon the authorization level of the CHANGE command. The following general conventions are used:

- Primary authorization allows the output priority of a file to be altered relative to all other files in the spooling environment.
- Secondary authorization allows the output priority of a file to be altered relative to all other files in the spooling environment that were placed on the spool queue with the same secondary authorization.
- User authorization allows a user to alter the output priority of a file relative to any other files placed on the queue by that user.

CHAPTER 5 TASKS AND LEVELS OF AUTHORIZATION

5.1 INTRODUCTION

SPL/32 recognizes three types of tasks in the spooling environment:

- Control tasks
- Subcontrol tasks
- User-level tasks

These three task types have descending degrees of control in the spooling environment relative to issuing spooler commands, controlling spooled devices, and monitoring spooled devices. The capabilities that a spooler command has when issued from one of these types of tasks is directly related to the authorization level of the issuing task. There are three levels of authorization:

- Primary
- Secondary
- User

The three types of tasks and three levels of authorization are not directly related. Control tasks are not limited to primary authorization because they can also have secondary or user authorization. Subcontrol tasks can have secondary or user authorization. User-level tasks are restricted to user authorization.

Each of these three tasks is designed to perform a specific function or functions. Control tasks communicate directly with the spooler and receive responses directly from the spooler.

Subcontrol tasks communicate directly with the spooler but can have spooler responses selectively routed either to themselves or to an associated control task. User-level tasks communicate with the spooler only through a control task or via an assign to a pseudo device.

5.2 CONTROL TASKS AND THEIR COMMUNICATION WITH THE SPOOLER

Control tasks are designed to control spool devices and the spooling environment and to monitor the status of these devices. Control of spool devices is accomplished through the use of spooler commands. Control tasks can issue command requests directly to the spooler and receive messages or data as a consequence of the request. Control tasks monitor devices by receiving messages from the spooler that pertain to device status or I/O status within the control task's environment. Control tasks communicate with the spooler via three types of messages. These specify functions that control tasks perform on behalf of the tasks/devices in their environment. The three message types are:

- Command messages
- Acceptance messages
- Request for mnemonic table messages

The formats of these messages are presented in Appendix D. All communications between tasks and the spooler are preceded by general information that identifies the origin of the message, the type of message, and other related information. This information is contained in a record called the message header. Appendix D details the contents and format of the message header.

5.2.1 Command Messages

Command messages are sent from a control task to the spooler and contain spooler commands that direct the spooler to perform a service. These commands can be sent in ASCII or preparsed form.

5.2.2 Acceptance Messages

Acceptance messages are sent from a control task to the spooler as a response to an authorization identifier change request. Acceptance messages indicate whether the authorization identifier passed to a particular control task is acceptable or not.

5.2.3 Request for Mnemonic Table Messages

Request for mnemonic table messages are sent from a control task to the spooler to request the spooler's command mnemonic table. The control task can then use this table to identify valid command mnemonics.

5.3 SPOOLER COMMUNICATION WITH CONTROL TASKS

The spooler communicates with control tasks via five types of messages:

- Service response messages
- Exception response messages
- Authority messages
- Attention messages
- Mnemonic messages

The formats of these messages are presented in Appendix D.

5.3.1 Service Response Messages

Service response messages are sent from the spooler to a control task when a command issued from the control task is valid and requires a data response from the spooler.

5.3.2 Exception Response Messages

Exception response messages are sent from the spooler to a control task when an invalid command is issued from the control task.

5.3.3 Authority Messages

Authority messages are sent from the spooler to a receiving control task. The receiving control task is notified that it is being given the authority to recognize attention messages.

5.3.4 Attention Messages

Attention messages are sent from the spooler to a control task to notify the control task of a situation that occurred internal to the spooler and requires a control task or operator to be notified.

5.3.5 Mnemonic Messages

Mnemonic messages are sent from the spooler to a control task to notify the control task of the valid spooler commands that should be passed to the spooler for execution.

5.4 INCLUDING CONTROL TASKS IN THE SPOOLING ENVIRONMENT

Control tasks are added to the spooling environment via the INCLUDE parameter of the CTASK command.

Example:

```
CTASK INCLUDE.MTM
```

The spooler now recognizes MTM as a control task. Control tasks can be specified at spooler start time in the START command file or they can be added to the spooling environment while the spooler is active. If no control tasks are specified in the spooling environment at start time, the spooler, by default, specifies control task .SPL. Communication with the spooler through the control task .SPL is accomplished by entering spooler commands at the system console.

5.5 DISPLAYING CONTROL TASKS IN THE SPOOLING ENVIRONMENT

The names of all control tasks in the spooling environment can be displayed by entering the CTASK command without parameters.

In the following example, tasks TEST1, TEST2, and TEST3 are designated as control tasks. The CTASK command is then entered without parameters to display the control tasks in the spooling environment.

Example:

```
*TASK INCLUDE TEST1,TEST2,TEST3
*CTASK
*CTASKS=TEST1,TEST2,TEST3
```

5.6 REMOVING CONTROL TASKS FROM THE SPOOLING ENVIRONMENT

Control tasks are removed from the spooling environment via the REMOVE parameter of the CTASK command.

Example:

```
CTASK REMOVE .MTM
```


Control task MTM is removed from the spooling environment. When a control task is removed from the spooling environment, any subcontrol tasks associated with the control task are also removed. See Section 5.7 on subcontrol tasks.

Control tasks may be removed from the spooling environment while the spooler is active.

5.7 SUBCONTROL TASKS AND THEIR COMMUNICATION WITH THE SPOOLER

Subcontrol tasks issue command requests directly to the spooler. Messages or data generated by the spooler in response to a command issued from a subcontrol task can be routed, either to the subcontrol task that issued the command or to the control task that is monitoring the subcontrol task that issued the command. The routing is determined by the routing flag field located in the message header that precedes the subcontrol command request. See Appendix D for detailed information on this procedure. Subcontrol tasks communicate with the spooler via command messages.

The format of a command message is presented in Appendix D. All communication between subcontrol tasks and the spooler is preceded by general information that identifies the origin of a message, the type of message, and other related information. This information is contained in a record called the message header. Appendix D presents a detailed breakdown of the message header record.

5.7.1 Command Messages

Command messages are sent from a subcontrol task to the spooler and contain spooler commands that direct the spooler to perform a service. These commands can be issued in ASCII or preprepared form.

5.8 SPOOLER COMMUNICATION WITH SUBCONTROL TASKS

A subcontrol task can only receive responses to command messages it issued. There are two types of responses a subcontrol task can receive from the spooler:

- Service response messages
- Exception response messages

The formats of these records are presented in Appendix D.

5.8.1 Service Response Messages

Service response messages are sent from the spooler to a subcontrol task when a command issued by the subcontrol task is valid and requires a data response.

5.8.2 Exception Response Messages

Exception response messages are sent from the spooler to a subcontrol task when an invalid command is issued by the subcontrol task.

5.9 INCLUDING SUBCONTROL TASKS IN THE SPOOLING ENVIRONMENT

Subcontrol tasks are included in the spooling environment via the INCLUDE parameter of the SCTASK command.

Example:

```
SCTASK CTASK=.MTM INCLUDE TASKONE
```

TASKONE is designated as a subcontrol task in the spooling environment with the control task .MTM as its monitor task.

When a subcontrol task is included in the spooling environment, it must be associated with a monitor control task. This monitor control task can be explicitly specified in the SCTASK command, as in the above example. If specification of a monitor control task is omitted, the spooler implicitly associates the control task that issued the SCTASK command as the monitor control task.

5.10 DISPLAYING SUBCONTROL TASKS IN THE SPOOLING ENVIRONMENT

The subcontrol tasks present in the spooling environment can be displayed by entering the SCTASK command without parameters. The monitor control task is displayed with each subcontrol task.

In the following example, a number of subcontrol tasks are assigned under two monitor control tasks, MONITOR1 and MONITOR2. These subcontrol tasks are then displayed with their associated monitor control task.

Example:

```
*SCTASK CTASK=MONITOR1 INCLUDE TASK1,TASK2
*SCTASK CTASK=MONITOR2 INCLUDE TASK3,TASK4
*SCTASK
*CTASK=MONITOR1 SCTASKS=TASK1,TASK2
*CTASK=MONITOR2 SCTASKS=TASK3,TASK4
```

5.11 REMOVING SUBCONTROL TASKS FROM THE SPOOLING ENVIRONMENT

Subcontrol tasks are removed from the spooling environment in one of two ways:

- explicitly, via the REMOVE parameter of the SCTASK command, or
- implicitly, by removing the monitor control task of the subcontrol task.

5.12 USER-LEVEL TASKS AND THEIR COMMUNICATION WITH THE SPOOLER

Tasks within the spooling environment that are not designated as either control or subcontrol tasks are known as user-level tasks. User-level tasks cannot communicate directly with the spooler but can request the use of spooler functions in one of two ways:

- by assigning to a pseudo device, and
- by passing spooler commands to the spooler through a monitor control task.

When a user-level task assigns to a pseudo device, the operating system intercepts the SVC7 and then routes it to the spooler for execution. When a user-level task issues a command, such as the INTERROGATE command, the command is first submitted to the control task monitoring the user-level task. This control task interprets the command and passes it to the spooler. The spooler does not track user-level tasks within the spooling environment. Therefore, commands originating from user-level tasks appear, to the spooler, to be originating from a control task. User-level tasks can actively use five spooler commands:

- PRINT
- PUNCH
- CHANGE
- FLUSH
- INTERROGATE

These commands can only be used to affect entries that were placed on the spool queue on behalf of that user.

A number of additional commands can be used by the user-level tasks to display current spooler environment information. However, the user-level cannot alter this environment.

5.13 LEVELS OF AUTHORIZATION

Spooler commands have three levels of authorization associated with them:

- Primary
- Secondary
- User

Primary authorization is assigned to a task/device via the PRIMARY command. Secondary authorization is assigned to a task/device using the SECONDARY command. User authorization is implicitly assigned whenever a task/device is not specified in a PRIMARY or SECONDARY command.

Associated with each level of authorization is a set of the spooler commands that are valid only when used from a task/device that has the appropriate authorization level. This design enables the system administrator to maintain control over the spooler environment by limiting the number of tasks/devices that can alter the spooling environment or manipulate the status of spooled devices. Commands that require primary authorization, the highest level, are used to establish and maintain the global spooling environment. A task or device that has primary authorization is able to use all spooler commands. Commands that require secondary authorization, the intermediate level, are used to control devices in a specific group of the spooling environment. A task or device that has secondary authorization is able to use commands at the secondary and user-levels, but cannot use the primary level commands.

Commands that require user authorization, the lowest level, can alter the status of file entries placed on the spool queue on behalf of the user and can display information pertaining to the spooling environment. A task or device that has user-level authorization is restricted to using the user-level commands.

Certain spooler commands are available for use at all three levels of authorization. However, the function these commands perform is dependent upon the authorization level of the task/device that issued the command. An INTERROGATE command issued at the primary level of authorization displays information about all entries on the spool queue. An INTERROGATE command issued at the secondary level of authorization displays information about the spool queue entries that were placed on the queue by a device or task using the same secondary authorization identifier as that contained in the INTERROGATE command. An INTERROGATE command issued at the user-level of authorization displays entries on the spool queue that were sent by that user.

Another command that functions in this way is the CHANGE command. A CHANGE command issued at the primary level can modify any entry on the spool queue. A CHANGE command issued at the secondary level can modify the files placed on the spool queue by a device or task using the same secondary authorization identifier as that contained in the CHANGE command. A CHANGE command issued at the user-level can only alter file entries on the spool queue that were placed there by the user. Certain spooler commands set, specify, or alter a parameter when issued at the primary level of authorization. However, these same commands can only be used to display these parameters when issued at the secondary or user-levels of authorization.

An example of this is the BLOCK command. To set the default data and index block sizes, the BLOCK command requires primary authorization. When the BLOCK command is issued at the secondary or user-levels of authorization, the default data and index block sizes are displayed. Table 5-1 presents a summary of spooler commands available at each level of authorization and includes the function of each command at each level of authorization.

TABLE 5-1 AUTHORIZATION LEVELS AND ASSOCIATED SPOOLER COMMANDS

SPOOLER COMMAND	AUTHORIZATION LEVELS		
	PRIMARY	SECONDARY	USER
ATTACH	0	NA	NA
BANNER	0	0	NA
BLOCK	▲	▲	▲
CHANGE	*	*	*
CHECKPOINT	0	▲	▲
CREATE	0	NA	NA
CTASK	0	▲	▲
DELETE	0	NA	NA
DETACH	0	NA	NA
DEVICE	0	0	NA
FLUSH	0	*	*
FORM	0	0	▲
GLOBAL	0	▲	▲
GROUP	0	▲	▲
INPUT	0	NA	NA
INTERROGATE	*	*	*
PRIMARY	0	NA	NA
PRINT	*	*	*
PRIORITY	0	▲	▲
PUNCH	*	*	*
QUEUE	0	NA	NA
QUIESCE	0	NA	NA
SCTASK	0	▲	▲
SECONDARY	0	NA	NA
SPVOLUME	0	▲	▲
TRAILER	0	0	NA

0 = authorization level able to set, specify, or alter parameter.

▲ = authorization level able to display information.

* = authorization level able to use command. The impact of the command is limited according to the authorization level of the issuing task/device.

NA = command not available at this authorization level.

5.13.1 Authorization Identifiers (AID)

The spooler recognizes the authorization level of a command request by reading the authorization identifier (AID) that accompanies it. The AID is carried in the message header of the command request (see Appendix D). The spooler then determines if the task or device issuing the command has the authority to issue that command. The AID is a 1- to 8-character alphanumeric name, the first character must be alphabetic, the remaining alphanumeric. There are two types of AIDs that can be present in the spooling environment:

- Primary AID
- Secondary AID

There can be only one primary AID but there can be more than one secondary AID. The primary AID is specified or changed using the PRIMARY command. The PRIMARY command denotes a control task, that is to be the primary control task, and the AID that will accompany command requests that require primary authorization. The primary AID may be specified at spooler start time in the START command file and it may be changed while the spooler is active. If the PRIMARY command is omitted from the START command file, the task .SPL is loaded and becomes the default primary control task and INITPAID becomes the default primary AID.

Secondary AIDs are specified using the SECONDARY command. The SECONDARY command specifies the AID, a control task, and a pseudo device to which the AID applies. The specified control task monitors the devices associated with the specified pseudo device. Command requests that will affect the status of devices associated with the specified pseudo device must contain the secondary AID or they will be rejected by the spooler.

Secondary authorization provides control over a set of devices grouped under a particular pseudo device. The AID functions as a password preventing unauthorized access to the group environment. There is no unique AID associated with the user-level of authorization. Command requests that do not contain a primary or secondary AID are implicitly assigned user authorization by the spooler.

5.13.2 Origin Identifiers (OID)

The origin identifier (OID) is a 1- to 8-character alphanumeric name that uniquely identifies a task or terminal. The OID is used by control tasks and the spooler. Control tasks use the OID to route messages or data from the spooler to the appropriate location within the control tasks environment. The spooler uses the OID (together with the AID) to determine default print locations for members of groups within the spooling environment.

CHAPTER 6 SPL/32 COMMANDS

6.1 INTRODUCTION

Control of SPL/32 and the interaction between the user and SPL/32 is accomplished through the use of SPL/32 commands. SPL/32 commands may be entered at the system console and/or the user's terminal. The commands available to a particular terminal are directly related to the authority level of that terminal (see Chapter 5). This chapter presents each SPL/32 command with complete format and optional parameters, a description of the command's function, and examples of the use of each command.

NOTE

The use of colons when specifying pseudo devices or real devices within a command line is entirely optional.

6.2 ATTACH COMMAND

The ATTACH command is used to logically associate a pseudo device (arbitrarily denoted by the user) with one or more real output devices; e.g., printers. This association may be accomplished without interruption of current spooler execution.

Format:

```
ATTACH PDEVICE=dev RDEVICE=dev1 [,dev2 ... ,dev19]
```

Parameters:

dev is a 1- to 4-character alphanumeric designator (of which the first character must be alphabetic and the remaining characters alphanumeric) of the pseudo device to be implicitly created as a pseudo device if it does not exist in the spooler environment.

dev1,dev2,
...dev19 is a 1- to 4-character alphanumeric designator (of which the first character must be alphabetic and the remaining characters alphanumeric) of the real device (or devices) to be associated with the preceding pseudo device. Directing subsequent output to the pseudo device will result in the output being generated on the associated real device. A total of up to 19 real devices can be specified in a single ATTACH command.

Functional Details:

If more than one real device is associated with a pseudo device, the record length of the pseudo device is defined by the real device with the largest record length entered in the ATTACH command. The pseudo device attributes are defined by the last real device entered in an ATTACH command for that pseudo device. Therefore, when multiple real devices are associated with a single pseudo device, it is advisable to ensure that these real devices have like device attributes. Primary or secondary control unit authorization is required to use the ATTACH command.

Examples:

ATTACH PDEVICE=PR1: RDEVICE= PRT1:, PRT2:, PRT3:

The real devices PRT1: and PRT2: are associated with pseudo device PR1:.

ATTACH PD=PR1: RD=PRT3:

The real device PRT3: is associated with the pseudo device PR1:.

BANNER

6.3 BANNER COMMAND

The BANNER command is used to specify the number of identification page(s) printed with each file output on a specified pseudo or physical device, and to format the contents of this identification page to suit the user's needs.

Format:

$$\text{BANNER} \left[\begin{array}{l} \text{pseudo dev:} \\ \text{real dev:} \end{array} \right] \left[\begin{array}{c} \text{DOUBLE} \\ \text{SINGLE} \\ \text{NONE} \end{array} \right] \left[\text{, CHARACTER=character} \right] \left[\text{, fd} \right]$$

Parameters:

- | | |
|-------------|---|
| pseudo dev: | specifies the pseudo device or real device to |
| real dev: | which the BANNER command specifications pertain. |
| DOUBLE | specifies that two banner pages are to be produced for each file output on the specified device. |
| SINGLE | specifies that one banner page is to be produced for each file output on the specified device. |
| NONE | specifies that no banner page is to be produced for each file output on the specified device. This option is useful for very slow printing devices, card punches, or printers using preprinted forms. |
| CHARACTER | Character specifies a single ASCII character that the spooler will use to print the contents of the BANNER pages. The default is the letter O. |
| fd | is the file descriptor of an information file that contains any combination of the following six keywords. The user has the option to select up to five of these keywords and include them in the file. These keywords signify the information to be printed on the BANNER pages. |

FD	formats the filename.
AC	formats the user account number.
TASKNAME	formats the task name.
DATE	formats the date.
TIME	formats the time of day.
LOGO='text'	formats a 1- to 12-character user-defined text record. The text record must be contained within the quotation marks.

The user has the option to select up to five of the above keywords and include them in the file. These keywords signify the information to be printed on the BANNER pages.

Functional Details:

The default condition is a one page banner for a pseudo or real device with the following information:

- Filename
- Account number
- Taskname
- Date
- Time

If the BANNER command is directed to a pseudo device; the specifications in the command apply to all real devices associated with that pseudo device. If a real device is associated with two or more pseudo devices, the latest BANNER command entered for any of the pseudo devices will apply to the real device. This command is only valid from a primary control unit or a secondary control unit for the pseudo device associated with the secondary AID.

In addition to the 132-column BANNER output, an information block is output with each file. This information block contains the following information:

- Full filename
- User account number
- Date and time the file was placed on the spool queue
- Date and time the file was output
- The physical output device
- The pseudo device name

Example:

```
BANNER PRN1:,S,CHARACTER=A
```

All output on device PRN1: will be output with single page banners containing the default information set. The BANNER page contents will be printed using the letter A.

6.4 BLOCK COMMAND

The BLOCK command is used to display the current physical data block size and index block size or to set the default values for these two parameters.

Format:

BLOCK blksize/indexsize

Parameters:

blksize is a decimal number indicating the data block size in 256-byte sectors.

indexsize is a decimal number indicating the index block size in 256-byte sectors.

Functional Details:

The BLOCK command can be used to display or set the data block size and index block size, in 256-byte sectors, for buffering and debuffering operations on spool files. To display the current values, BLOCK is entered without any additional parameters. To set the default block size (data and/or index) or change the current values, the blksize/indexsize parameters are used with the BLOCK command. When setting new values, do not exceed the maximum block size established at system generation (sysgen). The default value is one sector. Primary control unit authorization is required to set the data and index block sizes. Secondary and user-level control unit authority may display the current data and index block sizes.

Examples:

BLOCK 5/3

The default data block size for the allocation of spool files is 5, the default index blocking factor is 3.

BLOCK

The current data block size and index block size will be displayed.

CHANGE

6.5 CHANGE COMMAND

The CHANGE command is used to modify the characteristics of a spool file which is currently residing on the spool queue.

Format:

```
CHANGE fd DEVICE=pseud dev: [,COPIES=n][,FORM=formname][,PRIORITY=
[ {IMAGE} ] [ {NOIMAGE} ] [ {DELETE } ] [ {CHECKPOINT } ] [ {HOLD } ]
[ {VFC } ] [ {NOVFC } ] [ {NODELETE } ] [ {NOCHECKPOINT} ] [ {RELEASE} ]
```

Parameters:

fd	is the file descriptor (name) of the file to be modified.
DEVICE	allows the pseudo device specification to be changed.
COPIES	allows a change in the number of copies to be output.
FORM	allows preprinted forms specification to be changed.
PRIORITY	allows the output priority of a file to be altered. The capability of a user to alter the priority of a file is dependent upon the authorization level of that user. Primary authorization can change the priority of any file on the spool queue. Secondary authorization can change the priority of any file on the spool queue with the same secondary authorization identifier. User authorization can change the priority of files that the user placed on the spool queue.
IMAGE or VFC	allows a change in the output format of a file; either output in IMAGE (as is) or with VFC (vertical forms control). These parameters are mutually exclusive.

NOIMAGE or NOVFC	allows a change to the output format of a file; either turns VFC off or resets the image option. These parameters are mutually exclusive.
DELETE or NODELETE	allows a change to the handling of a file after output; if the file was allocated by the spooler, the default is DELETE. If the file is a standard user file, the default is NODELETE.
CHECKPOINT or NOCHECKPOINT	turns on or off checkpointing for a given file. If global checkpointing is not turned on in the spooling environment no checkpointing takes place. These parameters are mutually exclusive.
HOLD	causes a print request to be held on the queue. Request is held until a RELEASE is issued.
RELEASE	enables a print request which has been held to become available for output.

Functional Details:

If two conflicting parameters are entered in a CHANGE command, the default prevails. The CHANGE command is available to all authority levels.

Example:

```
CHANGE M300:REPORT.LST/5,HOLD,VFC
```

The file M300:REPORT.LST/5 is placed in a hold state. It will be output with vertical forms control (VFC) when a RELEASE command enables the file to be output.

CHECKPOINT

6.6 CHECKPOINT COMMAND

The CHECKPOINT command is used to set or display the checkpoint value. When set ON, each file placed on the queue will have an associated record number that is updated after every N records have been output.

Format:

CHECKPOINT $\left[\begin{array}{l} \{ON, N\} \\ \{OFF\} \end{array} \right]$

Parameters:

ON turns the checkpoint option on and sets N as the number of records to be output between queue updates. N must be a number between 1 and 2,147,483,647.

NOTE

A disk I/O is performed at each checkpoint, therefore, setting N to a very small number may degrade spooler and system performance.

OFF turns the checkpoint option off. This is the default condition.

Functional Details:

Checkpointing is a recovery procedure that provides the capability to restart spooling at the point where it stopped after a failure by keeping a record number pointer for each file in the queue. This avoids redundant output of already printed data and is especially useful when outputting very large files. As a file is output, the record number in the spool queue is updated to follow the progress of the file being output. As a result, when the spooler is restarted after a system failure, output can be requested to begin from the last record checkpointed instead of returning to the beginning of the file.

Only the primary control unit can set N in the CHECKPOINT ON command. However, any user may enter CHECKPOINT without optional parameters to display the current value of N.

The user can turn checkpointing on or off on a file-by-file basis via the CHECKPOINT option of the PRINT, PUNCH, CHANGE, or SPOOLFILE commands.

Examples:

CHECKPOINT ON,80

Sets checkpointing feature on. Every 80 records output will result in a queue update.

CHECKPOINT

Displays the current checkpoint number.

```
-----  
| CREATE |  
-----
```

6.7 CREATE COMMAND

The CREATE command is used to define and initiate pseudo device(s) in the spooler environment.

Format:

```
CREATE pseud dev1: [, ...pseud dev20:]
```

Parameters:

pseud dev1: is the 1- to 4-character name of the pseudo device to be created. The first character must be alphabetic and the remaining characters alphanumeric. The pseudo device cannot already exist in the spooler environment. Up to 20 pseudo device names can appear in a single CREATE command.

Functional Details:

The CREATE command is used to define a pseudo device that is then available for use in the ASSIGN, PRINT, PUNCH, GROUP, and ATTACH commands. Primary control unit authority is required to create pseudo devices.

If a user wants to display the names of the pseudo devices currently present in the spooling environment, the DISPLAY DEVICES command is available. See the OS/32 Operator Reference Manual for details.

Example:

```
CREATE PR1:,PR2:
```

Creates two pseudo devices, PR1: and PR2:.

6.8 CTASK COMMAND

The CTASK command is used to add or remove a task or tasks, called control tasks, which are capable of having a dialogue with the spooler and to display all such control tasks in the spooler environment.

Format:

```
CTASK  [ { INCLUDE task1 , ..., task10 }  
        { REMOVE task2 , ..., task10 } ]
```

Parameters:

INCLUDE	is the keyword used to add a control task to the spooler environment.
REMOVE	is the keyword used to remove a control task from the spooler environment.
task1 [task10]	identifies the specific control task(s) to be added or removed from the spooler environment. If a task is to be added to the spooler environment, it must first be loaded into memory using the LOAD command.

Functional Details:

If the CTASK command is entered without parameters, all control tasks in the spooling environment are displayed. The CTASK command can be used to manipulate the number of control tasks in the spooler environment. Control tasks are able to have a dialogue relationship to the spooler; i.e., able to send and receive messages from the spooler. If a control task having one or more subcontrol tasks associated with it is removed from the spooler environment (see SCTASK command), the subcontrol tasks are also removed.

If the primary control task is removed from the spooling environment, the following message is generated:

```
PRIMARY CTASK WAS CANCELLED
```

The default primary control task .SPL is then loaded as the primary control task.

If the default primary control task .SPL is removed, the message shown above is generated and then the default primary control task .SPL is automatically reloaded by the spooler.

Examples:

```
CTASK INCLUDE T1,T2
```

Establishes the tasks T1 and T2 as control tasks.

```
CTASK REMOVE T2
```

Removes control task T2 from the spooler environment. Any subcontrol tasks associated with T2 would be removed at this time.

6.9 DELETE COMMAND

The DELETE command is used to remove a pseudo device(s) and/or a physical device(s) from the spooler environment.

Format:

```
DELETE { (pseud dev1: ,...,pseud dev20: )  
        (real dev1: ,...,real dev20: ) }
```

Parameters:

pseud dev1: ,...,pseud dev20:	identify removed.	the pseudo	devices to be
real dev1: ,...,real dev20:	identify removed.	the real	devices to be

Functional Details:

Up to 20 devices, pseudo, real, or a combination of the two may be specified in a single DELETE command. When a pseudo device is removed, its association with a real device (or devices) is deleted. Subsequent assignments to the pseudo device will be rejected, but current assignments to the pseudo device are allowed to complete. When a real device is deleted, its association with a pseudo device(s) is broken. Output on the deleted real device will be allowed to complete. Secondary and user control units are not able to execute this command. It is valid from a primary control unit only.

Example:

```
DELETE PRN1:,PRN2:
```

The pseudo devices PRN1: and PRN2: are deleted from the spooling environment. Ongoing I/O will proceed until completion, new assignments to these devices will be rejected.

```
-----  
|   DETACH   |  
-----
```

6.10 DETACH COMMAND

The DETACH command is used to break the association between a real device and a pseudo device as established through the ATTACH command.

Format:

```
DETACH PDEVICE=pseud dev RDEVICE=real dev1: [...real dev19:]
```

Parameters:

pseud dev: is the name of the pseudo device the real device is to be detached from.

real dev1:
,...real
dev19: is the name of the real device to be detached.

Functional Details:

The DETACH command dissolves the logical association between a pseudo device and a real device. If the real device specified is not associated with any other pseudo device, the logical unit (lu) assigned to that real device is closed. However, if the real device is associated with other pseudo devices, these associations remain unaffected.

Example:

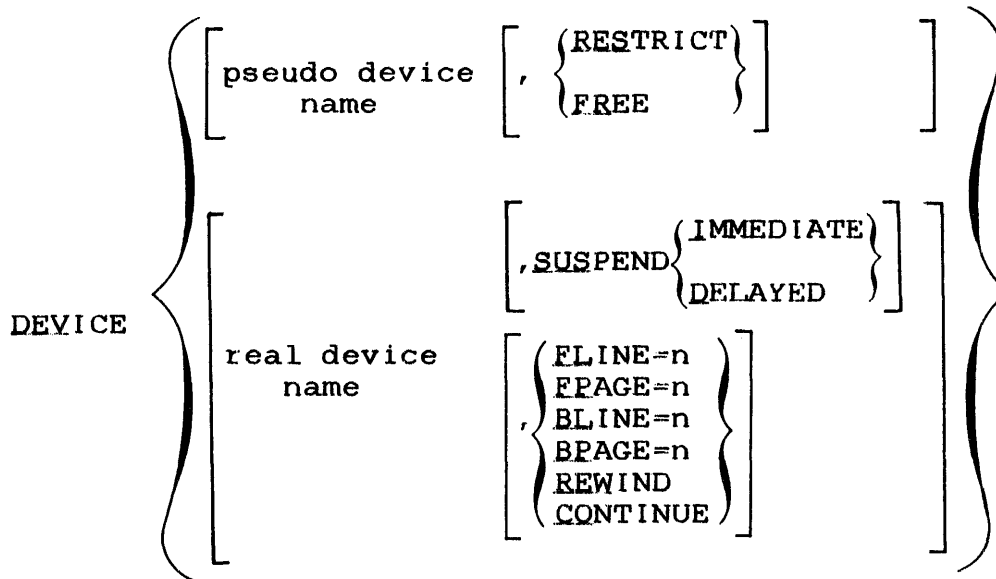
```
DETACH PD=PRN1 RD=PRT1,PRT2
```

The real devices PRT1 and PRT2 are detached from pseudo device PRN1.

6.11 DEVICE COMMAND

The DEVICE command is used to alter the status of a pseudo device or a physical device.

Format:



Parameters:

- pseudo dev: is the name of the pseudo device to which the command is directed.
- RESTRICT indicates that the real devices associated with the preceding pseudo device will only receive output if the pseudo device is expressly requested through the use of an assign call or by specifying the pseudo device in the 'DEVICE=' parameter of a PRINT/PUNCH/CHANGE command. This parameter is valid only in reference to a pseudo device.
- FREE resets the RESTRICT parameter condition. The pseudo device is used for output whenever it becomes available. This parameter is valid only in reference to a pseudo device.
- real dev: is the name of a real device to which the command is directed.

SUSPEND causes the specified real device to suspend output. If the DELAYED option is included, the device will be suspended after the currently printing file has completed. If the IMMEDIATE option is included, the device will be suspended immediately. This is the default option.

NOTE

This option must be used prior to using any of the following options.

FLINE indicates to forward space a print file n lines. The real device to which the command is issued must be suspended before this parameter is used. Output will automatically resume on a paused device after the entry of this parameter. The device need not be continued.

FPAGE indicates to forward space a print file n pages. If page length was not already defined in a FORM command, a default length of 66 lines is assumed. Output will automatically resume on a paused device after the entry of this parameter. The device need not be continued.

BLINE indicates to backward space a print file n lines. The real device to which this command is directed must be suspended before this parameter is used. Output will automatically resume on a paused device after the entry of this parameter. The device need not be continued.

BPAGE indicates to backward space a print file n pages. Page length conventions are the same as explained above for the FPAGE command. Output will automatically resume on a paused device after the entry of this parameter. The device need not be continued.

REWIND this causes the file currently spooling on the specified real device to be rewound and brought to the beginning of file, where spooling continues. Output will automatically resume on a paused device after the entry of this parameter. The device need not be continued.

NOTE

The file positioning parameters FLINE, FPAGE, BLINE, BPAGE, and REWIND can only be used when a file is actually being output. These parameters have no affect when entered while a Banner or Trailer page is being output, and if entered at that time they are ignored and no error message is generated.

CONTINUE restarts a suspended device. If a file was being spooled when the device was suspended, the next record in the file being spooled is processed.

Functional Details:

The DEVICE command requires primary or secondary level authorization. The DEVICE command enables orderly halting of spooled devices to correct paper jams, change forms and bypass printing long documents when only a few pages are needed.

Example:

DEVICE PRT1:,SUSPEND

The output of a file on device PRT1: is temporarily suspended.

DEVICE PRT1:,REWIND

The file being output on device PRT1: when it was suspended is rewound to the beginning of file and output is resumed.

```
-----  
|   END   |  
-----
```

6.12 END COMMAND

The END command is used to end spooling either immediately or after files currently on the spool queue are output.

Format:

```
END  [ { DELAYED }  
      { IMMEDIATE } ]
```

Parameters:

DELAYED causes spooler functions to end after all currently printing files are output. No new print or assignment requests are accepted. This option is also the default.

IMMEDIATE causes spooler functions to cease and all pseudo devices and real devices to be released immediately.

Functional Details:

The END command is only valid from a primary control unit.

6.13 FLUSH COMMAND

The FLUSH command is used to:

- delete a file from the spool queue,
- delete a file from the spool queue and from disk in one operation, or
- stop the output of a file on a specified device.

Format:

FLUSH [(fd [DELETE])
real device:]

Parameters:

fd is the name of the file to be removed from the spool queue. If the file is not a spool file (allocated by SPL/32), it is not deleted from disk. If the file is a spool file (that is allocated by the SPL/32), it is automatically deleted from disk.

DELETE specifies that the file is to be deleted from disk after it is removed from the spool queue regardless of whether it was allocated by SPL/32 or not. If the file is currently being output, it will not be deleted from disk.

real device: is the name of a real device. The file being output on this device is removed from the spool queue and output ceases immediately.

Functional Details:

The FLUSH dev: option requires primary or secondary level authorization. A user can only FLUSH files which originated from that user and then must use the FLUSH fd option.

Examples:

FLUSH M300:REPORT.LST/5,DEL

File M300:REPORT.LST in account 5 is removed from the spool queue. The file is also deleted from disk unless the file is currently being output.

FLUSH PRT1:

The file currently being printed on PRT1: is stopped and removed from the spool queue. The file is deleted from disk if it was allocated by the spooler. Other file types are not deleted from disk. Only primary or secondary units can perform this operation.

6.14 FORM COMMAND

The FORM command has three functions:

- to declare valid form names,
- to show which forms are to be mounted on each of the printers, and
- to display valid form names.

Format:

```

FORM  [ { INCLUDE frm1 [ { PLENGTH=p }
    { BMARGIN=b }
    { TMARGIN=t }
    { PWIDTH=pw } ] [ ..., frm4 ] [ { PLENGTH=p }
    { BMARGIN=b }
    { TMARGIN=t }
    { PWIDTH=pw } ] }
    REMOVE frm1 [, frm2 ] [, ..., frm5 ]
    MOUNT  frm1 [VERIFY=n ,dev1:] [, ...,dev15:] ] ]
  
```

Parameters:

INCLUDE	is used to specify a valid form name (1- to 8-characters long) for each unique paper type. The default form name is STD. Up to four forms can be included in a single command line.
PLENGTH	is the number of lines per page for the form specified in the INCLUDE command. Default is 66 lines per page.
BMARGIN	is the number of blank lines to be left at the bottom of the page. Default is 5 lines.
TMARGIN	is the number of blank lines to be left at the top of the page. Default is 5 lines.
PWIDTH	is the number of print positions horizontally across the page.
REMOVE	causes a currently valid form name to be removed from the list of valid form names.

MOUNT this option of the FORM command is used to associate a particular form type with a particular printer. Subsequent output that requires that particular form can then be directed to the device that has the form mounted.

VERIFY this option of the MOUNT parameter allows the user to specify that forms alignment is to be verified. The user can specify (in number of pages) at what point alignment should be verified. Output of the first file on the device will be suspended after n pages have been output. The operator can then verify forms alignment and continue the device, or realign the form, rewind to beginning of file, and resume output.

dev1:,..., is the device the FORM MOUNT command refers to. Up to 15 devices can be specified in a single command line.
dev15:

Functional Details:

The FORM command has three distinct functions dealing with printer paper control. The INCLUDE, REMOVE, and MOUNT keywords denote the functions available. INCLUDE enables new forms to be added to the system and format of these forms can be defined. REMOVE enables currently valid forms to be removed from system availability. MOUNT enables form mounting requests to be issued and verification of alignment of these forms to be requested. The FORM command is valid only when issued from a primary control unit or from a secondary control unit for the output device in question. Issuing a FORM command with no parameters results in forms information for all devices in the spooler environment to be displayed. Whenever a file is placed on the spool queue that requires a form not currently mounted on any device, the message:

REQUEST PENDING FOR FORM: formname

is generated.

Examples:

```
FORM INCLUDE TWO (PL=70, TM=3, BM=4), CHKS (PL=62), BLUE
```

Three forms are defined in this example: TWO, CHKS, and BLUE. TWO is further defined to have a page length of 70, top margin of 3, and bottom margin of 4 lines. CHKS is further defined to have a page length of 62 lines and default top and bottom margins. BLUE has default characteristics for page length, page width, and top and bottom margin sizes.

```
FORM MOUNT=TWO, PRT1:/VE=2
```

The spooler is notified that form type TWO is mounted on device PRT1:. The device will be paused after two pages have been output so that forms alignment can be verified.

GLOBAL

6.15 GLOBAL COMMAND

The GLOBAL command is used to define the global or default pseudo device name or to display the current global pseudo device name.

Format:

GLOBAL pseud dev:

Parameter:

pseud dev: specifies a 1- to 4-character alphanumeric name (first character must be alphabetic) of the pseudo device to be assigned as the global pseudo device.

Functional Details:

The GLOBAL command is valid when issued from the primary control unit when defining a new global pseudo device. Issuing a GLOBAL command with no pseudo device name causes the current global pseudo device name to be displayed. PR: is the default global pseudo device.

Example:

GLOBAL PR1:

The global pseudo device name becomes PR1:.

6.16 GROUP COMMAND

The GROUP command is used to establish a logical relationship between a number of units (terminals or tasks) and a pseudo device, and to assign a default print location for the members of this logical relationship.

Format:

```
GROUP PDEVICE=pseud devl { INCLUDE origin id1 , ..., origin id19 }
                          { REMOVE origin id1 , ..., origin id19 }
```

Parameters:

pseud devl is an already existing pseudo device name.

INCLUDE denotes that the default print location for the members of the group will be the preceding pseudo device.

REMOVE allows a specified member or members of the group to be disassociated from the preceding pseudo device as the default print location.

origin id1: is a unique 1- to 8-character identifier for
 , ..., origin a task or location (terminal).
 id19

Functional Details:

Once an origin is assigned to a group, that origin can no longer assign output to the system global pseudo device. Any assignment to the global system pseudo device is translated by the spooler to an assignment to the group's default pseudo device. A unit may be associated with one group only, meaning it can have only one default print location. The GROUP command may be used without the INCLUDE or REMOVE keywords, the resulting output will detail the units grouped with the pseudo device. If a primary control unit issues the GROUP command alone without any parameters, each pseudo device with its associated group units will be displayed. If a secondary or user control unit issues the GROUP command without any parameters, the units in its group will be displayed.

Examples:

```
GROUP PDEVICE=PR1: INCLUDE CRT1:,TASKABCD
```

The task TASKABCD and the terminal CRT1: are associated with pseudo device PR1:. Therefore, their default print location is the real device associated with pseudo device PR1:.

```
GROUP PDEVICE=PR2: INCLUDE CRT1:,TASKABCD
```

```
GROUP PDEVICE=PR2: REMOVE TASKABCD
```

```
GROUP PDEVICE=PR2: INCLUDE CRT4:
```

In this example, different terminals and tasks under different control tasks are assigned the same default print location, pseudo device PR2:. First, terminal CRT1: and task TASKABCD are associated with PR2:. Then TASKABCD is removed from the group whose default print location is the real device associated with pseudo device PR2:. Finally, terminal CRT4: is included in the group with a default pseudo device of PR2:.

6.17 INPUT COMMAND

The INPUT command is used to specify, remove, or display input spooling devices.

Format:

$$\text{INPUT} \left[\begin{array}{l} \text{CTASK=taskname} [\text{INCLUDE cr1:} [\dots, \text{cr8:}]] \\ \text{REMOVE cr1:} [\dots, \text{cr8:}] \end{array} \right]$$

Parameters:

taskname	specifies the control task that is to receive attention messages generated by the specified input devices.
INCLUDE	assigns the specified devices to input spooling.
REMOVE	removes the specified devices from input spooling.
cr1: ...,cr8:	identifies the input device(s) to be assigned or removed from input spooling.

Functional Details:

If the control task is omitted from the INPUT command, the control task monitoring the device which entered the INPUT command will receive attention messages generated by the INCLUDE input devices. The INPUT command, when used to include or remove input devices, is only valid from the primary control unit. Any user may use the INPUT command without parameters to display the input devices in the spooler environment.

Example:

```
INPUT CTASK=CTASK1 INCLUDE CR1:,CR2:
```

Card input devices CR1: and CR2: are assigned to input spooling. The control task CTASK1 will receive any attention messages pertaining to CR1: and CR2:.

6.18 INTERROGATE COMMAND

The INTERROGATE command is used to display information pertaining to the spool queue or to display the current configuration of the spooler environment.

Format:

```

INTERROGATE {
  QUEUE
  PD pd:
  FORM formname
  FD filename
  CONFIGURATION
}
  
```

Parameters:

QUEUE	<p>provides the following information about user's queued files:</p> <ul style="list-style-type: none"> ● print priority ● filename ● number of copies requested ● pseudo device name ● form name specified ● options (VFC, IMAGE, CHECKPOINT, DELETE, HOLD)
PD	<p>displays the same information as described above for all files on the queue that are assigned to the specified pseudo device.</p>
FORM	<p>displays the same information as described above for all files on the queue that requested the form name specified.</p>
FD	<p>displays the same information as described above for the specific filename specified here.</p>

CONFIGURATION displays the following information about all physical devices on the spooler:

- status
- current form mounted
- currently printing file
- associated pseudo devices

Functional Details:

If no parameters are entered with an INTERROGATE command, all spool queue file information and the spooler device configuration will be displayed.

The output of an INTERROGATE command is dependent upon the authority level of the user who initiated the command. An INTERROGATE command issued from a primary control unit will display information for the system-wide spooler environment. An INTERROGATE command issued from a secondary control unit will display information pertinent to devices associated with that secondary control unit. An INTERROGATE command issued from a user authority level will display information pertinent to that user's account.

Examples:

INTERROGATE QUEUE

This example requests that all of the user's files on the queue be displayed with associated information about each file as described in the QUEUE parameter explanation.

INTERROGATE FORM CHECKS

This example requests that all files on the queue that requested form name CHECKS be displayed.

6.19 PRIMARY COMMAND

The PRIMARY command is used to define the authorization identifier (AID) for the primary control origin and to specify the control task from which commands requiring the primary level of authorization will be accepted.

Format:

PRIMARY ctask/aid

Parameters:

ctask is the control task that is to become a primary control unit.

aid is the identifier name or code that will accompany all communications that require the primary level of authorization. A unit assigned to or monitored by the control task specified in the PRIMARY command that uses the AID specified in the PRIMARY command will have primary control privileges.

Functional Details:

The task .SPL is the default primary control task and INITPAID is the default AID. The primary control unit is the only valid user of the PRIMARY command to set or display the current AID or ctask. The following commands can only be executed from the primary control unit to set or specify:

ATTACH	DETACH	QUEUE
BLOCK	GLOBAL	QUIESCE
CHECK	GROUP	SCTASK
CREATE	INPUT	SECONDARY
CTASK	PRIMARY	SPVOLUME
DELETE	PRIORITY	

These commands are available to other authorization levels to display information only. If the PRIMARY command is entered without a parameter, the current primary task and AID are displayed.

Example:

Primary CTL/ABC

This establishes the AID as ABC. Any command issued from a unit under control task CTL with ABC as the AID has primary authorization privileges.

6.20 PRINT COMMAND

The PRINT command is used to assign a file for output on a printing device.

Format:

```
PRINT fd [,DEVICE=pseud dev:] [,COPIES=n] [,FORM=formname] [ { IMAGE }
                                                    { VFC } ]
[ { NOIMAGE } ] [ { NODELETE } ] [ { NOCHECKPOINT } ] [ , PRIORITY=p ]
[ { NOVFC } ] [ { DELETE } ] [ { CHECKPOINT } ]
```

Parameters:

- fd is the name of the file to be printed. This is a required parameter.
- DEVICE identifies a pseudo device associated with the desired physical output device. If omitted, the spooler determines a default print location.
- COPIES n specifies the number of copies to be output. Must be in the range of 1 to 255. If omitted, one copy will be output.
- FORM specifies that the file is to be output on a special preprinted form. If the specified form name was not previously declared using a FORM command, an error message is generated and the print request is output using the default form name STD.
- IMAGE or VFC specifies that the file is to be output with vertical forms control (VFC) or in image format (IMAGE). With the VFC option, the first character of each 80-byte record is read as a carriage control character. These commands are mutually exclusive.

NOIMAGE or NOVFC	specifies that the VFC option is to be reset and the file is to be output without vertical forms control (NOVFC) or that the IMAGE option is to be reset and the file is to be output with VFC.
NODELETE	resets the DELETE option. The file is not deleted after output. This is the default option for standard user files output by the spooler.
DELETE	specifies the file can be deleted after output. This is the default for files created by the spooler through an ASSIGN or SPOOLFILE command.
NOCHECKPOINT	turns off checkpointing for this file.
CHECKPOINT	turns on checkpointing for this file. This is the default option. The global checkpoint option must be on.
PRIORITY	p is the print priority of this file. p is a number from 1 to 240 with the lower number having a higher print priority.

Functional Details:

If two conflicting parameters are entered in a print request, such as DELETE and NODELETE, the default of the two is executed and an error message is sent.

Example:

```
PRINT M300:REPORT.LST/5,DEV=PRN1:,COP=3,F=FRM1
```

Three copies of the file M300:REPORT.LST in account 5 are to be output on form FRM1 with pseudo device PRN1: specified as the destination.

6.21 PRIORITY COMMAND

The PRIORITY command is used to set the default output priority or display the current default output priority of files on the queue.

Format:

PRIORITY number

Parameter:

number is a number from 1 to 255 that becomes the default priority for the spooling environment.

Functional Details:

Primary control unit authority is required to change the default priority with the PRIORITY command. Secondary control units and user control units can display the current default priority by entering the PRIORITY command with no option. If the PRIORITY command is entered without a new value, the current default output priority is displayed.

Example:

PRIORITY 240

The default priority for print or punch requests becomes 240.

PUNCH

6.22 PUNCH COMMAND

The PUNCH command is used to direct a user file to be output on a card punch device.

Format:

```
PUNCH fd [,DEVICE=pseud dev:] [,COPIES=n][,FORM=form name]
  [ { DELETE } ] [ { CHECKPOINT } ] [, PRIORITY=p]
  [ { NODELETE } ] [ { NOCHECKPOINT } ]
```

Parameters:

fd is the name of the file to be punched. This is a required parameter.

DEVICE= identifies the pseudo device the file is assigned to for output. If omitted, the default punch location is determined by spooler.

COPIES= n is the number of copies to be output. n may range from 1 through 255. If this parameter is omitted, one copy is output.

FORM= specifies that the file be output onto a special preprinted or prepunched card. If the specified form name was not previously declared using a FORM command, an error message is generated and the punch request is output using the default form name STD.

DELETE specifies that the file be deleted after output. This is the default for files created by the spooler through an ASSIGN or SPOOLFILE command.

NODELETE specifies that the file is not to be deleted after output. This is the default option for standard user files output by the spooler.

CHECKPOINT CHECKPOINT turns on checkpointing for the specified punch file. This is the default condition.

NOCHECKPOINT resets the CHECKPOINT option to off.

PRIORITY p is the priority of the punch request.

Functional Details:

If two conflicting parameters are entered in the same PUNCH command, the default parameter is executed and an error message is generated. This command is available to all authority levels in the spooling environment.

Example:

```
PUNCH M300:REPORT.LST/5,DEV=CRD1:,COP=3,DELETE
```

Three copies of the file M300:REPORT.LST in account 5 are to be output on pseudo device CRD1:. The file is to be deleted after output.

QUEUE

6.23 QUEUE COMMAND

The QUEUE command is used to designate the file to be used as the system spool queue or to display the file currently performing this function.

Format:

QUEUE fd

Parameter:

fd specifies the file descriptor of the file to be used as the spool queue in the form VOLN:filename.ext/acct.

Functional Details:

The QUEUE command is generally issued during the START command. If it is omitted from the START command, SPL/32, by default, allocates a file named SPL32.QUE as the spool queue file. SPL32.QUE will reside on the first volume entered in a previous SPVOLUME command or, by default, on the system volume if no SPVOLUME command has been previously entered. If the allocation of the queue file on the system volume fails, the file is allocated on the TEMP volume.

If the user wants to specify a new spool queue while the spooler is active, SPL/32 will finish outputting any file currently being output. Those entries waiting on the spool queue are not merged with the newly specified queue.

A file specified as a queue file must either:

- not currently exist, or
- must have been previously allocated as a queue file by the spooler.

The QUEUE command alone, with no file descriptor parameter, can be used at any time to display the current spool queue filename. Primary control unit authority is required to specify the spool queue file.

Example:

QUEUE M67B:SPLQUEUE.NEW

File M67B:SPLQUEUE.NEW becomes the spool queue. If M67B:SPLQUEUE.NEW does not exist, it will be allocated and spooling will begin. If M67B:SPLQUEUE.NEW already exists, and it was previously allocated as a queue file by the spooler, it will be reused.

SCTASK

6.24 SCTASK COMMAND

The SCTASK command is used to add or remove subcontrol tasks in the spooling environment. These tasks are capable of sending commands to the spooler, and can selectively route spooler responses. Response messages to a subcontrol command can be directed to the control task that the subcontrol task is associated with or to the initiating subcontrol task itself.

Format:

$$\text{SCTASK} \left\{ \begin{array}{l} \text{CTASK=taskname [INCLUDE task1, ..., task7]} \\ \text{REMOVE task1 [, ..., task8]} \end{array} \right\}$$

Parameters:

taskname	denotes the control task that can receive response messages from SPL/32 for the subcontrol tasks denoted in the INCLUDE option of this command.
INCLUDE	is the keyword used to add a subcontrol task(s) to the spooler environment.
REMOVE	is the keyword used to remove a subcontrol task(s) from the spooler environment.
task1 ...,task7 ...,task8	identifies the specific tasks to become subcontrol tasks or to be removed from the spooler environment. Up to seven tasks may be included in one SCTASK command and up to eight tasks may be removed in a single SCTASK command.

Functional Details:

This command, when used to add or remove tasks, is valid only from the primary control unit. All users are able to display a list of subcontrol tasks in the spooler environment by entering the SCTASK command without parameters.

Examples:

SCTASK CTASK=T1 INCLUDE T2,T3

Establishes the tasks T2 and T3 as subcontrol tasks. Responses to commands issued by these tasks can be directed to the control task T1.

SCTASK REMOVE T2

Subcontrol task T2 is removed from the spooler environment.

SECONDARY

6.25 SECONDARY COMMAND

The SECONDARY command is used to define an authorization identifier (AID) for a secondary control unit and/or to display all secondary AIDs in the spooler's environment.

Format:

```
SECONDARY ctask/aid,pd:
```

Parameters:

ctask is the control task which will receive all attention messages for any device associated with the specified pseudo device pd:. This task will be notified of the AID selected in this command.

aid is the authorization identifier which will accompany all commands sent to the spooler with secondary authority.

pd: is the pseudo device to which the specified aid applies.

Functional Details:

This command identifies a control task, a secondary level AID, and a pseudo device. The pseudo device must be created prior to its use in this command. Commands received by the spooler which affect pd: must have the specified secondary AID or a primary AID to be valid. This command is valid from the primary control unit only. When entered without options, all secondary AIDs in the spooling environment will be displayed.

Example:

```
SEC CTASK1/XYZ,PR2:
```

The pseudo device PR2: has the secondary AID XYZ. Any command issued to control devices associated with PR2: must have XYZ as their AID. I/O error messages generated by the physical device associated with PR2: will be directed to CTASK1.

6.26 SPOOLFILE COMMAND

The SPOOLFILE command allows a user to allocate a spool file on behalf of a specified pseudo device and assign that file to a specified lu of the currently selected task. This command makes all spooling options available at a terminal or command substitution system (CSS) level. This command is not processed by SPL/32 as a spooler command. It is handled by the operating system and the multi-terminal monitor (MTM).

Format:

```
SPOOLFILE lu&lul,pseud dev,FORM=formname
    [,{YFC }]
    [,{NOIMAGE }] [,{CHECKPOINT }] [COPIES=n] [,{HOLD }]
    [,{NOVFC }] [,{NOCHECKPOINT }] [,{RELEASE }]
    [BLOCK= blocksize/indexsize] [,{DELETE }] [PRIORITY=p]
    [,{NODELETE }]
```

Parameters:

lu	is a decimal number specifying the logical unit to which the pseudo device is to be assigned.
lul	indicates that lu is to be assigned to the same spool file as lul. lul must be the first lu assigned to the spool file.
pseud dev	is the 1- to 4-character name of a pseudo device. The first character must be alphabetic; the remaining alphanumeric.
FORM=	is a desired preprinted form name that can be specified here. If the form specified was not previously enabled using a FORM command, an error message is sent to the monitoring control or subcontrol task and the request is processed using the default standard form name, STD.

VFC specifies the use of vertical forms control for the assigned lu. When VFC is used, the first character of each record is interpreted as a vertical forms control character. If VFC is not included, there is no vertical forms control for the device assigned to the specified lu.

NOVFC turns the vertical forms control option off for the assigned lu. This is the default option.

COPIES= identifies the number of copies to be output. It must be between 1 and 255 or an error message is sent.

HOLD causes the specified file to remain on the spool queue until a RELEASE request is issued.

RELEASE enables a spool file for output when the lu is closed.

BLOCK specifies the index and/or data block size.

blocksize is a decimal number specifying the physical block size in 256-byte sectors, to be used for buffering and debuffering operations involving the file. The default size is 1 or the value entered using the BLOCK command. If this value exceeds the maximum block size established at sysgen time, an error will be printed when attempting to allocate the file.

indexsize is a decimal number specifying the index block size in 256-byte sectors. The default size is 1 or the value entered using the BLOCK command. Index size cannot exceed the maximum index block size established at sysgen time or an error will occur when attempting to allocate the file.

DELETE the file is deleted after output. This is the default option.

NODELETE the file is not deleted after output.

PRIORITY=p p is the desired print priority. If this option is not specified, the priority of the task the spool file assign is on behalf of becomes the print priority.

Functional Details:

The SPOOLFILE command can be used to make an assignment to a pseudo device from the terminal or CSS level. If two conflicting parameters are entered in a single SPOOLFILE command, such as DELETE and NODELETE, the second parameter is executed and an error message is generated. The SPOOLFILE command is primarily for users operating in an MTM environment.

Example:

```
SPOOLFILE 4,pdl:,VFC,DELETE
```

This example causes a spool file to be allocated for pseudo device pdl: and assigns that file to logical unit 4 of the current task. Vertical forms control has been specified for the specified lu and the DELETE option has been selected which means the file will be deleted after output.

SPVOLUME

6.27 SPVOLUME COMMAND

The SPVOLUME command is used to designate new spool volumes, remove current spool volumes, or to display the spool volumes currently in use on the system.

Format:

$$\text{SPVOLUME } \left[\left\{ \begin{array}{l} \text{INCLUDE volume1 [, ..., volume20]} \\ \text{REMOVE volume1 [, ..., volume20]} \end{array} \right\} \right]$$

Parameters:

INCLUDE	is used to designate new spool volumes.
REMOVE	is used to designate spool volumes to be disassociated with the spooling function.
volume1 , ..., volume20	designates the name(s) of the disk volume(s) to which the INCLUDE or REMOVE parameters refer. The name is one to four characters long. The first character must be alphabetic and the remaining alphanumeric.

Functional Details:

The SPVOLUME command's function is dependent on the user's authorization level. Primary control unit authorization is necessary to add or remove disk volumes in the spooling environment. Primary, secondary, and user control units are able to DISPLAY what disk volumes are currently being used for the spooling function by entering the SPVOLUME command without options.

Files being spooled (output and input) without a specific volume designated are allocated by the spooler on the spool volumes on a round-robin basis so that no particular spool volume gets a disproportionate number of spool files. This applies to systems utilizing multiple spool volumes.

Examples:

SPVOLUME INCLUDE M300,M301

Disk volumes M300 and M301 are now available for the allocation of spool files.

SPVOLUME REMOVE M300

Disk volume M300 is no longer available for the allocation of spool files.

SPVOLUME

Displays the spooled disks currently on the system.

TRAILER

6.28 TRAILER COMMAND

The TRAILER command is used to specify the number of trailing identification page(s) to be output with each file on a specified pseudo or real device and, to format the contents of this identification page(s) to suit users needs.

Format:

TRAILER [pseudo dev:] [real dev:] [{ DOUBLE } SINGLE } NONE] [, CHARACTER=character] [,fd]

Parameters:

- pseudo dev: specifies the pseudo device or real device to which the TRAILER command applies.
- real dev: specifies the pseudo device or real device to which the TRAILER command applies.
- DOUBLE specifies that two trailer pages are to be produced for each file output on the specified device.
- SINGLE specifies that one trailer page is to be produced for each file output on the specified device.
- NONE specifies that no trailer page is to be produced for each file output on the specified device. This option is useful for very slow printing devices, card punches, or printers formatting very large files.
- CHARACTER character specifies a single ASCII character that the spooler will use to print the contents of the TRAILER page. The default is the letter O.
- fd is the file descriptor of an information file that contains any combination of the following six keywords.

FD formats the filename.
AC formats the user account number.
TASKNAME formats the task name.
DATE formats the date.
TIME formats the time of day.
LOGO='text' formats a 1- to 12-character user-defined text record. The test record must be contained within the quotation marks.

The user has the option to select up to five of the above keywords and include them in the file. These keywords signify the information to be printed on the TRAILER pages.

Functional Details:

The default condition is a one page trailer for a pseudo or real device with the following information:

- Filename
- Account number
- Task name
- Date
- Time

If the TRAILER command is directed to a pseudo device, the specifications in the command apply to all real devices associated with the pseudo device. If a real device is associated with two or more pseudo devices, the latest TRAILER command entered for any of the pseudo devices will apply to the real device. This command is only valid from a primary control unit or a secondary control unit for pseudo device associated with the specified real device.

In addition to the 132-column TRAILER output, an information block is output for each file. This information block contains the following information.

- Full filename
- User account number
- Date and time the file was placed on the spool queue
- Date and time the file was output
- The physical output device
- Pseudo device name

Example:

```
TRAILER PRT1:,D, CHARACTER=A
```

Output from device PRT1: will be printed with two trailer pages containing the default information set. The contents of the TRAILER page will be printed using the letter A.

APPENDIX A
SPL/32 COMMAND SUMMARY

ATTACH PDEVICE=dev RDEVICE=dev1 [, dev2 ... , dev19]

BANNER [pseudo dev:] [{ DOUBLE }
[real dev:] [{ SINGLE }] [, CHARACTER=character [, fd]
[NONE }

BLOCK blksize/indexsize

CHANGE fd DEVICE=pseud dev: [, COPIES=n] [, FORM=formname] [, PRIORITY=n]

[{ IMAGE }] [{ NOIMAGE }] { DELETE } { CHECKPOINT } { HOLD }
[{ VFC }] [{ NOVFC }] { NODELETE } { NOCHECKPOINT } { RELEASE }

CHECKPOINT [{ ON, N }]
[{ OFF }]

CREATE pseud dev1: [, ...pseud dev20:]

CTASK [{ INCLUDE task1 , ... , task10 }]
[{ REMOVE task2 , ... , task10 }]

DELETE [{ pseud dev1: , ... , pseud dev20: }]
[{ real dev1: , ... , real dev20: }]

DETACH PDEVICE=pseud dev RDEVICE=real dev1: [, ...real dev19:]

DEVICE {

 pseudo device name [{ RESTRICT }]

 real device name [{ SUSPEND { IMMEDIATE } }]

 [{ DELAYED }]

 [{ ELINE=n }]

 [{ EPAGE=n }]

 [{ BLINE=n }]

 [{ BPAGE=n }]

 [{ REWIND }]

 [{ CONTINUE }]
 }

END [{ DELAYED }]

 [{ IMMEDIATE }]

FLUSH [{ fd [DELETE] }]

 [{ real device: }]

FORM {

 INCLUDE frm1 [{ PLENGTH=p }] [..., frm4] [{ PLENGTH=p }]

 [{ BMARGIN=b }] [{ BMARGIN=b }]

 [{ TMARGIN=t }] [{ TMARGIN=t }]

 [{ PWIDTH=pw }] [{ PWIDTH=pw }]

 REMOVE frm1 [, frm2] [, ..., frm5]

 MOUNT frm1 [VERIFY=n , dev1:] [, ..., dev15:]
 }

GLOBAL pseud dev:

GROUP PDEVICE=pseud dev1 {

 INCLUDE origin id1 , ..., origin id19

 REMOVE origin id1 , ..., origin id19
 }

INPUT [{ CTASK=taskname [INCLUDE cr1:] [..., cr8:] }]

 [{ REMOVE cr1: [, ..., cr8:] }]

INTERROGATE $\left\{ \begin{array}{l} \text{QUEUE} \\ \text{PD pd:} \\ \text{FORM formname} \\ \text{FD filename} \\ \text{CONFIGURATION} \end{array} \right\}$

PRIMARY ctask/aid

PRINT fd [,DEVICE=pseud dev:] [,COPIES=n] [,FORM=formname] $\left[\left\{ \begin{array}{l} \text{IMAGE} \\ \text{VFC} \end{array} \right\} \right]$
 $\left[\left\{ \begin{array}{l} \text{NOIMAGE} \\ \text{NOVFC} \end{array} \right\} \right] \left[\left\{ \begin{array}{l} \text{NODELETE} \\ \text{DELETE} \end{array} \right\} \right] \left[\left\{ \begin{array}{l} \text{NOCHECKPOINT} \\ \text{CHECKPOINT} \end{array} \right\} \right] [, PRIORITY=p]$

PRIORITY number

PUNCH fd [,DEVICE=pseud dev:] [,COPIES=n][,FORM=form name]
 $\left[\left\{ \begin{array}{l} \text{DELETE} \\ \text{NODELETE} \end{array} \right\} \right] \left[\left\{ \begin{array}{l} \text{CHECKPOINT} \\ \text{NOCHECKPOINT} \end{array} \right\} \right] [, PRIORITY=p]$

QUEUE fd

SCTASK $\left\{ \begin{array}{l} \text{CTASK=taskname}[\text{INCLUDE task1 ,task7}] \\ \text{REMOVE task1 [, ...,task8]} \end{array} \right\}$

SECONDARY ctask/aid,pd:

SPOOLFILE lu&lul,pseud dev,FORM=formname $\left[\left\{ \begin{array}{l} \text{VFC} \\ \text{IMAGE} \end{array} \right\} \right]$
 $\left[\left\{ \begin{array}{l} \text{NOIMAGE} \\ \text{NOVFC} \end{array} \right\} \right] \left[\left\{ \begin{array}{l} \text{CHECKPOINT} \\ \text{NOCHECKPOINT} \end{array} \right\} \right] [, COPIES=n] \left[\left\{ \begin{array}{l} \text{HOLD} \\ \text{RELEASE} \end{array} \right\} \right]$
 $[\text{,BLOCK= blocksize/indexsize}] \left[\left\{ \begin{array}{l} \text{DELETE} \\ \text{NODELETE} \end{array} \right\} \right] [, PRIORITY=p]$

SPVOLUME $\left[\begin{array}{l} \text{INCLUDE volume1 [, ..., volume20]} \\ \text{REMOVE volume1 [, ..., volume20]} \end{array} \right]$

TRAILER $\left[\begin{array}{l} \text{pseudo dev:} \\ \text{real dev:} \end{array} \right] \left[\begin{array}{l} \text{DOUBLE} \\ \text{SINGLE} \\ \text{NONE} \end{array} \right] \left[\text{, CHARACTER=character} \right] \left[\text{,fd} \right]$

APPENDIX B
SPL/32 COMMAND MESSAGE SUMMARY

CANNOT COMMUNICATE WITH CTASK: ctask

The specified control task is not present in the spooling environment or has gone to end of task.

DEVICE ACTIVE: device name given

A command requiring the specified device to be in a suspended state was issued to an active device.

DEVICE INACTIVE: device name given

A command requiring the specified device to be in an active state was issued to an inactive device.

DEVICE NOT INCLUDED: device name

The specified device is not in the spooler environment.

DUPLICATE DEVICE: dev:

The specified device name already exists in the spooler environment.

DUPLICATE TASK NAME: task name

The specified task already exists in the spooling environment.

DUPLICATE UNIT NAME: unit id

The specified unit name already exists and may not be used again.

DUPLICATE VOLUME: volume name

The specified volume is already assigned for the allocation of spool files.

FILE DESCRIPTOR ERROR: fd

The file descriptor was syntactically incorrect.

FILE NOT FOUND: fd

The specified file was not found on the spool queue or in the disk directory. Check the filename.

FORM NOT INCLUDED: formname

The specified form has not been declared a valid form prior to use in a PRINT/PUNCH request.

INVALID AUTHORIZATION-ID: authorization identifier

The authorization ID specified in a PRIMARY or SECONDARY command is invalid.

INVALID AUTHORIZATION LEVEL: authorization level

The command attempted is not available to the specified authorization level.

INVALID COMMAND: command

Command is syntactically incorrect or is not a valid SPL/32 command.

INVALID CONTROL CARD

A card deck with an incorrect control card was read.

INVALID DATA BLOCK SIZE: blocksize

The data block size specified exceeds the maximum established at sysgen time.

INVALID DEVICE: dev:

The specified device is not assigned to the spooler.

INVALID FORM: form

The specified form is not a valid form name.

INVALID INDEX BLOCK SIZE: i

The index block size specified exceeds the maximum established at sysgen time.

INVALID KEYWORD: keyword

Keyword is syntactically incorrect or is not valid with the associated command.

INVALID PARAMETER: parameter

The specified parameter is not valid with the associated command.

INVALID PRIORITY: priority

The requested priority exceeds the maximum set at Link time, or the requested priority is outside the range of 10 to 249.

INVALID SEPARATOR: separator

The specified separator is invalid or is syntactically incorrect.

INVALID TASK: taskname

The specified taskname does not exist.

INVALID VOLUME: vol:

The specified volume is syntactically incorrect.

I/O TASK WAS CANCELLED: I/O taskname

The specified I/O task has been cancelled.

MISSING CONTROL CARD

A card deck without an initial control card was read.

MISSING PARAMETER

A required parameter is missing from the command.

NO AVAILABLE LUS

There are no logical units available for use.

NO SPOOL VOLUMES

There are no spool volumes specified in the spooling environment.

NO SUBCONTROL TASKS

There are no subcontrol tasks in the spooling environment.

NO INPUT DEVICES

There are no spooled card readers in the spooling environment.

NO QUEUE ENTRIES

There are presently no files in the spool queue.

NO GROUPS

There are currently no groups specified in the spooling environment.

PRIMARY CTASK WAS CANCELLED: taskname

The primary control task has been cancelled.

PARAMETER CONFLICT: parameter1, parameter2

Two mutually exclusive parameters are present.

PRIVILEGE ERROR

The access privilege mnemonic was syntactically incorrect.

TASK NOT INCLUDED: taskname

The specified task was not included in the spooling environment.

UNIT NOT INCLUDED: unit id

The specified unit was not included in the grouping request of a GROUP command.

VERIFY COMPLETED ON: device name

A request to verify forms alignment has been completed on the specified device.

VOLUME NOT INCLUDED: volume name

The specified volume was not included as a spool volume.

VOLUME NOT IN SYSTEM: vol:

The specified volume is not present in the system and therefore cannot be assigned as a spool volume.

APPENDIX C
SPL/32 VS. SPOOLER COMMAND CORRELATION

C.1 INTRODUCTION

The following table correlates the SPL/32 command set with the Spooler command set as a guide for converting command substitution system (CSS) procedures.

SPL/32 COMMAND	CORRESPONDING SPOOLER COMMAND	COMMAND FUNCTION
ATTACH	START parameter	Associates pseudo and physical devices
BANNER	OPTION	Defines the Banner page options
BLOCK	*	Sets/displays index/data block sizes
CHANGE	*	Changes output parameters of spool file
CHECKPOINT	*	Turns on/off/displays check-pointing
CREATE	*	Creates pseudo devices
CTASK	*	Sets/displays control tasks
DELETE	DELETE	Removes pseudo or physical devices from the spooling environment
DETACH	DELETE	Dissociates physical and pseudo devices
DEVICE	OPTION	Controls/displays device status
-RESTRICT	-PROTECT	Prevents use of device as default
-FREE	-NOPROTECT	Enables use of device as default
-FLINE	-FORWARD	Forward spaces a file n lines
-FPAGE	-FORWARD	Forward spaces a file n pages
-BLINE	-BACKWARD	Backward spaces a file n lines
-BPAGE	-BACKWARD	Backward spaces a file n pages
-REWIND	-REWIND	Rewinds a file
-CONTINUE	-CONTINUE	Restarts a device if paused
-SUSPEND	-SUSPEND	Pauses a device
END	*	Ends spooler task
FLUSH	CANCEL	Stops a file from being output

SPL/32 COMMAND	CORRESPONDING SPOOLER COMMAND	COMMAND FUNCTION
FLUSH	PURGE	Removes a file from the queue
FORM	*	Creates/displays valid form names
GLOBAL	*	Specifies/displays global pseudo device
GROUP	*	Sets/displays default print locations for logically grouped sets of terminals
INPUT	START CARD	Defines input spool devices
INTERROGATE	INQUIRE	Displays spool queue information
PRIMARY	*	Defines the primary authorization identifier
PRINT	PRINT	Prints a spool file
PRIORITY	*	Sets/displays default priority
PUNCH	PUNCH	Punches a spool file
QUEUE	*	Specifies file to be used as the spool queue
SCTASK	*	Sets/displays subcontrol tasks
SECONDARY	*	Defines secondary level authorization identifier
SPOOLFILE	*	Makes SPL/32 features available level at the MTM terminal or CSS level
SPVOLUME	*	Sets/displays spool volumes
TRAILER	*	Specifies trailer page options

* indicates no command of similar function was available with spooler

| - connection indicates parameters of the SPL/32 DEVICE command (SPL/32) or the spooler OPTION command (Spooler)

APPENDIX D MESSAGE HANDLING

D.1 INTRODUCTION

This chapter provides a discussion of the messages and message handling procedures that SPL/32 uses to communicate with users and tasks within the spooling environment. The format of each type of spooler message is presented. The role of the uniform spooler control interface (USCI) is discussed. This chapter is presented primarily for the benefit of applications programmers who want to construct tasks that can function as control tasks within the spooling environment. These tasks must be able to communicate with the spooler and receive messages from the spooler in response.

D.1.1 Message Overview

Communication between the spooler and tasks consists of a series of dialogues. The task that issues the first message in a dialogue is the initiator. The initiator of a dialogue must assign a dialogue identifier that uniquely identifies where the dialogue initiated from. The dialogue identifier is then used to locate the initiator of the dialogue within the control tasks environment. In general, the structure of the messages used by SPL/32 is designed to allow control tasks, such as MTM, to access the contents of the message as necessary. Messages sent to the spooler from tasks/terminals consist of a 24-byte message buffer for SVC6 Send Data functions, a 44-byte SPL/32 message header, and an 80-byte ASCII record. This format allows commands to be entered and read directly from one 80-byte line on a terminal. Messages sent to tasks/terminals also use this format. This format allows the ASCII data to be displayed directly to a terminal or printer.

D.2 THE UNIFORM SPOOLER CONTROL INTERFACE (USCI)

The uniform spooler control interface (USCI) is the interface between control and subcontrol tasks and the spooler. The USCI accepts spooler commands and service requests from multiple concurrent tasks and if valid, submits them for execution. The USCI is designed to minimize spooler dependent code within tasks that use the spooler and to allow the spooler to be independent of the internal design of the tasks it communicates with.

D.2.1 Establishing Communication With the Spooler

Only control and subcontrol tasks can communicate directly with the spooler. In the case of control tasks, communication with the spooler is established when the spooler receives a request to designate a task as a control task. The request to create the control task is rejected and communication is not established if the task does not exist. If the task does exist, it is designated a control task and communication is established. In the case of subcontrol tasks, communication with the spooler is established when the subcontrol task is included. Both control and subcontrol tasks must be resident in the spooling environment and able to receive traps generated by an SVC6 send data request, in order to communicate with the spooler.

If no control tasks are included in the spooling environment, the task .SPL is initially the default primary control task to allow communication between SPL/32 and the system console. It is cancelled when another primary control task is specified and accepted.

D.2.2 Losing or Terminating Communication With a Task

There are two conditions that can result in a loss of communication between a task and the spooler.

- A control task goes to end of task (normal termination).
- The spooler is unable to send a message to a control task (abnormal termination).

D.2.2.1 Normal Termination of a Control Task

The spooler intercepts any SVC3 issued by control tasks to ascertain when they no longer exist. If a control task goes to end of task while the spooler is running, the following actions are taken according to the task's level of authorization:

- If the primary control task goes to end of task, the default primary control task .SPL, is loaded and given primary authorization. A message is sent to the system console indicating that the primary control task was cancelled. Subsequent attention messages generated by pseudo devices associated with the ending primary control task are directed to .SPL. Subcontrol tasks associated with the previous primary control task are removed from the spooling environment.
- If a control task with secondary authorization goes to end of task, subsequent attention messages generated by pseudo devices associated with the ending task are sent to the primary control task. All subcontrol tasks associated with the ending task are removed from the spooling environment.

D.2.3 Response Routing (Spooler to Task)

Spooler responses to control tasks and subcontrol tasks are routed in the following manner:

- Spooler responses to directives issued by control tasks are routed directly back to the issuing control task.
- Spooler responses to directives issued by subcontrol tasks can be selectively routed:
 1. The response can be routed back to the issuing subcontrol task.
 2. The response can be routed back to the control task monitoring the issuing subcontrol task.

Message routing is specified by the routing flag byte (byte 64) in the message header preceding the message from the subcontrol task. A value of X'FF' in the routing flag byte causes responses to be routed back to the subcontrol task. A value of X'00' in the routing flag bytes causes the message to be routed to the control task monitoring the subcontrol task. The spooler reads the routing flag byte for each message sent. This allows the message routing to be changed on a message by message basis, if so desired.

D.2.4 Dialogue Termination

The manner in which a dialogue between a control or subcontrol task and the spooler is terminated depends on the initiating task in the dialogue and the type of message sent to initiate the dialogue.

If a dialogue is initiated by a control task sending a command message, then it is terminated by the spooler with an exception or service message.

An exception response message is sent if there is an error in the command, or if no service message is required and no error occurred.

A service message is sent if there is no error and a service message is required.

A dialogue is not terminated until all message buffers have been received.

The mnemonic request initiates a dialogue and the mnemonic response terminates it.

The attention message initiates and terminates a dialogue.

The authority message is terminated when the control task returns an acceptance message in the case when the authority message is requesting acceptance, or the authority message terminates itself when it notifies the originating control task of whether the transfer succeeded or failed.

D.3 MESSAGE HEADER

The message header is a 44-byte record that precedes each message received or issued by SPL/32. It contains identification information that notifies the spooler or a task about the initiator of the message and provides information about the type of message that follows. The format of a message header record is presented in Figure D-1.

0 (00)	MESSAGE BUFFER FOR SVC6 SEND DATA FUNCTION			
4 (04)				
8 (08)				
12 (C)				
16 (10)				
20 (14)				
24 (18)	DIALOGUE IDENTIFIER			
28 (1C)	INITIATING TASK NAME			
32 (20)				
36 (24)	ORIGIN IDENTIFIER (OID)			
40 (28)				
44 (2C)	AUTHORITY IDENTIFIER (AID)			
48 (30)				
52 (34)	MESSAGE TYPE ID	53(35) MESSAGE SUBTYPE CODE	54(36) COMMAND ID	55(37) COMMAND FUNCTION
56 (38)	USER ACCOUNT		58 (4A) GROUP ACCOUNT	
60 (3C)	DEFAULT VOLUME			
64 (40)	ROUTING FLAG	65(41) RESERVED		

Figure D-1 SPL/32 Message Header Record Format

Fields:

MESSAGE BUFFER FOR
SVC 6 SEND DATA FUNCTION

the format of this field and a discussion of its use is presented in the OS/32 Supervisor Call (SVC) Reference Manual.

DIALOGUE ID

contains a unique identifier within the task that initiated the dialogue. The contents of this field are used to identify the message being responded to.

INITIATING TASK

contains the name of the task that initiated the dialogue.

ORIGIN IDENTIFIER

contains the origin identifier of a terminal or task associated with a group. This field is used to determine the default print location and must match an origin identifier (OID) entered in a GROUP command before the default printing location will be used. (Left justified with trailing blanks.)

AUTHORITY IDENTIFIER

contains the authority identifier (AID) terminal/task issuing the message. The AID is used by the spooler to determine whether or not an origin location has the authority to perform the command function requested. (Left justified with trailing blanks.)

MESSAGE TYPE ID

contains the setting that identifies the general type of message being sent. The possible settings are:

From task to SPL/32:

X'0' = Command message
X'2' = Acceptance response message
X'3' = Request for mnemonic table

From SPL/32 to ctasks or sctasks:

X'02' = Acceptance response message
X'10' = Service message
X'11' = Exception response message
X'12' = Attention message
X'14' = Mnemonic message
X'15' = Authority message

MESSAGE SUBTYPE CODE

contains the type of attention or exception message present.

COMMAND IDENTIFIER

contains the numeric identifier of a prepared command. (The setting for each command is presented in Appendix E.) When ASCII command format is being used, this field contains zeros.

For an exception response message, this field is the numeric identifier of the command that generated the exception response.

COMMAND FUNCTION

contains the setting that identifies what function of the command is to be performed. Codes identify a display or set function request and identify the various option settings for some commands. The command function settings are presented in Appendix E.

USER ACCOUNT

contains the account number of a request if the account is not explicitly specified in a file descriptor. It also provides a control task with a means of restricting access to files. If the account number specified in a request does not match the user's account, group account, or system account, the request is invalid. In an MTM environment, account numbers are used to identify the P, G, and S account class specifiers. This field must be set to X'FFFF' if not used.

DEFAULT VOLUME

contains the default volume name for each request. If the volume is not specified in a request, the default volume is used. This field must be set to X'FFFFFFFF' if not used.

ROUTING FLAG

specifies the route that responses from the spooler to a subcontrol task take. A value of X'FF' causes responses to be returned to the originating subcontrol task. A value of X'00' causes the message to be returned to the control task monitoring the originating subcontrol task.

D.4 MESSAGE FORMATS

The following sections detail the format of the various messages that SPL/32 uses to perform the spooling function. The function of each message is described in Sections 5.2 through 5.3. In Sections D.4 through D.4.11, the following convention is used:

Fields within the SVC6 send data message buffer and the message header are not redefined unless they change for the message being presented. The contents of these fields are discussed in Section D.3.

D.4.1 Task to SPL/32 Message Formats

There are three message types that control tasks can issue to the spooler:

- Command messages
- Acceptance response messages
- Request for mnemonic table messages

These formats are presented in the following sections.

D.4.1.1 The Command Message

The message format for command messages is presented in Figure D-2. This format is used for both ASCII and preparsed commands.

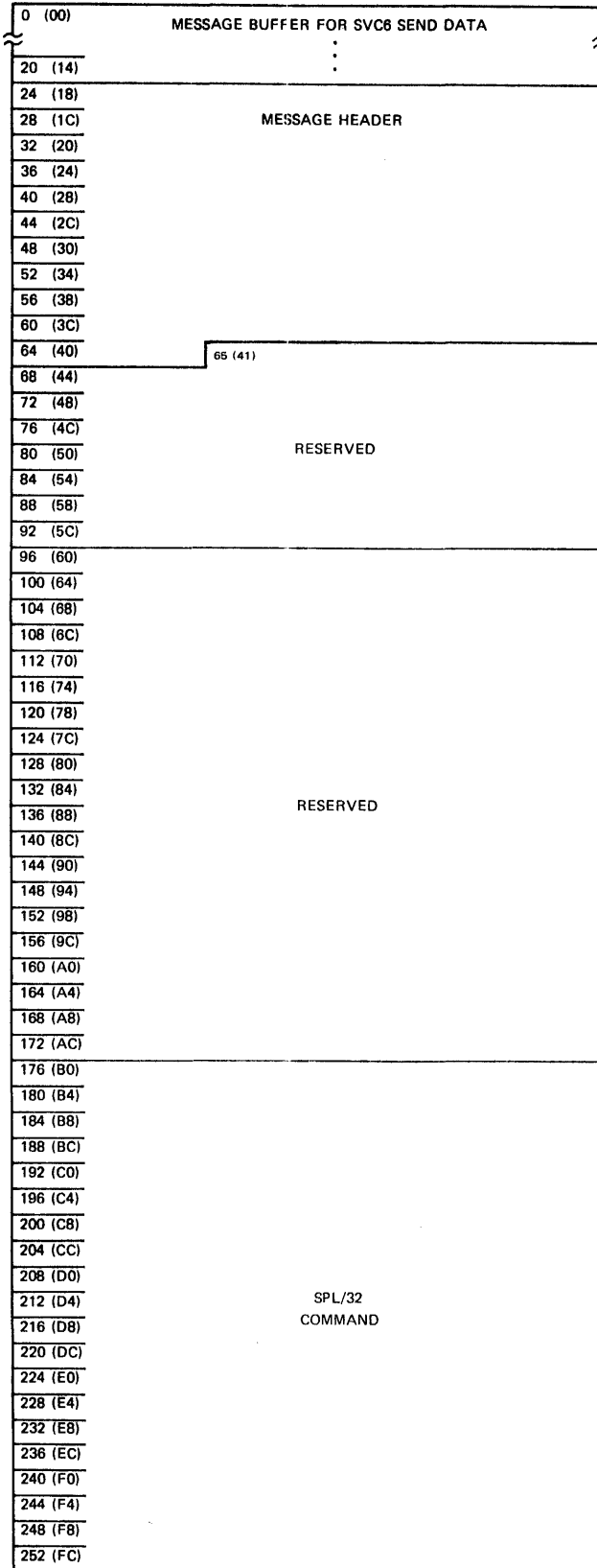


Figure D-2 Command Message Format

Fields:

COMMAND contains a preparsed or ASCII format spooler command.

Discussion:

If a command message is issued from a control task, then SPL/32 issues a response to the command. If the command is accepted, then the message subtype code contains zero. If there is an exception error, the message subtype code indicates the problem.

Command messages from subcontrol tasks are returned to the subcontrol task (if the routing flag so indicates) in the same manner.

If the command message is from a subcontrol task and the response is to be returned to a control task (as indicated by the routing flag), then only responses required to service the message are sent to the control task. Exception responses with error code 0 are not sent to the control task.

D.4.1.2 Acceptance Response Message Format

The format of an acceptance response message is presented in Figure D-3.

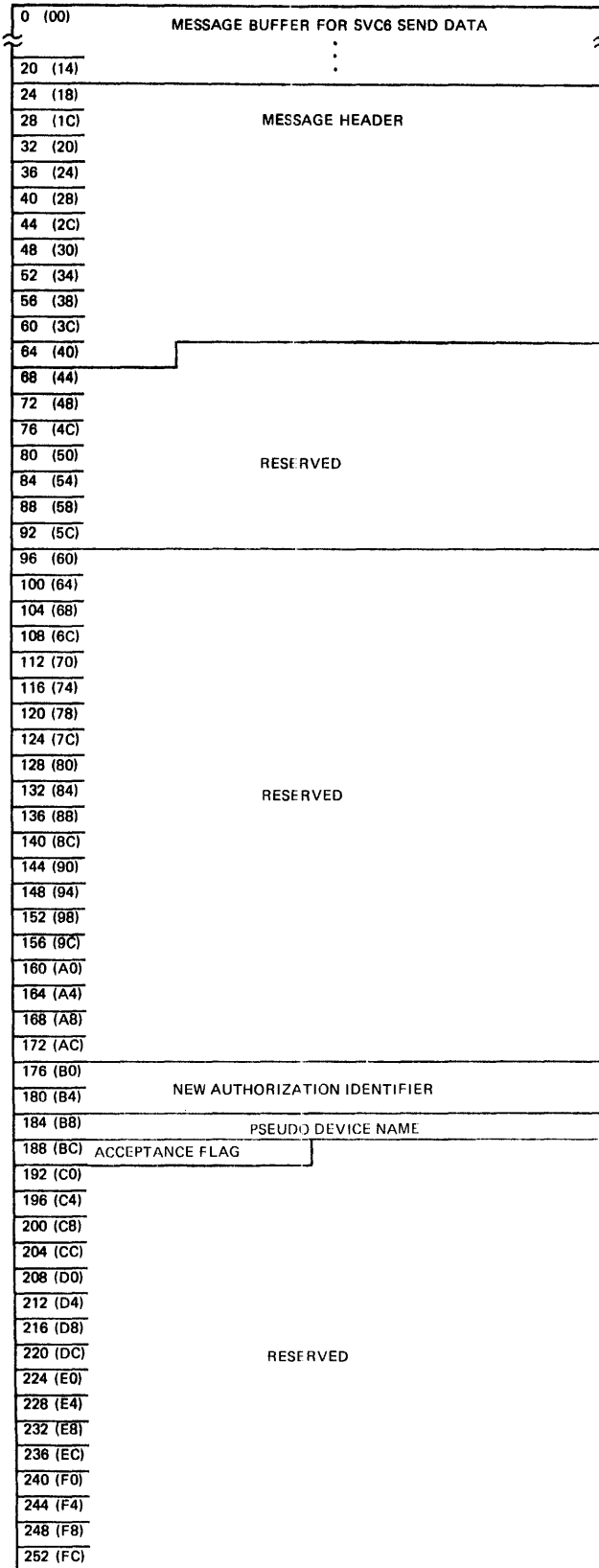


Figure D-3 Acceptance Response Message Record Format.

Fields:

MESSAGE HEADER	information in the fields of this portion of the message is identical to the information contained in the corresponding fields of the authority message being responded to with the exception of the MESSAGE TYPE ID field which is set to X'2'.
NEW AUTHORIZATION IDENTIFIER	contains the new authorization identifier carried in the corresponding field of the authority message being responded to.
PSEUDO DEVICE NAME	contains the name of a pseudo device. This field is used only when the AID specified in the NEW authorization identifier field pertains to a secondary authorization location. The pseudo device specified is the pseudo device to which the NEW authorization identifier pertains.
ACCEPTANCE FLAG	contains one of the following settings: X'0000' = AID accepted X'FFFF' = AID rejected

D.4.1.3 Request for Mnemonic Table Message

The format of a request for mnemonic table message is presented in Figure D-4.

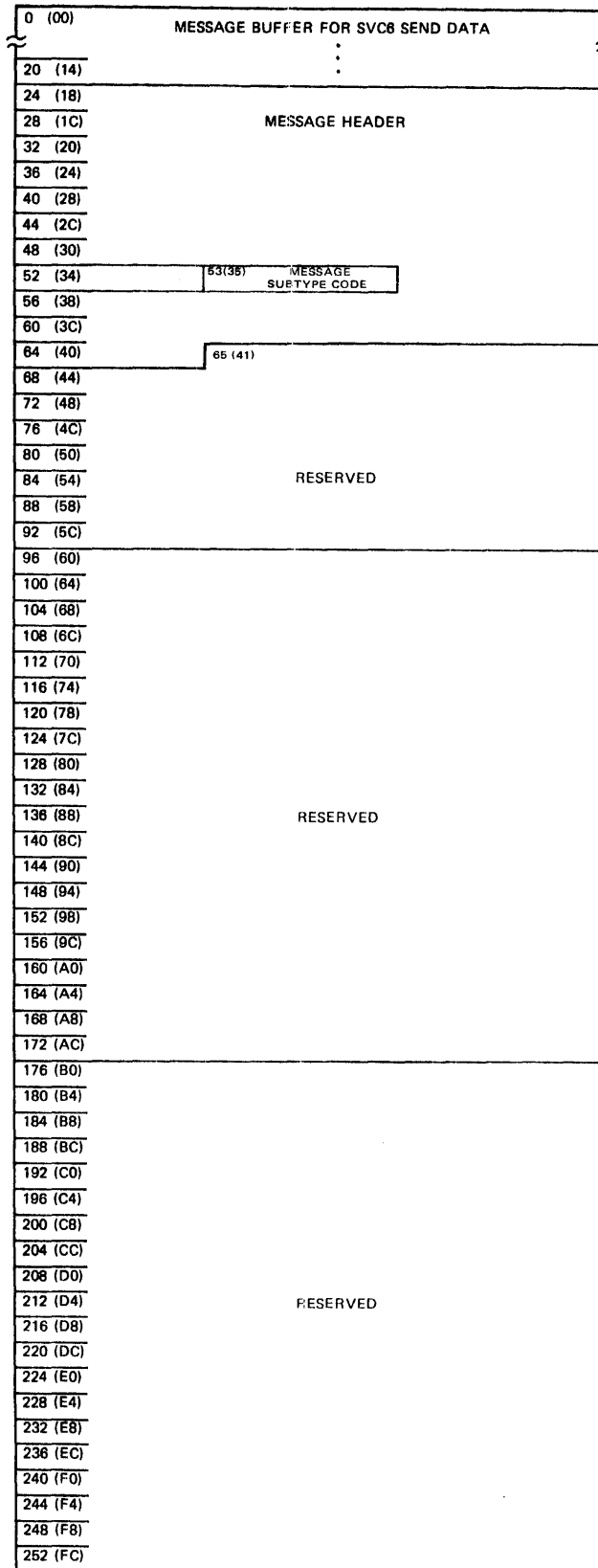


Figure D-4 Request for Mnemonic Table Record Format

Fields:

MESSAGE HEADER the message subtype code must be set to X'3'.

Discussion:

Once a request for a mnemonic table message has been received, the spooler sends the mnemonic table to the requesting control task using the mnemonic message format presented in Section D.4.6.

D.4.2 SPL/32 to Task Message Formats

There are five types of messages that SPL/32 can issue to a control task:

- Service response messages
- Exception response messages
- Authority messages
- Attention messages
- Mnemonic messages
- Acceptance response messages

D.4.2.1 Service Response Message Format

The service response message is used to supply information, from the spooler to a requestor, that pertains to the current spooler environment. The amount of information supplied depends on the size and configuration of the spooling environment, and is therefore variable. To accommodate this variability, service response messages consist of a chain of buffers. Each chain consists of at least one 256-byte buffer. Therefore, the requestor must have at least 256 bytes of buffer space to receive service response messages from the spooler. Each message is sent with one SVC6 SEND DATA function. If the receiving task does not have enough buffer space to accommodate an entire message, the message is broken into a series of single buffer messages. Each single buffer uses an SVC6 SEND DATA function. The last single buffer sent in the series is indicated by an X'FFFF' in the last buffer field of the service response message buffer structure.

D.4.2.2 Buffer Set Format

A service response message uses two different buffer formats, an initial buffer and a continuation buffer. When a message is sent and the user has enough buffer space to accommodate the entire message, both buffer formats are used. If the user does not have enough buffer space to accommodate the entire message, the message is broken down into a series of single buffer messages. Each of these messages uses the initial buffer format. The format of an initial buffer is presented in Figure D-5.

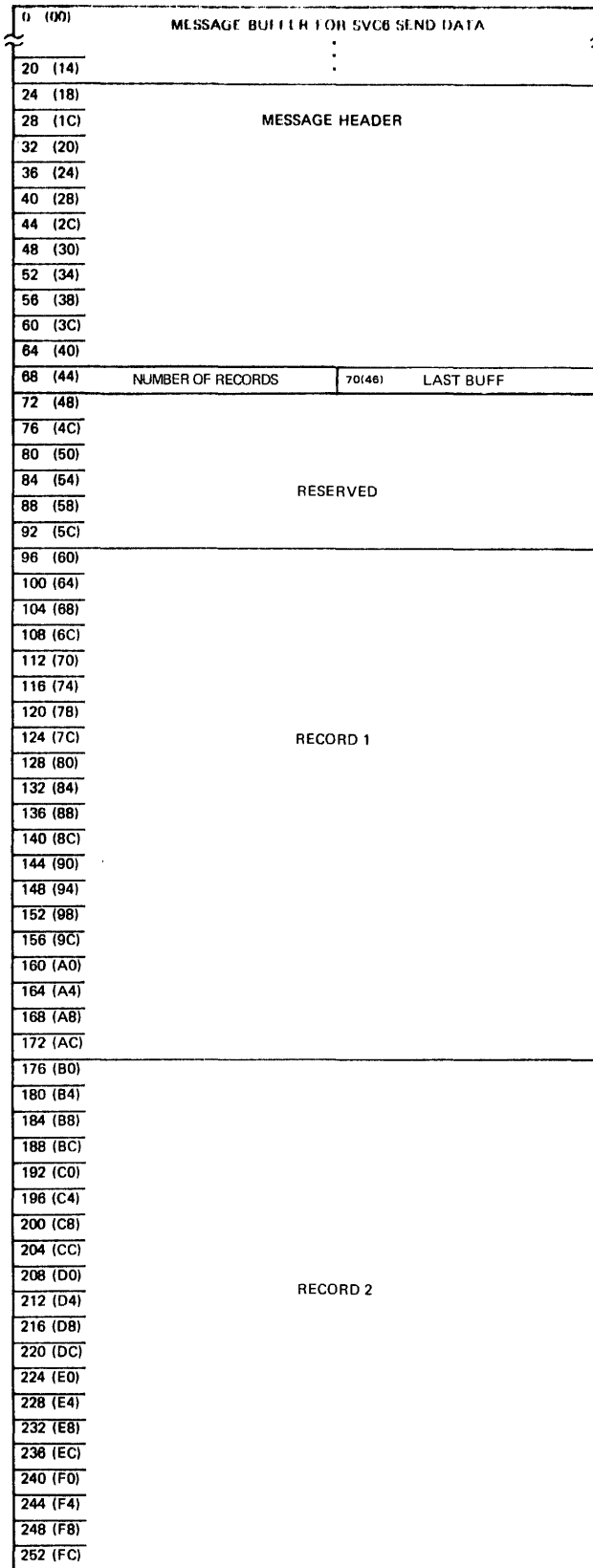


Figure D-5 Service Response Message - Initial Buffer Format

Fields:

MESSAGE HEADER	the type of service records contained in a buffer are identified by the command identifier and command function bytes of the message header.
NUMBER OF RECORDS	contains an X'1' or X'2' to indicate the number of records used in the buffer.
LAST BUFF	contains a setting (X'FFFF') that indicates whether the buffer is the last in a chain of buffers. If the buffer is not the last buffer, this field contains zeros.
RECORD 1	contains the spooler response to a command request. The response is in ASCII and variable in length depending on the spooler environment. When the full 80 bytes are not used, the remainder of the field is blank filled.
RECORD 2	contains the spooler response to a command request. The response is in ASCII format and is variable in length depending on the spooler environment. When the full 80 bytes are not used, the remainder of the field is blank filled.

The continuation buffer in a buffer set is structured as presented in Figure D-6. The message buffer for the SVC6 send data function in the first buffer of the set applies to all buffers of the set and is not repeated in subsequent buffers.

0	SVC6 CONTINUATION	
4	HEADER	
8	NUMBER OF RECORDS	10(A) LAST BUFF
12	RESERVED	
16	RECORD 3	
20 (14)		
24 (18)		
28 (1C)		
32 (20)		
36 (24)		
40 (28)		
44 (2C)		
48 (30)		
52 (34)		
56 (38)		
60 (3C)		
64 (40)		
68 (44)		
72 (48)		
76 (4C)		
80 (50)		
84 (54)		
88 (58)		
92 (5C)		
96 (60)		
100 (64)		
104 (68)	RECORD 5	
108 (6C)		
112 (70)		
116 (74)		
120 (78)		
124 (7C)		
128 (80)		
132 (84)		
136 (88)		
140 (8C)		
144 (90)		
148 (94)		
152 (98)		
156 (9C)		
160 (A0)		
164 (A4)		
168 (A8)		
172 (AC)		
176 (B0)		
180 (B4)		
184 (B8)		
188 (BC)		
192 (C0)		
196 (C4)		
200 (C8)		
204 (CC)		
208 (D0)		
212 (D4)		
216 (D8)		
220 (DC)		
224 (E0)		
228 (E4)		
232 (E8)		
236 (EC)		
240 (F0)		
244 (F4)		
248 (F8)		
252 (FC)		

Figure D-6 Service Response Message - Continuation Buffer Format

Fields:

- NUMBER OF RECORDS contains an X'1', X'2', or X'3' to indicate the number of records used in the buffer.
- RECORD 3 contains the spooler response to a command request. The response is in ASCII and variable in length depending on the spooling environment. When the full 80 bytes are not used, the remainder of the field is blank filled.
- RECORD 4 contains the spooler response to a command request. The response is in ASCII and is variable in length depending on the spooling environment. When the full 80 bytes are not used, the remainder of the field is blank filled.
- RECORD 5 contains the spooler response to a command request. The response is in ASCII and is variable in length depending on the spooling environment. When the full 80 bytes are not used, the remainder of the field is blank filled.

The spooler determines the number of buffers and buffer sets required by a service response message. If a message consists of one to two 80-byte records, only the initial buffer is required, the records are placed in the RECORD 1 and RECORD 2 fields of the initial buffer. If the message consists of more than two 80-byte records, the continuation buffer may be used depending on the amount of receive buffer space the user has allocated. If the receive buffer space allocated cannot accommodate the entire service message, SPL/32 will use multiple initial buffers to send the message. The continuation buffer is not used. If the user does have enough receive buffer space allocated to accommodate the entire message, then the continuation buffer will be used to transport records 3 through 5, 6 through 8, etc. Service response messages that are variable in length are noted as such in the description of the message in the following sections so that the reader is aware that multiple buffer sets are used to transport these messages.

D.4.3 Individual Command Service Response Messages

The following sections present the individual record format for each spooler command service response message. The messages are presented in one 256-byte buffer format. The reader should note that messages consisting of more than two 80-byte records will be formatted according to the procedure described in the previous section based on the available receive buffer space allocated.

D.4.3.1 ATTACH Command Service Record

The format of the ATTACH command service record is presented in Figure D-7. This message is issued by the spooler when a pseudo device specified in an ATTACH command was not previously created.

0 (00)		MESSAGE BUFFER FOR SVC6 SEND DATA	
20 (14)	:		
24 (18)	:		
28 (1C)	MESSAGE HEADER		
32 (20)			
36 (24)			
40 (28)			
44 (2C)			
48 (30)			
52 (34)	54(36) COMMAND ID	55(37) COMMAND FUNCTION	
56 (38)			
60 (3C)			
64 (40)			
68 (44)	NUMBER OF RECORDS	70(46) LAST BUFF	
72 (48)			
76 (4C)			
80 (50)			
84 (54)	RESERVED		
88 (58)			
92 (5C)			
96 (60)			
100 (64)			
104 (68)			
108 (6C)			
112 (70)			
116 (74)			
120 (78)			
124 (7C)			
128 (80)	BLANK FILLED		
132 (84)			
136 (88)			
140 (8C)			
144 (90)			
148 (94)			
152 (98)			
156 (9C)			
160 (A0)			
164 (A4)			
168 (A8)			
172 (AC)			
176 (B0)			
180 (B4)			
184 (B8)			
188 (BC)			
192 (C0)	"PSEUDO DEVICE CREATED IMPLICITLY"		
196 (C4)			
200 (C8)			
204 (CC)			
208 (D0)	210(D2) PSEUDO		
212 (D4)	DEVICE NAME		
216 (D8)			
220 (DC)			
224 (E0)			
228 (E4)			
232 (E8)			
236 (EC)			
240 (F0)			
244 (F4)			
248 (F8)			
252 (FC)			

Figure D-7 ATTACH Command Service Message Record Format

Fields:

COMMAND ID	contains an X'1' to identify the ATTACH command.
COMMAND FUNCTION	contains an X'3' to identify the SPECIFY function of the ATTACH command.
NUMBER OF RECORDS	contains an X'1' or X'2' to indicate the number of records used in the buffer.
LAST BUFF	contains a setting (X'FFFF') that indicates that the buffer is the last in a chain of buffers. If the buffer is not the last buffer in a chain, the field contains zeros.
PSEUDO...IMPLICITLY	is a text message that precedes the new pseudo device name.
PSEUDO DEVICE NAME	contains the name of the implicitly created pseudo device.

D.4.3.2 BLOCK Command Service Record

The format of the BLOCK command service record is presented in Figure D-8. This message is issued in response to a request to display the data and index blocksize values.

0 (00)	MESSAGE BUFFER FOR SVC6 SEND DATA	
20 (14)	:	
24 (18)	:	
28 (1C)	MESSAGE HEADER	
32 (20)		
36 (24)		
40 (28)		
44 (2C)		
48 (30)		
52 (34)	54(36) COMMAND ID	55(37) COMMAND FUNCTION
56 (38)		
60 (3C)		
64 (40)		
68 (44)	NUMBER OF RECORDS	70(46) LAST BUFF
72 (48)		
76 (4C)		
80 (50)		
84 (54)	RESERVED	
88 (58)		
92 (5C)		
96 (60)		
100 (64)	"DATA BLOCK SIZE = "	
104 (68)		
108 (6C)		
112 (70)	114(72) NUMBER OF	
116 (74)	BLOCKS	118(76)
120 (78)	BLANKS	
124 (7C)		
128 (80)	"INDEX BLOCK SIZE = "	
132 (84)		
136 (88)		
140 (8C)	142(8E) NUMBER OF	
144 (90)	BLOCKS	146(92)
148 (94)		
152 (98)		
156 (9C)		
160 (A0)	BLANKS	
164 (A4)		
168 (A8)		
172 (AC)		
176 (B0)		
180 (B4)		
184 (B8)		
188 (BC)		
192 (C0)		
196 (C4)		
200 (C8)		
204 (CC)		
208 (D0)		
212 (D4)	BLANKS	
216 (D8)		
220 (DC)		
224 (E0)		
228 (E4)		
232 (E8)		
236 (EC)		
240 (F0)		
244 (F4)		
248 (F8)		
252 (FC)		

Figure D-8 BLOCK Command Service Message Record Format

Fields:

COMMAND ID	contains an X'3' to identify the BLOCK command.
COMMAND FUNCTION	contains an X'0' to identify the DISPLAY function of the BLOCK command.
NUMBER OF RECORDS	contains an X'1' or X'2' to indicate the number of records used in the buffer.
LAST BUFF	contains a setting (X'FFFF') that indicates that the buffer is the last in a chain of buffers. If the buffer is not the last buffer in a chain, the field contains zeros.
DATA BLOCK SIZE =	is a text header that precedes the data blocking factor.
NUMBER OF BLOCKS	is the current data blocking factor.
INDEX BLOCK SIZE =	is a text header that precedes the index blocking factor.
NUMBER OF BLOCKS	is the current index blocking factor.

D.4.3.3 CHECKPOINT Command Service Record

The format of the CHECKPOINT command service record is presented in Figure D-9. This message is issued in response to displaying the current checkpoint value.

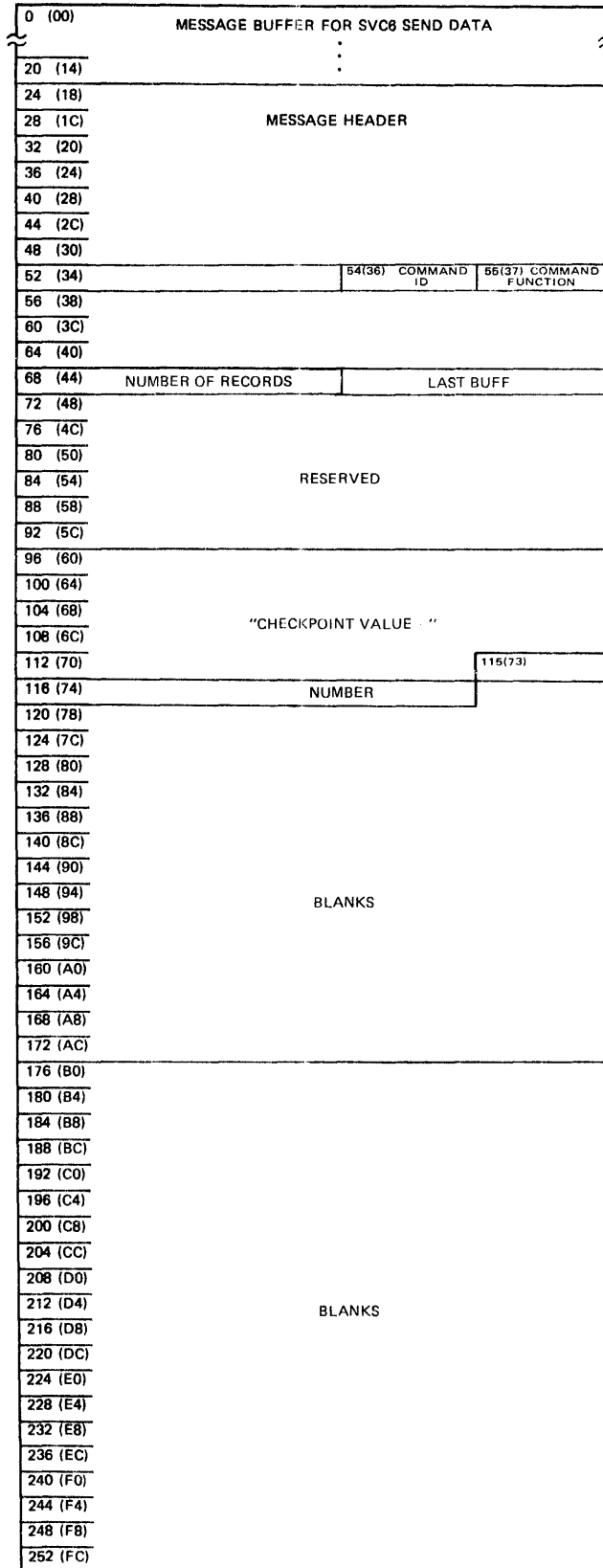


Figure D-9 CHECKPOINT Command Service Message Record Format

Fields:

COMMAND ID	contains an X'5' to identify the CHECKPOINT command.
COMMAND FUNCTION	contains an X'0' to identify the DISPLAY function of the CHECKPOINT command.
NUMBER OF RECORDS	contains an X'1' or X'2' to indicate the number of records used in the buffer.
LAST BUFF	contains a setting (X'FFFF') that indicates that the buffer is the last in a chain of buffers. If the buffer is not the last buffer in a chain, the field contains zeros.
CHECKPOINT VALUE =	is a text message that precedes the checkpoint value.
NUMBER	is the current CHECKPOINT value.

D.4.3.4 CTASK Command Service Record

The format of the CTASK command service record is presented in Figure D-10. The CTASK command service record has a variable number of entries depending on the number of control tasks present in the spooling environment.

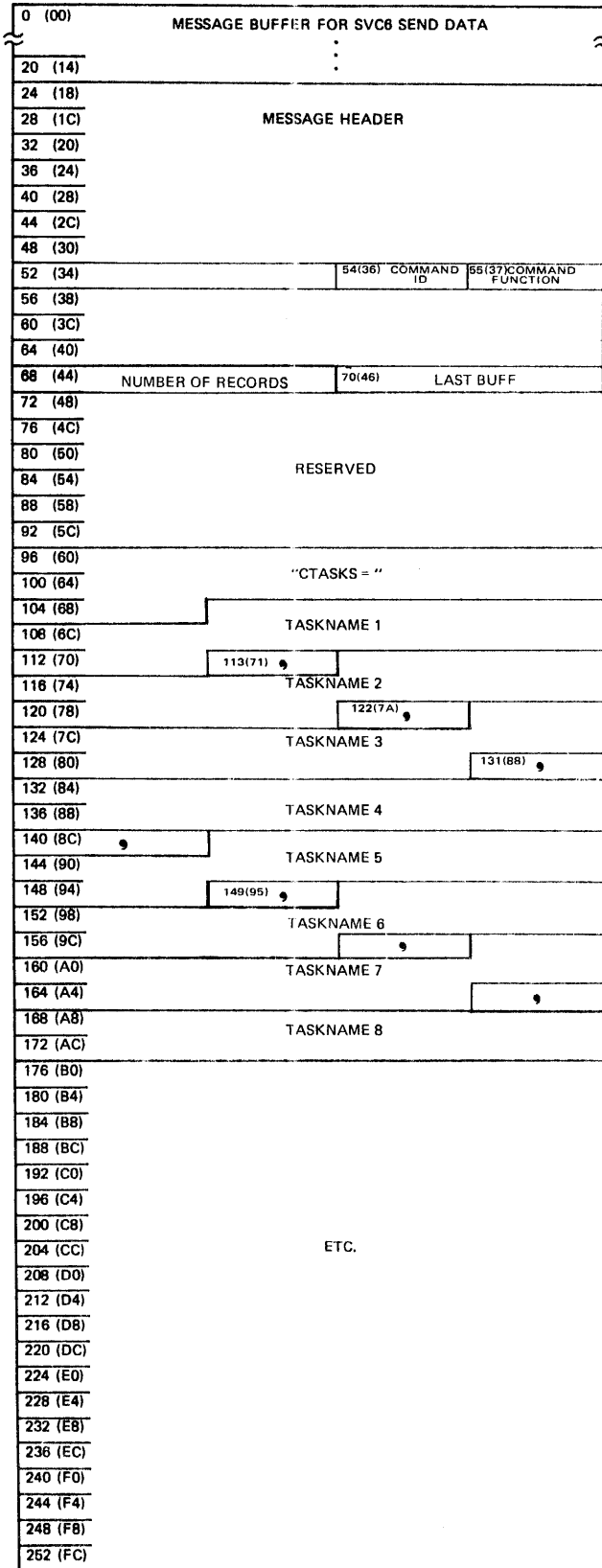


Figure D-10 CTASK Command Service Message Record Format

Fields:

COMMAND ID	contains an X'7' to identify the CTASK command.
COMMAND FUNCTION	contains an X'0' to identify the DISPLAY function of the CTASK command.
NUMBER OF RECORDS	contains an X'1' or X'2' to indicate the number of records used in the buffer.
LAST BUFF	contains a setting (X'FFFF') that indicates that the buffer is the last in a chain of buffers. If the buffer is not the last buffer in a chain, the field contains zeros.
CTASKS=	is a text header that precedes the list of control tasks.
TASKNAME 1	is the name of a control task in the spooler environment.
,	separates the displayed control task names.

D.4.3.5 FORM Command Service Record

The format of the FORM command service record is presented in Figure D-11. This message has a variable number of entries depending on the number of forms in the spooling environment.

0 (00)	MESSAGE BUFFER FOR SVC6 SEND DATA	
20 (14)	:	
24 (18)	:	
28 (1C)	MESSAGE HEADER	
32 (20)		
36 (24)		
40 (28)		
44 (2C)		
48 (30)		
52 (34)	64(36) COMMAND ID	56(37) COMMAND FUNCTION
56 (38)		
60 (3C)		
64 (40)		
68 (44)	NUMBER OF RECORDS	70(46) LAST BUFF
72 (48)		
76 (4C)		
80 (50)		
84 (54)	RESERVED	
88 (58)		
92 (5C)		
96 (60)		
100 (64)	"FORMS = "	
104 (68)		
108 (6C)		
112 (70)		
116 (74)	FORMNAME 1	
120 (78)		
124 (7C)		
128 (80)		
132 (84)		
136 (88)		
140 (8C)	FORMNAME 2	
144 (90)		
148 (94)		
152 (98)		
156 (9C)		
160 (A0)		
164 (A4)	FORMNAME 3	
168 (A8)		
172 (AC)		
176 (B0)		
180 (B4)		
184 (B8)		
188 (BC)		
192 (C0)		
196 (C4)		
200 (C8)		
204 (CC)		
208 (D0)		
212 (D4)		
216 (D8)	ETC.	
220 (DC)		
224 (E0)		
228 (E4)		
232 (E8)		
236 (EC)		
240 (F0)		
244 (F4)		
248 (F8)		
252 (FC)		

Figure D-11 FORM Command Service Message Record Format

Fields:

COMMAND ID	contains an X'B' to identify the FORM command.
COMMAND FUNCTION	contains an X'0' to identify the DISPLAY function of the FORM command.
NUMBER OF RECORDS	contains an X'1' or X'2' to indicate the number of records used in the buffer.
LAST BUFF	contains a setting (X'FFFF') that indicates that the buffer is the last in a chain of buffers. If the buffer is not the last buffer in a chain, the field contains zeros.
FORMS=	is a text header that precedes the form names to be displayed.
FORM NAME 1	is a valid form name.
,	separates each displayed form name.

D.4.3.6 GLOBAL Command Service Record

The format of the GLOBAL command service record is presented in Figure D-12.

0 (00)		MESSAGE BUFFER FOR SVC8 SEND DATA	
20 (14)			⋮
24 (18)			
28 (1C)	MESSAGE HEADER		
32 (20)			
36 (24)			
40 (28)			
44 (2C)			
48 (30)			
52 (34)		54(36) COMMAND ID	55(37) COMMAND FUNCTION
56 (38)			
60 (3C)			
64 (40)			
68 (44)	NUMBER OF RECORDS	70(46)	LAST BUFF
72 (48)			
76 (4C)			
80 (50)	RESERVED		
84 (54)			
88 (58)			
92 (5C)			
96 (60)			
100 (64)			
104 (68)	"GLOBAL PSEUDO DEVICE = "		
108 (6C)			
112 (70)			
116 (74)			PSEUDO
120 (78)	DEVICE NAME		⋮
124 (7C)			
128 (80)			
132 (84)			
136 (88)			
140 (8C)	BLANKS		
144 (90)			
148 (94)			
152 (98)			
156 (9C)			
160 (A0)			
164 (A4)			
168 (A8)			
172 (AC)			
176 (B0)			
180 (B4)			
184 (B8)			
188 (BC)			
192 (C0)			
196 (C4)			
200 (C8)			
204 (CC)			
208 (D0)			
212 (D4)	BLANKS		
216 (D8)			
220 (DC)			
224 (E0)			
228 (E4)			
232 (E8)			
236 (EC)			
240 (F0)			
244 (F4)			
248 (F8)			
252 (FC)			

Figure D-12 GLOBAL Command Service Message Record Format

Fields:

COMMAND ID	contains an X'C' to identify the GLOBAL command.
COMMAND FUNCTION	contains an X'0' to identify the DISPLAY function of the GLOBAL command.
NUMBER OF RECORDS	contains an X'1' or X'2' to indicate the number of records used in the buffer.
LAST BUFF	contains a setting (X'FFFF') that indicates that the buffer is the last in a chain of buffers. If the buffer is not the last buffer in a chain, the field contains zeros.
GLOBAL PSEUDO DEVICE	is a text message.
NAME	is the current global pseudo device.

D.4.3.7 GROUP Command Service Record

The GROUP command service record is variable in length and consists of two records. The first record contains the names of the default pseudo device and the monitor control task for a group. The second record contains the names of the group members with a maximum of seven members per record. The formats of the two records are presented in Figure D-13.

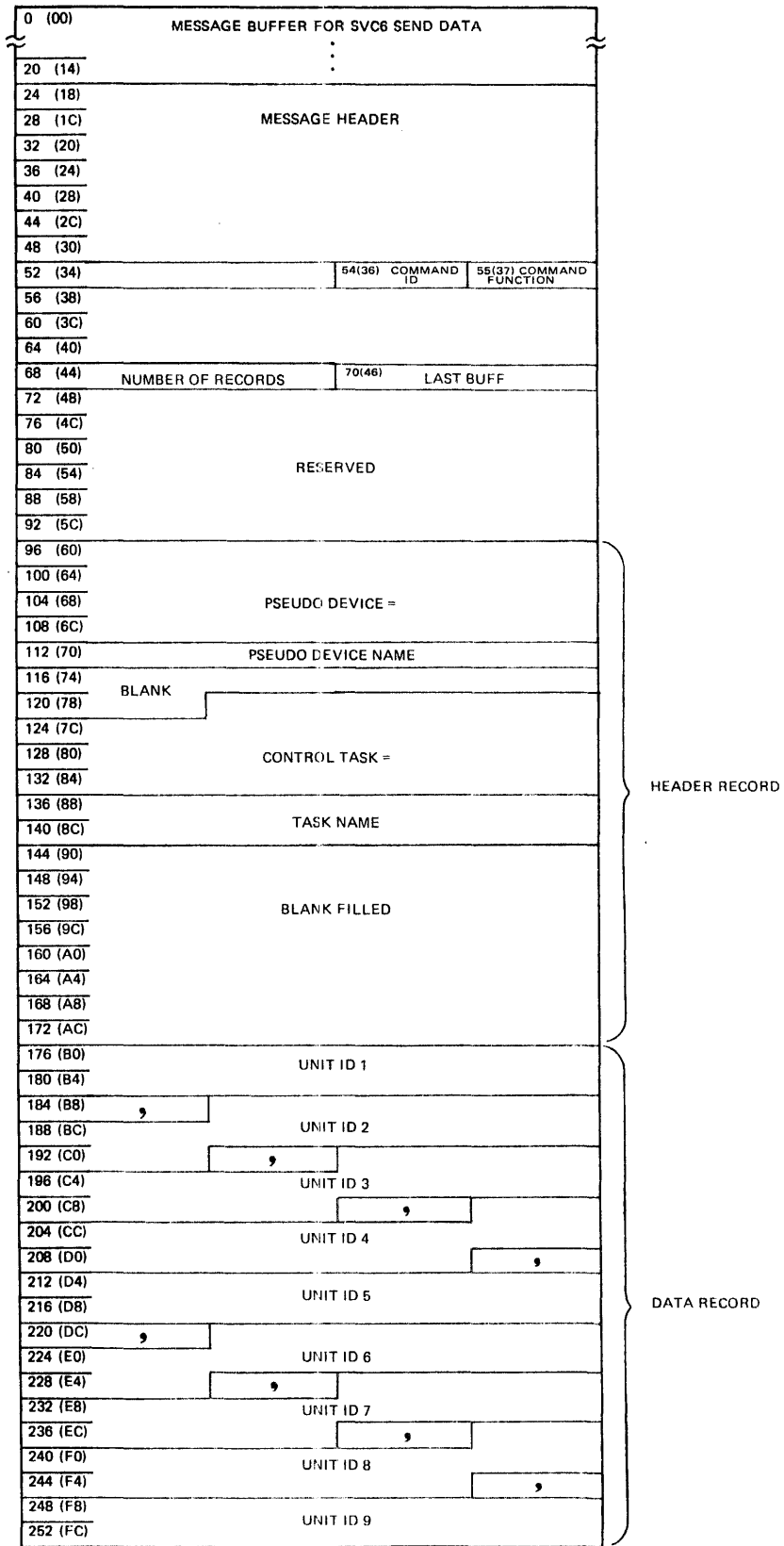


Figure D-13 GROUP Command Service Message - Header Record and Data Record Formats

Fields:

CHAIN INDICATOR	indicates the position the buffer set occupies in a chain: first, last, or middle.
COMMAND ID	contains an X'D' to identify the GROUP command.
COMMAND FUNCTION	contains an X'0' to identify the DISPLAY function of the GROUP command.
NUMBER OF RECORDS	contains an X'1' or X'2' to indicate the number of records used in the buffer.
LAST BUFF	contains a setting (X'FFFF') that indicates that the buffer is the last in a chain of buffers. If the buffer is not the last buffer in a chain, the field contains zeros.
PSEUDO DEVICE=	is a text header.
DEVICE NAME	is the name of the default pseudo device for the group.
CONTROL TASK=	is a text header.
TASKNAME	is the name of the monitor control task.
UNIT ID	is the name of a terminal/task in the group.

D. 4. 3. 8 INPUT Command Service Record

The structure of the INPUT command service record is presented in Figure D-14. This record can have a variable number of entries depending on the number of input devices in the spooling environment.

0 (00)	MESSAGE BUFFER FOR SVC6 SEND DATA		
20 (14)	:		
24 (18)	:		
28 (1C)	MESSAGE HEADER		
32 (20)			
36 (24)			
40 (28)			
44 (2C)			
48 (30)			
52 (34)	54(36) COMMAND ID	55(37) COMMAND FUNCTION	
56 (38)			
60 (3C)			
64 (40)			
68 (44)	NUMBER OF RECORDS	70(46) LAST BUFF	
72 (48)			
76 (4C)	RESERVED		
80 (50)			
84 (54)			
88 (58)			
92 (5C)			
96 (60)	INPUT DEVICES =		
100 (64)			
104 (68)			
108 (6C)			
112 (70)	DEVICE 1		:
116 (74)	,	DEVICE	
120 (78)	2	:	,
124 (7C)	DEVICE 3		:
128 (80)	,	DEVICE	
132 (84)	4	:	,
136 (88)	DEVICE 5		:
140 (8C)	,	DEVICE	
144 (90)	6	:	,
148 (94)	DEVICE 7		:
152 (98)	,	DEVICE	
156 (9C)	8	:	,
160 (A0)	DEVICE 9		:
164 (A4)	,	DEVICE	
168 (A8)	10	:	,
172 (AC)	DEVICE 11		:
176 (B0)			
180 (B4)	ETC.		
184 (B8)			
188 (BC)			
192 (C0)			
196 (C4)			
200 (C8)			
204 (CC)			
208 (D0)			
212 (D4)			
216 (D8)			
220 (DC)			
224 (E0)			
228 (E4)			
232 (E8)			
236 (EC)			
240 (F0)			
244 (F4)			
248 (F8)			
252 (FC)			

Figure D-14 INPUT Command Service Message Record Format

Fields:

CHAIN INDICATOR	indicates the position the buffer set occupies in a chain: first, last, or middle.
COMMAND ID	contains an X'E' to identify the INPUT command
COMMAND FUNCTION	contains an X'0' to identify the DISPLAY function of the INPUT command.
NUMBER OF RECORDS	contains an X'1' or X'2' to indicate the number of records used in the buffer.
LAST BUFF	contains a setting (X'FFFF') that indicates that the buffer is the last in a chain of buffers. If the buffer is not the last buffer in a chain, the field contains zeros.
INPUT DEVICES=	is a text header.
DEVICE 1	is the name of an input device.
:,	separates each displayed input device.

D.4.3.9 INTERROGATE Command Service Record

The format of an INTERROGATE command service record depends upon the type of INTERROGATE command entered. The format of an INTERROGATE QUEUE, FD, PD, or FORM request is presented in Figure D-15 and D-16. The first record contains the column headers and the second record contains the spool queue entries. This message is variable in length depending upon the number of entries on the spool queue.

0 (00)	MESSAGE BUFFER FOR SVC6 SEND DATA		
20 (14)	:		
24 (18)	:		
28 (1C)	MESSAGE HEADER		
32 (20)			
36 (24)			
40 (28)			
44 (2C)			
48 (30)			
52 (34)		54(36) COMMAND ID	58(37) COMMAND FUNCTION
56 (38)			
60 (3C)			
64 (40)			
68 (44)	NUMBER OF RECORDS	70(46)	LAST BUFF
72 (48)			
76 (4C)			
80 (50)	RESERVED		
84 (54)			
88 (58)			
92 (5C)			
96 (60)	PSEUDO DEVICE		
100 (64)			
104 (68)			
108 (6C)	109(6D)	FILE NAME	
112 (70)			
116 (74)	118(76)		
120 (78)	FORMNAME		
124 (7C)			127(7F)
128 (80)	COPY		131(83)
132 (84)			135(87)
136 (88)	PRIORITY		
140 (8C)	OPTIONS		
144 (90)	146(92)		
148 (94)			
152 (98)			
156 (9C)			
160 (A0)	BLANKS		
164 (A4)			
168 (A8)			
172 (AC)			
176 (B0)	PSEUDO DEVICE NAME		
180 (B4)	:	181(85)	BLANKS 183(87)
184 (B8)			
188 (BC)			
192 (C0)	FILE DESCRIPTOR		
196 (C4)			
200 (C8)			
204 (CC)	206(CE)	BLANKS	
208 (D0)			
212 (D4)			
216 (D8)			
220 (DC)	FORMNAME		
224 (E0)			
228 (E4)			231(E7)
232 (E8)	BLANKS		235(FB) COPIES NUMBER
236 (EC)	BLANKS		
240 (F0)	242(F2) PRIORITY NUMBER		243(F3)
244 (F4)	BLANKS		
248 (F8)	HOLD OR RELEASE	249(F9)	250(FA) IMAGE OR FORMAT
252 (FC)	DELETE OR NODELETE	253(FD)	254(FE) VT OR NOVTC
			255(F7) BLANK

RECORD 1
HEADER RECORDRECORD 2
DISPLAY RECORD

Figure D-15 INTERROGATE Command Service Message Header and Display Record Format

Fields:

COMMAND ID	contains an X'F' to identify the INTERROGATE command.
COMMAND FUNCTION	contains one of the following to identify the parameter being interrogated: <ul style="list-style-type: none">● X'10' = QUEUE● X'11' = PD● X'12' = FORM● X'13' = FD
NUMBER OF RECORDS	contains an X'1' or X'2' to indicate the number of records used in the buffer.
LAST BUFF	contains a setting (X'FFFF') that indicates that the buffer is the last in a chain of buffers. If the buffer is not the last buffer in a chain, the field contains zeros.
PSEUDO DEVICE	is a text header.
FILENAME	is a text header.
FORM NAME	is a text header.
COPY	is a text header.
PRIORITY	is a text header.
OPTIONS	is a text header.
PSEUDO DEVICE NAME	is the name of the pseudo device the file is assigned to.

FILE DESCRIPTOR	is the file descriptor of the spool file.
FORM NAME	is the name of the form that the file requires for output.
NUMBER OF COPIES	is the number of copies of the file to be output.
PRIORITY	is the output priority of the file.
HOLD, IMAGE, DELETE, VFC	are optional output parameters of the file.

The format of an INTERROGATE CONFIGURATION command consists of two record types: a header record and a spooler device entry record. The response to this command can be variable in length depending upon the number of devices in the spooler environment. The format of the header record and the device entry record is presented in Figure D-16.

4046

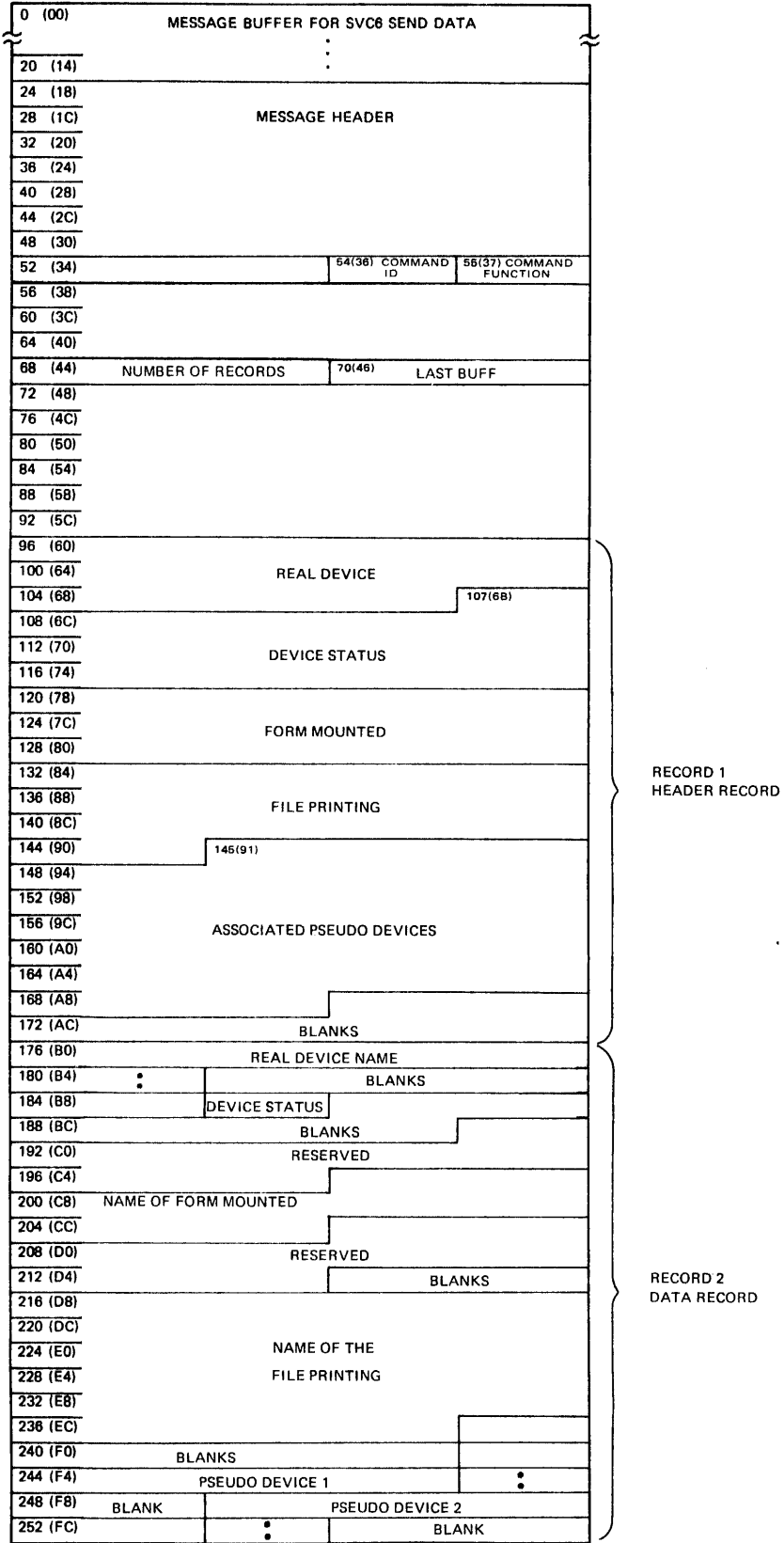


Figure D-16 INTERROGATE Configuration Command Service Message Record Format

Fields:

COMMAND ID	contains an X'F' to identify the INTERROGATE command.
COMMAND FUNCTION	contains one of the following to identify the parameter being interrogated: <ul style="list-style-type: none">● X'10' = QUEUE● X'11' = PD● X'12' = FORM● X'13' = FD
NUMBER OF RECORDS	contains an X'1' or X'2' to indicate the number of records used in the buffer.
LAST BUFF	contains a setting (X'FFFF') that indicates that the buffer is the last in a chain of buffers. If the buffer is not the last buffer in a chain, the field contains zeros.
REAL DEVICE	is a text header.
DEVICE STATUS	is a text header.
FORM MOUNTED	is a text header.
FILE PRINTING	is a text header.
ASSOCIATED PSEUDO DEVICES	is a text header.
REAL DEVICE NAME	is the name of a real device in the spooler environment.
DEVICE STATUS	is the status of the preceding real device.
NAME OF FORM MOUNTED	is the name of the form mounted on the preceding real device.
NAME OF FILE PRINTING	is the name of the file printing on the preceding real device.
PSEUDO DEVICE1,...,N	are the names on pseudo devices associated with the preceding real devices.

D.4.3.10 PRIMARY Command Service Record

The format of the PRIMARY command service record is presented in Figure D-17.

0 (00)	MESSAGE BUFFER FOR SVC8 SEND DATA	
20 (14)	:	
24 (18)		
28 (1C)	MESSAGE HEADER	
32 (20)		
36 (24)		
40 (28)		
44 (2C)		
48 (30)		
52 (34)		54(36) COMMAND ID 56(37) COMMAND FUNCTION
56 (38)		
60 (3C)		
64 (40)		
68 (44)	NUMBER OF RECORDS	70(46) LAST BUFF
72 (48)		
76 (4C)		
80 (50)		
84 (54)	RESERVED	
88 (58)		
92 (5C)		
96 (60)		
100 (64)	"PRIMARY CTASK = "	
104 (68)		
108 (6C)		
112 (70)		
116 (74)	CONTROL TASK NAME	
120 (78)	BLANKS	
124 (7C)		126(7E)
128 (80)	"AID = "	
132 (84)		
136 (88)	AUTHORIZATION IDENTIFIER (AID)	
140 (8C)		
144 (90)		
148 (94)		
152 (98)		
156 (9C)	BLANKS	
160 (A0)		
164 (A4)		
168 (A8)		
172 (AC)		
176 (B0)		
180 (B4)		
184 (B8)		
188 (BC)		
192 (C0)		
196 (C4)		
200 (C8)		
204 (CC)		
208 (D0)		
212 (D4)		
216 (D8)	BLANKS	
220 (DC)		
224 (E0)		
228 (E4)		
232 (E8)		
236 (EC)		
240 (F0)		
244 (F4)		
248 (F8)		
252 (FC)		

Figure D-17 PRIMARY Command Service Message Record Format

Fields:

COMMAND ID	contains an X'10' to identify the PRIMARY command.
COMMAND FUNCTION	contains an X'0' to identify the DISPLAY function of the PRIMARY command.
NUMBER OF RECORDS	contains an X'1' or X'2' to indicate the number of records used in the buffer.
LAST BUFF	contains a setting (X'FFFF') that indicates that the buffer is the last in a chain of buffers. If the buffer is not the last buffer in a chain, the field contains zeros.
PRIMARY CTASK=	is a text message.
TASKNAME	is the primary control task.
,AID=	is a text message.
PRIMARY AID	is the primary authorization identifier.

D.4.3.11 PRIORITY Command Service Record

The format of the PRIORITY command service record is presented in Figure D-18.

0 (00)	MESSAGE BUFFER FOR SVC6 SEND DATA	
20 (14)	:	
24 (18)	:	
28 (1C)	MESSAGE HEADER	
32 (20)		
36 (24)		
40 (28)		
44 (2C)		
48 (30)		
52 (34)	54(38) COMMAND ID	65(37) COMMAND FUNCTION
56 (38)		
60 (3C)		
64 (40)		
68 (44)	NUMBER OF RECORDS	70(46) LAST BUFF
72 (48)		
76 (4C)		
80 (50)	RESERVED	
84 (54)		
88 (58)		
92 (5C)		
96 (60)		
100 (64)	"PRIORITY = "	
104 (68)		107(68)
108 (6C)	PRIORITY VALUE	
112 (70)		
118 (74)		
120 (78)		
124 (7C)		
128 (80)		
132 (84)		
138 (88)		
140 (8C)	BLANK	
144 (90)		
148 (94)		
152 (98)		
156 (9C)		
160 (A0)		
164 (A4)		
168 (A8)		
172 (AC)		
176 (B0)		
180 (B4)		
184 (B8)		
188 (BC)		
192 (C0)		
196 (C4)		
200 (C8)		
204 (CC)		
208 (D0)		
212 (D4)	BLANK	
216 (D8)		
220 (DC)		
224 (E0)		
228 (E4)		
232 (E8)		
236 (EC)		
240 (F0)		
244 (F4)		
248 (F8)		
252 (FC)		

Figure D-18 PRIORITY Command Service Message Record Format

Fields:

COMMAND ID	contains an X'12' to identify the PRIORITY command.
COMMAND FUNCTION	contains an X'0' to identify the DISPLAY function of the PRIORITY command.
NUMBER OF RECORDS	contains an X'1' or X'2' to indicate the number of records used in the buffer.
LAST BUFF	contains a setting (X'FFFF') that indicates that the buffer is the last in a chain of buffers. If the buffer is not the last buffer in a chain, the field contains zeros.
PRIORITY	is a text message.
DEFAULT OUTPUT PRIORITY	is the default output priority value.

D.4.3.12 QUEUE Command Service Record

The format of the QUEUE command service record is presented in Figure D-19.

0 (00)		MESSAGE BUFFER FOR SVC8 SEND DATA	
20 (14)			
24 (18)			
28 (1C)	MESSAGE HEADER		
32 (20)			
36 (24)			
40 (28)			
44 (2C)			
48 (30)			
52 (34)		54(36) COMMAND ID	55(37) COMMAND FUNCTION
56 (38)			
60 (3C)			
64 (40)			
68 (44)	NUMBER OF RECORDS	70(46)	LAST BUFF
72 (48)			
76 (4C)			
80 (50)	RESERVED		
84 (54)			
88 (58)			
92 (5C)			
96 (60)			
100 (64)	QUEUE FILE =		
104 (68)			
108 (6C)		109(6D)	
112 (70)			
116 (74)	FILE DESCRIPTOR OF QUEUE FILE		
120 (78)			
124 (7C)			
128 (80)			
132 (84)			
136 (88)			
140 (8C)			
144 (90)	BLANKS		
148 (94)			
152 (98)			
156 (9C)			
160 (A0)			
164 (A4)			
168 (A8)			
172 (AC)			
176 (B0)			
180 (B4)			
184 (B8)			
188 (BC)			
192 (C0)			
196 (C4)			
200 (C8)	BLANKS		
204 (CC)			
208 (D0)			
212 (D4)			
216 (D8)			
220 (DC)			
224 (E0)			
228 (E4)			
232 (E8)			
236 (EC)			
240 (F0)			
244 (F4)			
248 (F8)			
252 (FC)			

Figure D-19 QUEUE Command Service Message Record Format

Fields:

COMMAND ID	contains an X'14' to identify the QUEUE command.
COMMAND FUNCTION	contains an X'0' to identify the DISPLAY function of the QUEUE command.
NUMBER OF RECORDS	contains an X'1' or X'2' to indicate the number of records used in the buffer.
LAST BUFF	contains a setting (X'FFFF') that indicates that the buffer is the last in a chain of buffers. If the buffer is not the last buffer in a chain, the field contains zeros.
QUEUE FILE=	is a text message.
FILE DESCRIPTOR	contains the full file descriptor for the current spool queue file.

D. 4. 3. 13 SCTASK Command Service Record

The SCTASK command service record has a variable number of entries. The format of this message is presented in Figure D-20.

0 (00)		MESSAGE BUFFER FOR SVC8 SEND DATA	
20 (14)			⋮
24 (18)			⋮
28 (1C)	MESSAGE HEADER		
32 (20)			
36 (24)			
40 (28)			
44 (2C)			
48 (30)			
52 (34)		54(36) COMMAND ID	55(37) COMMAND FUNCTION
56 (38)			
60 (3C)			
64 (40)			
68 (44)	NUMBER OF RECORDS	70(46)	LAST BUFF
72 (48)			
76 (4C)			
80 (50)			
84 (54)	RESERVED		
88 (58)			
92 (5C)			
98 (60)	"CTASK = "		
100 (64)			
104 (68)	MONITOR CONTROL TASK NAME		
108 (6C)			
112 (70)	BLANKS		
116 (74)			
120 (78)	SCTASKS =		
124 (7C)			
128 (80)	SUBCONTROL TASK 1		
132 (84)			
136 (88)	,		
140 (8C)	SUBCONTROL TASK 2		
144 (90)			
148 (94)	,		
152 (98)	SUBCONTROL TASK 3		
156 (9C)			
160 (A0)	,		
164 (A4)	SUBCONTROL TASK 4		
168 (A8)			
172 (AC)	SUBCONTROL TASK 5		
176 (B0)			
180 (B4)	BLANK		
184 (B8)			
188 (BC)			
192 (C0)			
196 (C4)			
200 (C8)			
204 (CC)			
208 (D0)			
212 (D4)			
216 (D8)			
220 (DC)			
224 (E0)			
228 (E4)			
232 (E8)			
236 (EC)			
240 (F0)			
244 (F4)			
248 (F8)			
252 (FC)			

Figure D-20 SCTASK Command Service Message Record Format

Fields:

COMMAND ID	contains an X'18' to identify the SCTASK command.
COMMAND FUNCTION	contains an X'0' to identify the display function of the SCTASK command.
NUMBER OF RECORDS	contains an X'1' or X'2' to indicate the number of records used in the buffer.
LAST BUFF	contains a setting (X'FFFF') that indicates that the buffer is the last in a chain of buffers. If the buffer is not the last buffer in a chain, the field contains zeros.
CTASK=	is a text header.
MONITOR CONTROL TASK NAME	contains the name of the control task monitoring the subcontrol tasks.
SCTASKS=	is a text header.
SUBCONTROL TASK N	is the name of a subcontrol task.

D. 4. 3. 14 SECONDARY Command Service Record

The SECONDARY command service record has a variable number of entries depending upon the spooling environment. The format of the SECONDARY command service record is presented in Figure D-20.

0 (00)		MESSAGE BUFFER FOR SVC6 SEND DATA	
20 (14)	:		
24 (18)	MESSAGE HEADER		
28 (1C)			
32 (20)			
36 (24)			
40 (28)			
44 (2C)			
48 (30)			
52 (34)	64(36) COMMAND ID	56(37) COMMAND FUNCTION	
56 (38)			
60 (3C)			
64 (40)			
68 (44)	NUMBER OF RECORDS	70(46) LAST BUFF	
72 (48)			
76 (4C)	RESERVED		
80 (50)			
84 (54)			
88 (58)			
92 (5C)			
96 (60)	DEVICE/AID =		
100 (64)			
104 (68)			
108 (6C)	PSEUDO DEVICE		
112 (70)	NAME 1	:	/
116 (74)	SECONDARY AUTHORIZATION IDENTIFIER 1		
120 (78)	,		
124 (7C)	PSEUDO DEVICE NAME 2		
128 (80)	:	/	
132 (84)	SECONDARY AUTHORIZATION IDENTIFIER 2		
136 (88)	,		PSEUDO
140 (8C)	DEVICE NAME 3		:
144 (90)	/		
148 (94)	SECONDARY AUTHORIZATION IDENTIFIER 3		
152 (98)	,		PSEUDO DEVICE
156 (9C)	NAME 4		:
160 (A0)	SECONDARY AUTHORIZATION IDENTIFIER 4		
168 (A8)	BLANKS		
172 (AC)			
176 (B0)			
180 (B4)			
184 (B8)			
188 (BC)			
192 (C0)			
196 (C4)	ETC.		
200 (C8)			
204 (CC)			
208 (D0)			
212 (D4)			
216 (D8)			
220 (DC)			
224 (E0)			
228 (E4)			
232 (E8)			
236 (EC)			
240 (F0)			
244 (F4)			
248 (F8)			
252 (FC)			

Figure D-21 SECONDARY Command Service Message Record Format

Fields:

COMMAND ID	contains an X'17' to identify the SECONDARY command.
COMMAND FUNCTION	contains an X'0' to identify the DISPLAY function of the SECONDARY command.
NUMBER OF RECORDS	contains an X'1' or X'2' to indicate the number of records used in the buffer.
LAST BUFF	contains a setting (X'FFFF') that indicates that the buffer is the last in a chain of buffers. If the buffer is not the last buffer in a chain, the field contains zeros.
DEVICE/AID=	is a text message.
PSEUDO DEVICE 1	is the name of a pseudo device.
SECONDARY AID FOR PSEUDO DEVICE 1	is the secondary authorization identifier for the preceding pseudo device.

D.4.3.15 The SPVOLUME Command Service Record

The SPVOLUME command service record has a variable number of entries depending upon the number of spooled volumes on the system. The format of the SPVOLUME command service record is presented in Figure D-21.

0 (00)	MESSAGE BUFFER FOR SVC8 SEND DATA		
20 (14)	:		
24 (18)	MESSAGE HEADER		
28 (1C)			
32 (20)			
36 (24)			
40 (28)			
44 (2C)			
48 (30)			
52 (34)			
56 (38)			
60 (3C)			
64 (40)			
68 (44)	NUMBER OF RECORDS	76 (46)	LAST BUFF
72 (48)			
76 (4C)			
80 (50)			
84 (54)			
88 (58)			
92 (5C)			
96 (60)			
100 (64)	SPOOL VOLUMES =		
104 (68)			
108 (6C)			SPOOL
112 (70)	VOLUME	1	,
116 (74)	SPOOL VOLUME 2		
120 (78)	,	SPOOL VOLUME	
124 (7C)	3	,	SPOOL
128 (80)	VOLUME 4	,	SPOOL
132 (84)	VOLUME	5	,
136 (88)	SPOOL VOLUME 6		
140 (8C)	,	SPOOL VOLUME	
144 (90)	7	,	SPOOL
148 (94)	VOLUME 8	,	SPOOL
152 (98)	VOLUME 9		,
156 (9C)	SPOOL VOLUME 10		
160 (A0)	,	SPOOL VOLUME	
164 (A4)	11	,	SPOOL
168 (A8)	VOLUME 12	,	SPOOL
172 (AC)	VOLUME	13	,
176 (B0)			
180 (B4)	ETC.		
184 (B8)			
188 (BC)			
192 (C0)			
196 (C4)			
200 (C8)			
204 (CC)			
208 (D0)			
212 (D4)			
216 (D8)			
220 (DC)			
224 (E0)			
228 (E4)			
232 (E8)			
236 (EC)			
240 (F0)			
244 (F4)			
248 (F8)			
252 (FC)			

Figure D-22 SPVOLUME Command Service Message Record Format

Fields:

COMMAND ID	contains an X'19' to identify the SPVOLUME command.
COMMAND FUNCTION	contains an X'0' to identify the DISLAY function of the SPVOLUME command.
NUMBER OF RECORDS	contains an X'1' or X'2' to indicate the number of records used in the buffer.
LAST BUFF	contains a setting (X'FFFF') that indicates that the buffer is the last in a chain of buffers. If the buffer is not the last buffer in a chain, the field contains zeros.
SPVOLUMES=	is a text header.
VOLUME N	contains the volume name of a spooled disk volume.

D.4.4 Exception Response Messages

The format of an exception response message is presented in Figure D-23.

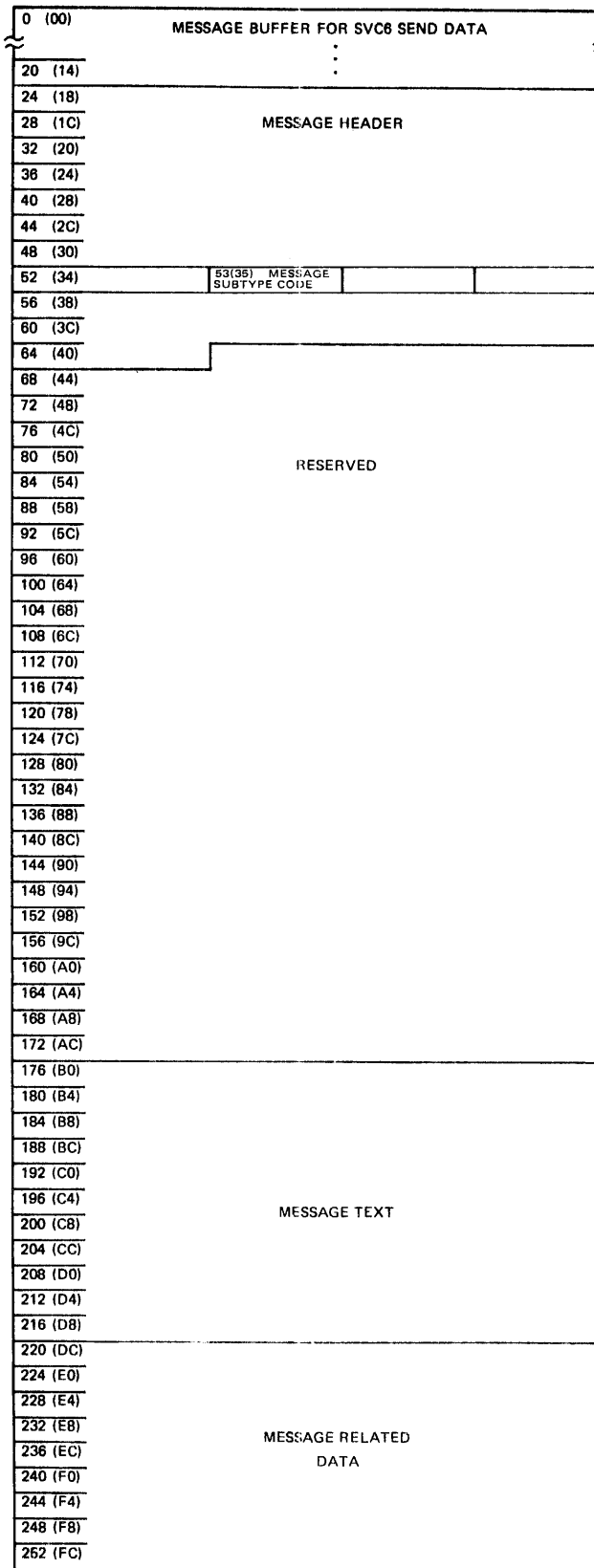


Figure D-23 EXCEPTION Response Message Format

Fields:

MESSAGE SUBTYPE CODE contains one of the error code numbers presented in Table D-1. This code indicates the type of exception that occurred.

MESSAGE TEXT contains the message text description that corresponds with the preceding error code identified in the message subtype field. The message text field entries are presented in Table D-1.

MESSAGE RELATED DATA contains the item that caused the error with the exception of error code 5. The message related data field entries are presented in Table D-1. When the error code is 5, this field contains the authorization level required to issue a command.

TABLE D-1 EXCEPTION MESSAGE ERROR CODES, MESSAGE TEXT, AND MESSAGE RELATED DATA SUMMARY

ERROR CODE (DECIMAL)	MESSAGE TEXT	MESSAGE RELATED DATA
0	No message (no error occurred)	
1	INVALID COMMAND:	command that caused error
2	INVALID KEYWORD:	keyword that caused error
3	INVALID PARAMETER:	parameter that caused error
4	INVALID SEPARATOR:	separator that caused error
5	INVALID AUTHORIZATION LEVEL:	level required
6	MISSING PARAMETER:	-
7	INVALID DEVICE:	device name given

TABLE D-1 EXCEPTION MESSAGE ERROR CODES, MESSAGE TEXT, AND MESSAGE RELATED DATA SUMMARY (Continued)

ERROR CODE (DECIMAL)	MESSAGE TEXT	MESSAGE RELATED DATA
8	INVALID TASK:	task name given
9	DUPLICATE TASK NAME	task name given
10	DUPLICATE DEVICE	device name given
11	TASK NOT INCLUDED:	task name given
12	DEVICE NOT INCLUDED:	device name given
13	VOLUME NOT INCLUDED:	volume name given
14	CANNOT COMMUNICATE WITH CTASK	control task name
15	INVALID VOLUME	volume name given
16	DUPLICATE VOLUME:	volume name given
17	INVALID INDEX BLOCK SIZE:	index block size given
18	INVALID DATA BLOCK SIZE:	data block size given
19	INVALID PRIORITY	priority given
20	PRIMARY AUTHORITY REJECTED	
21	UNIT NOT INCLUDED	unit-id given
22	DUPLICATE UNIT NAME	unit-id given
23	INVALID AUTHORIZATION ID:	authorization id given
24	FILE NOT FOUND:	file descriptor given
25	PARAMETER CONFLICT:	parameter1, parameter2
26	INVALID FORM:	form name given
27	DEVICE ACTIVE:	device name given

TABLE D-1 EXCEPTION MESSAGE ERROR CODES, MESSAGE TEXT, AND MESSAGE RELATED DATA SUMMARY (Continued)

ERROR CODE (DECIMAL)	MESSAGE TEXT	MESSAGE RELATED DATA
28	DEVICE INACTIVE:	device name given
29	FORM NOT INCLUDED	form name given
30	FILE PRINTING	
31	REQUEST IS PENDING FOR FORM:	name of form
32	SVC1 ERROR	type
33	SVC7 ERROR	type
34	SVC6 ERROR	type
35	IMPLICITLY CREATED PSEUDO DEVICE	name of pseudo device
36	MISSING CONTROL CARD	-
37	INVALID CONTROL CARD	-
38	IO TASK WAS CANCELLED	-
39	PRIMARY CTASK WAS CANCELLED	-
40	NO AVAILABLE LUs	-
41	NO SPOOL VOLUMES	-
42	NO SUBCONTROL TASKS	-
43	NO INPUT DEVICES IN SYSTEM	-
44	NO QUEUE ENTRY FOUND	-
45	NO GROUP DEFAULT ESTABLISHED	-
46	VERIFY COMPLETED ON	name of device
66	SECONDARY AUTHORITY REJECTED	-
67	INVALID FILE NAME:	name of invalid file

TABLE D-1 EXCEPTION MESSAGE ERROR CODES, MESSAGE TEXT, AND MESSAGE RELATED DATA SUMMARY (Continued)

ERROR CODE (DECIMAL)	MESSAGE TEXT	MESSAGE RELATED DATA
68	NO DEVICES INCLUDED	-
69	SPL/32 OS SUPPORT NOT INCLUDED	-
70	PLEASE CONTINUE	-
71	NO SECONDARY AUTHORIZATION	-
72	NO FILES TO BE PRINTED	-
73	INVALID BANNER/TRAILER PARAMETER:	invalid parameter
74	INVALID COPIES:	invalid # of copies
75	NO QUEUE ENTRIES, NO DEVICES	-

D.4.5 Attention Messages

There are two types of attention messages:

- SVC1 and 7 error messages
- General attention messages

The format of the SVC error messages is presented in Figure D-24.

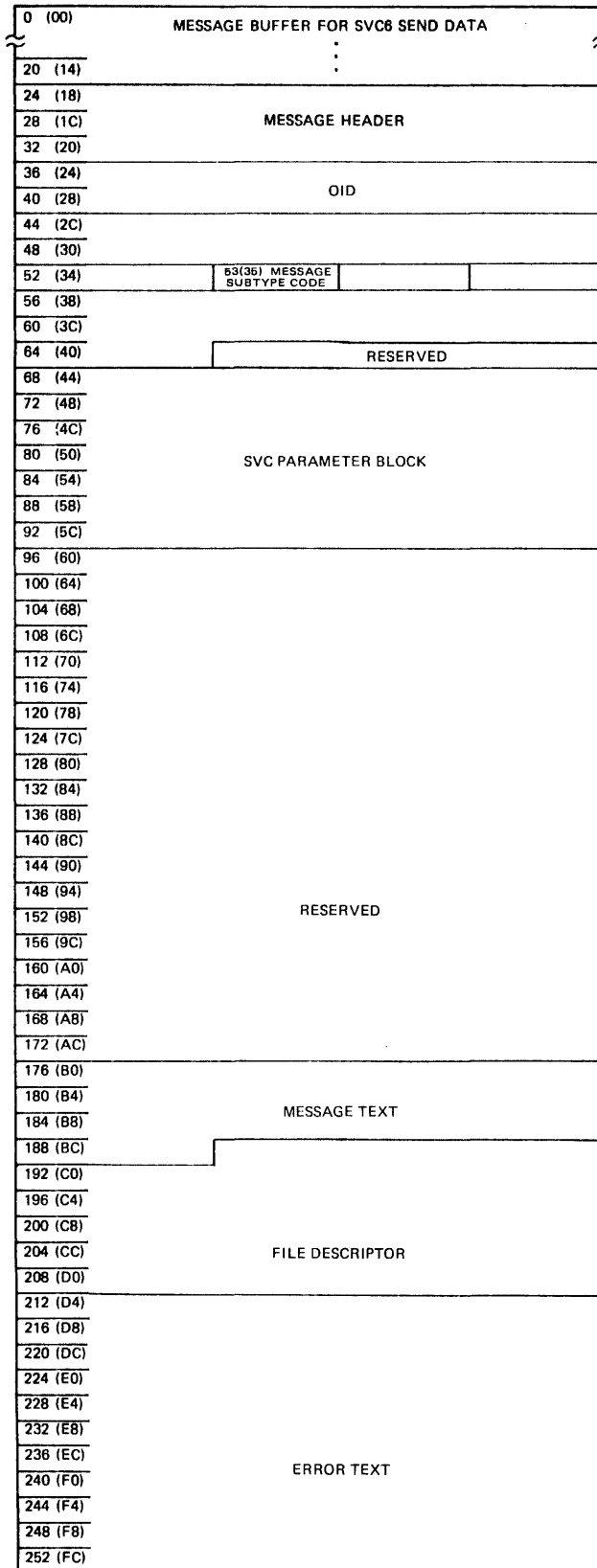


Figure D-24 SVC1 and 7 Attention Messages Record Format

Fields:

OID This field is undefined for attention messages. The control task must route the message according to the AID supplied.

MESSAGE SUBTYPE CODE contains either of the following error codes:

- 1 for SVC1 errors
- 2 for SVC7 errors

MESSAGE TEXT contains one of the following error descriptions:

- If the attention code is 1; then the message text is:

I/O ERROR ON

- If the attention code is 2, then the message text is:

FILE ERROR ON

FILE DESCRIPTOR contains the name of the file or device where the error occurred.

ERROR TEXT contains one of the entries presented in Table D-2 according to the attention code present.

TABLE D-2 ERROR TEXT FIELD ENTRIES FOR
SVC1 AND 7 ATTENTION MESSAGES

ATTENTION CODE 1	ATTENTION CODE 2
Illegal function	Illegal function
Device unavailable	lu error
End of medium	Volume error
End of file	Name error
Unrecoverable error	File error
Recoverable error	Size error
Illegal lu	Protect error
Device Busy	Privilege error
	Buffer error
	Assignment error
	Form error

The format of the general attention message is presented in Figure D-25.

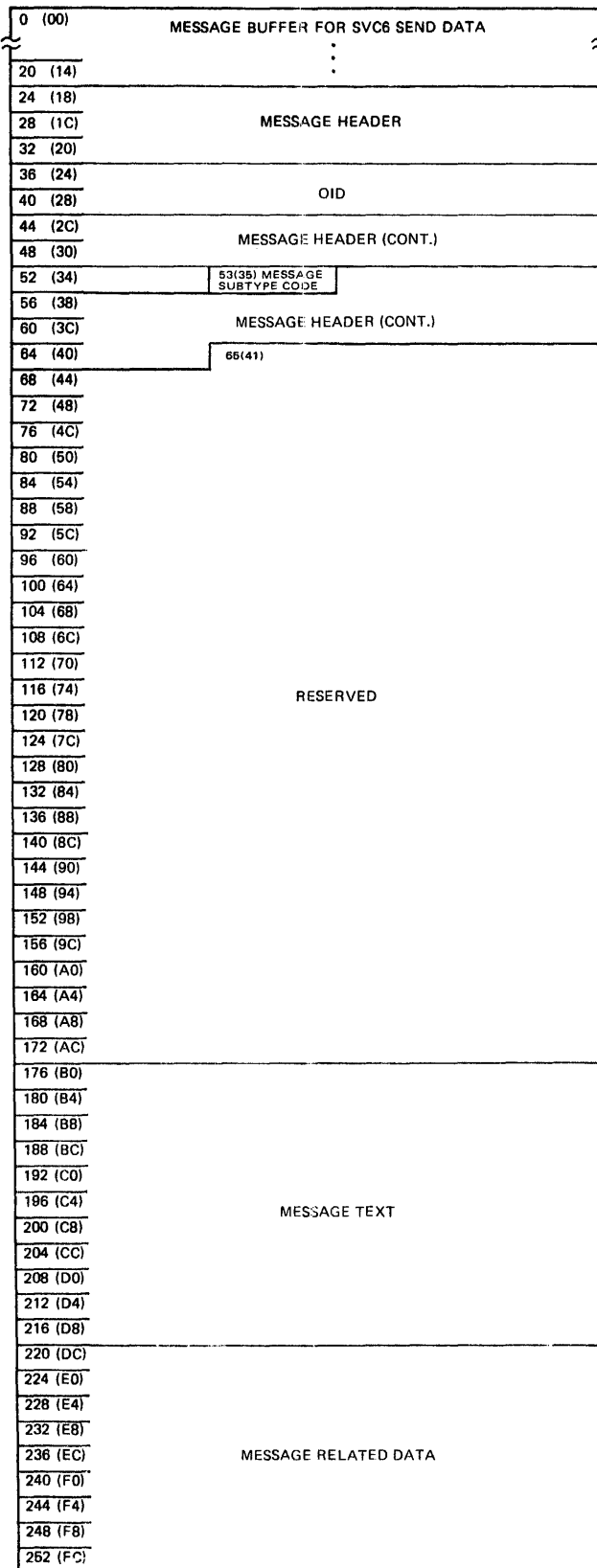


Figure D-25 General Attention Message Format

Fields:

OID is not defined for attention messages. The control task must route the message according to the AID supplied.

MESSAGE SUBTYPE CODE contains one of the error codes printed in Table D-3.

MESSAGE TEXT contains a description of the error. The entries for this field are presented in Table D-3.

MESSAGE RELATED DATA contains the entry in Table D-3 that corresponds with the attention code and message text present.

TABLE D-3 GENERAL ATTENTION MESSAGE FIELD ENTRIES

ATTENTION CODE	MESSAGE TEXT	MESSAGE DATA
32	REQUEST IS PENDING FOR FORM:	form name
37	MISSING CONTROL CARD:	device name
38	INVALID CONTROL CARD:	device name

D.4.6 Mnemonic Message Format

Mnemonic messages use the same buffer structure and chain structure as previously described in Section D.4.2.2. The format of a mnemonic message is presented in Figure D-26.

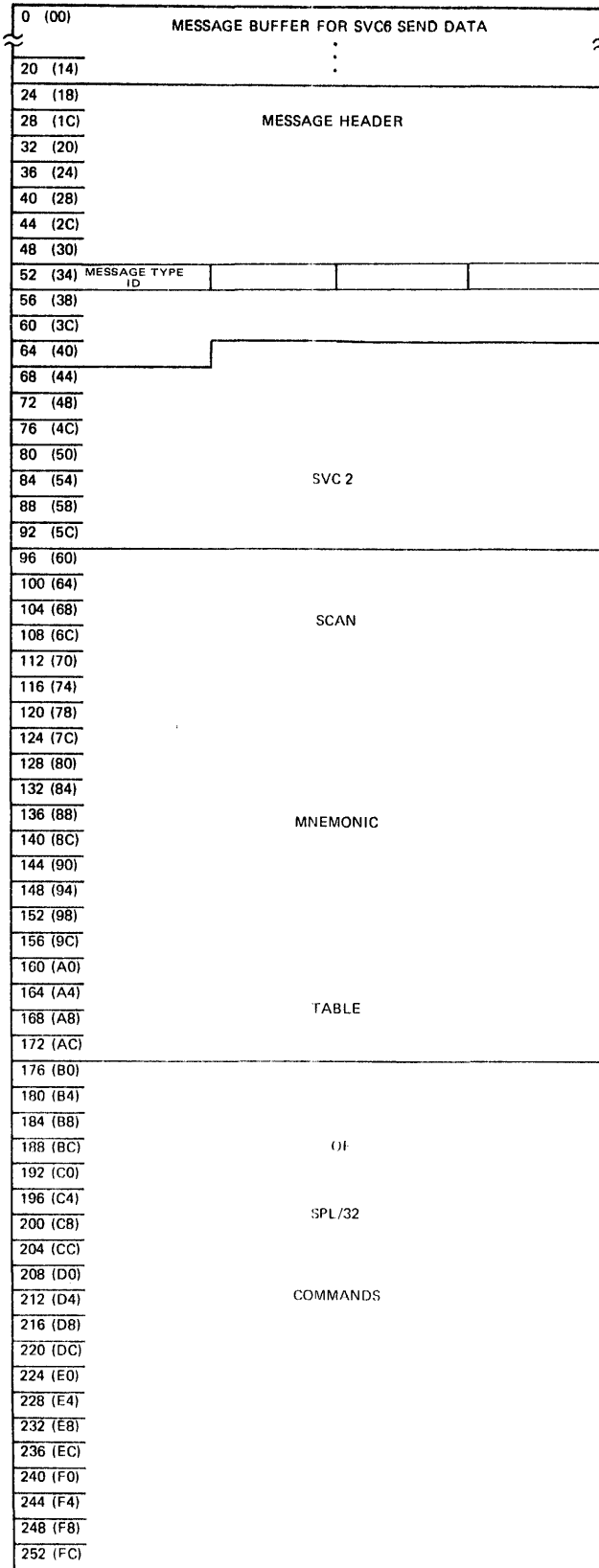


Figure D-26 Mnemonic Message Format

Fields:

MESSAGE TYPE ID	contains an X'14' to identify a mnemonic message.
SVC2 SCAN MNEMONIC TABLE	contains all the valid command mnemonics for the spooler. SVC2 SCAN uses this table to check for valid spooler commands.

D.4.7 Authority Message Format

The format of an authority message is presented in Figure D-27.

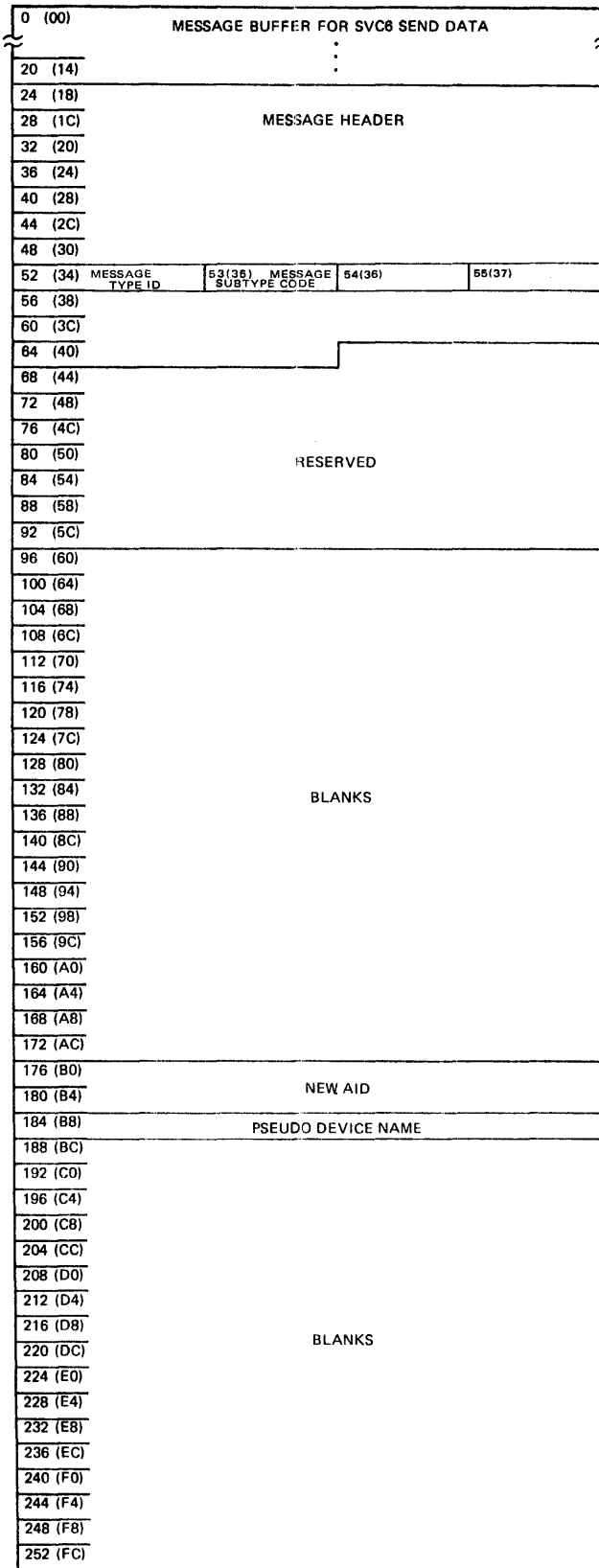


Figure D-27 Authority Message Format

Fields:

MESSAGE TYPE ID	contains an X'15' to identify the authority message.
MESSAGE SUBTYPE CODE	contains one of the following settings, based on the function of the message: <ul style="list-style-type: none">● 1 = Primary AID approval request● 2 = Secondary AID approval request● 3 = Authority change accepted● 4 = Authority change rejected
NEW AID	is the new authorization identifier to be accepted or rejected.
PSEUDO DEVICE NAME	if the message subtype code field is set to 2, this field contains the name of the pseudo device that the new AID applies to.

D.5 MESSAGE RECOVERY PROCEDURES

If a message from SPL/32 to a control or subcontrol task fails to reach its destination, SPL/32 attempts recovery depending on the status returned by the SVC6 Send Data function. The error status and recovery procedures of SPL/32 are listed in Table D-4.

TABLE D-4 SPL/32 ERROR STATUS AND RECOVERY INFORMATION

ERROR STATUS	RECOVERY PROCEDURE
X'04'	Messages are rerouted to the appropriate control origin when communication with a secondary or primary location is lost. When a secondary location is lost, messages are rerouted to the monitoring control task. When the primary location is lost, messages are rerouted to the default control task and location, .SPL and the system console, respectively.
X'0B'	The send is retried five times at 5 second intervals. If communication fails all five times, the last error is handled as if an error status X'0B' was obtained.
X'0C'	Handled the same as status X'0B'.

APPENDIX E
SPL/32 COMMANDS PREPARED FORMAT

E.1 INTRODUCTION

The prepared format of the spooler commands is for use by control tasks that do not use the ASCII form of the commands as previously presented in Chapter 6.

Prepared spooler commands are passed to the spooler through the uniform spooler control interface (USCI). The SVC6 send data format for prepared commands consists of an SVC6 header, a message header, and a record containing the command in its prepared form. The format for SVC6 and message headers is identical to that used to pass ASCII commands to the spooler. A command is designated as prepared by setting byte 54 of the message header to the desired command identifier. This byte is set to zero when using the ASCII command format. Byte 55 of the message header defines the specific function within the command (i.e., include, display, or remove).

In the following sections, the following information is presented for each prepared command:

- the command mnemonic,
- the command-identifier setting for byte 54 of the message header,
- the record format of the parameters associated with the command, and
- the command-function-identifier setting for byte 55 of the message header.

The function of each command and its associated parameters is presented in detail in Chapter 6.

NOTE

The byte offset within the record fields of each command format is the offset into the data record, starting at byte 176 of the command buffer. Any unused fields should be set to nulls (X'0000').

E.2 ATTACH COMMAND

Command mnemonic: ATTACH

Command identifier (byte 54 of message header): 1

Command function identifier (byte 55 of message header):

X'03' = to ATTACH physical devices to pseudo device

Record format:

4005

0 (00)	PSEUDO DEVICE NAME
4 (04)	PHYSICAL DEVICE 1
8 (08)	PHYSICAL DEVICE 2
12 (C)	PHYSICAL DEVICE 3
16 (10)	PHYSICAL DEVICE 4
20 (14)	PHYSICAL DEVICE 5
24 (18)	PHYSICAL DEVICE 6
28 (1C)	PHYSICAL DEVICE 7
32 (20)	PHYSICAL DEVICE 8
36 (24)	PHYSICAL DEVICE 9
40 (28)	PHYSICAL DEVICE 10
44 (2C)	PHYSICAL DEVICE 11
48 (30)	PHYSICAL DEVICE 12
52 (34)	PHYSICAL DEVICE 13
56 (38)	PHYSICAL DEVICE 14
60 (3C)	PHYSICAL DEVICE 15
64 (40)	PHYSICAL DEVICE 16
68 (44)	PHYSICAL DEVICE 17
72 (48)	PHYSICAL DEVICE 18
76 (4C)	PHYSICAL DEVICE 19

Fields:

Pseudo device number

specifies the pseudo device to which the physical devices are being attached.

Physical device name

specifies the names of the physical devices to be attached to the specified pseudo device.

E.3 BANNER COMMAND

Command mnemonic: BANNER

Command identifier (byte 54 of message header): 2

Command function identifier (byte 55 of the message header):

X'10' = NONE (no banner page)

X'20' = SINGLE (one banner page)

X'40' = DOUBLE (two banner pages)

Record format:

4006

0 (00)	DEVICE NAME	
4 (04)	VOLUME	
8 (08)	FILENAME	
12 (C)		
16 (10)	EXTENSION	19(18) ACCOUNT
20 (14)	CHARACTER	
24 (18)		
28 (1C)		
32 (20)		
36 (24)		
40 (28)		
44 (2C)		
48 (30)		BLANKS
52 (34)		
56 (38)		
60 (3C)		
64 (40)		
68 (44)		
72 (48)		
76 (4C)		

Fields:

Device name	contains the pseudo or physical device name to which the BANNER command pertains.
Volume	contains the volume name where the file containing the banner page specifications resides.
Filename	contains the name of the file that contains the banner page specifications.
Extension	contains the extension for the file that contains the banner page specifications.
Account	contains the account number for the file that contains the banner page specifications
CHARACTER	is the character with which the letters in the banner are to be formed.

E.4 BLOCK COMMAND

Command mnemonic: BLOCK

Command identifier (byte 54 of message header): 3

Command function identifier (byte 55 of message header):

X'00' = to display

X'03' = to specify

Record format:

4007

0 (00)	DATA BLOCKSIZE	2(02)	INDEX BLOCKSIZE
4 (04)			
8 (08)			
12 (C)			
16 (10)			
20 (14)			
24 (18)			
28 (1C)			BLANKS
32 (20)			
36 (24)			
40 (28)			
44 (2C)			
48 (30)			
52 (34)			
56 (38)			
60 (3C)			
64 (40)			
68 (44)			
72 (48)			
76 (4C)			

Fields:

Data blocksize contains the number for the default data blocksize.

Index blocksize contains the number for the default index blocksize.

E.5 CHANGE COMMAND

Command mnemonic: CHANGE

Command identifier (byte 54 of message header): 4

Command function identifier (byte 55 of message header):

X'03' = change

Record format:

4008

0 (00)	VOLUME	
4 (04)		
8 (08)	FILENAME	
12 (C)	EXTENSION	15(F) ACCOUNT
16 (10)	RESERVED	
20 (14)		
24 (18)	FORMNAME	
28 (1C)	RESERVED	
32 (20)	PSEUDO DEVICE	
36 (24)	OPTIONS	
40 (28)	COPIES	
44 (2C)	PRIORITY	
48 (30)		
52 (34)		
56 (38)		
60 (3C)		
64 (40)		
68 (44)		
72 (48)		
76 (4C)		

Fields:

Volume contains the volume name of the file to be changed.

Filename contains the name of the file to be changed.

Extension contains the name extension of the file to be changed.

Account	contains the account number of the file to be changed.
Formname	contains the name of a new form to be used for output.
Pseudo device	contains the name of a new pseudo device to which the file is to be directed. This field will remain blank if there is no change to the pseudo device.
Options	contains one or a combination of the following settings to indicate the options in effect: Y'0000' No change Y'0001' VFC (if NOVFC set) Y'0002' NOVFC (if VFC set) Y'0004' IMAGE (if NOIMAGE set) Y'0008' NOIMAGE (if IMAGE set) Y'0010' DELETE (if NODELETE set) Y'0020' NODELETE (if DELETE set) Y'0040' CHECKPOINT (if NOCHECKPOINT set) Y'0080' NOCHECKPOINT (if CHECKPOINT set) Y'0100' HOLD (if RELEASE set) Y'0200' RELEASE (if HOLD set)
Number of copies	contains a new value for the number of copies of the file to be output. This field will contain zeros if there is no change in the number of copies.
Priority	contains a new value for the output priority of the file to be output. This field will contain 1s if there is no change in the output priority of the file.

E.6 CHECKPOINT COMMAND

Command mnemonic: CHECKPOINT

Command identifier (byte 54 of message header): 5

Command function identifier (byte 55) of message header):

X'00' = to display

X'01' = CHECKPOINTING on

X'02' = CHECKPOINTING off

Record format:

4009	0 (00)	CHECKPOINT VALUE
	4 (04)	
	8 (08)	
	12 (C)	
	16 (10)	
	20 (14)	
	24 (18)	
	28 (1C)	
	32 (20)	
	36 (24)	RESERVED
	40 (28)	
	44 (2C)	
	48 (30)	
	52 (34)	
	56 (38)	
	60 (3C)	
	64 (40)	
	68 (44)	
	72 (48)	
	76 (4C)	

Fields:

Checkpoint value

contains the value for the number of records output between checkpointing.

E.7 CREATE COMMAND

Command mnemonic: CREATE

Command identifier (byte 54 of message header): 6

Command function identifier (byte 55 of message header):

X'03' = to specify new pseudo devices

Record format:

4010

0 (00)	PSEUDO DEVICE 1
4 (04)	PSEUDO DEVICE 2
8 (08)	PSEUDO DEVICE 3
12 (C)	PSEUDO DEVICE 4
16 (10)	PSEUDO DEVICE 5
20 (14)	PSEUDO DEVICE 6
24 (18)	PSEUDO DEVICE 7
28 (1C)	PSEUDO DEVICE 8
32 (20)	PSEUDO DEVICE 9
36 (24)	PSEUDO DEVICE 10
40 (28)	PSEUDO DEVICE 11
44 (2C)	PSEUDO DEVICE 12
48 (30)	PSEUDO DEVICE 13
52 (34)	PSEUDO DEVICE 14
56 (38)	PSEUDO DEVICE 15
60 (3C)	PSEUDO DEVICE 16
64 (40)	PSEUDO DEVICE 17
68 (44)	PSEUDO DEVICE 18
72 (48)	PSEUDO DEVICE 19
76 (4C)	PSEUDO DEVICE 20

Fields:

Pseudo device number

contains the name of a pseudo device to be created.

E.8 CTASK COMMAND

Command mnemonic: CTASK

Command identifier (byte 54 of message header): 7

Command function identifier (byte 55 in message header):

X'00' = to display

X'01' = to include

X'02' = to remove

Record format:

4011

0 (00)	
4 (04)	CONTROL TASK 1
8 (08)	
12 (C)	CONTROL TASK 2
16 (10)	
20 (14)	CONTROL TASK 3
24 (18)	
28 (1C)	CONTROL TASK 4
32 (20)	
36 (24)	CONTROL TASK 5
40 (28)	
44 (2C)	CONTROL TASK 6
48 (30)	
52 (34)	CONTROL TASK 7
56 (38)	
60 (3C)	CONTROL TASK 8
64 (40)	
68 (44)	CONTROL TASK 9
72 (48)	
76 (4C)	CONTROL TASK 10

Fields:

Control task number

contains the name of a control task being included or removed from the spooling environment.

E.9 DELETE COMMAND

Command mnemonic: DELETE

Command identifier (byte 54 of message header): 22(16)

Command function identifier (byte 55 of message header):

X'03' = to REMOVE

Record format:

4026

0 (00)	DEVICE 1
4 (04)	DEVICE 2
8 (08)	DEVICE 3
12 (0C)	DEVICE 4
16 (10)	DEVICE 5
20 (14)	DEVICE 6
24 (18)	DEVICE 7
28 (1C)	DEVICE 8
32 (20)	DEVICE 9
36 (24)	DEVICE 10
40 (28)	DEVICE 11
44 (2C)	DEVICE 12
48 (30)	DEVICE 13
52 (34)	DEVICE 14
56 (38)	DEVICE 15
60 (3C)	DEVICE 16
64 (40)	DEVICE 17
68 (44)	DEVICE 18
72 (48)	DEVICE 19
76 (4C)	DEVICE 20

Fields:

Device number

contains the name of a pseudo or physical device to be deleted.

E.10 DETACH COMMAND

Command mnemonic: DETACH

Command identifier (byte 54 of message header): 8

Command function identifier (byte 55 of message header):

X'03' = to detach

Record format:

4012

0 (00)	PSEUDO DEVICE NAME
4 (04)	PHYSICAL DEVICE 1
8 (08)	PHYSICAL DEVICE 2
12 (C)	PHYSICAL DEVICE 3
16 (10)	PHYSICAL DEVICE 4
20 (14)	PHYSICAL DEVICE 5
24 (18)	PHYSICAL DEVICE 6
28 (1C)	PHYSICAL DEVICE 7
32 (20)	PHYSICAL DEVICE 8
36 (24)	PHYSICAL DEVICE 9
40 (28)	PHYSICAL DEVICE 10
44 (2C)	PHYSICAL DEVICE 11
48 (30)	PHYSICAL DEVICE 12
52 (34)	PHYSICAL DEVICE 13
56 (38)	PHYSICAL DEVICE 14
60 (3C)	PHYSICAL DEVICE 15
64 (40)	PHYSICAL DEVICE 16
68 (44)	PHYSICAL DEVICE 17
72 (48)	PHYSICAL DEVICE 18
76 (4C)	PHYSICAL DEVICE 19

Fields:

Pseudo device number

contains the name of a pseudo device to be detached.

Physical device number

contains the name of a physical device to be detached.

E.11 DEVICE COMMAND

Command mnemonic: DEVICE

Command identifier (byte 54 of message header): 9

Command function identifier (byte 55 of message header):

X'10'	= to forward space line	X'30'	= FREE
X'11'	= to forward space page	X'31'	= RESTRICT
X'20'	= to backward space line	X'40'	= SUSPEND IMMEDIATE
X'21'	= to backward space page	X'41'	= SUSPEND DELAYED
X'22'	= to rewind	X'50'	= CONTINUE

Record format:

4013

0 (00)	DEVICE NAME
4 (04)	NUMBER OF LINES OR PAGES
8 (08)	
12 (C)	
16 (10)	
20 (14)	
24 (18)	
28 (1C)	
32 (20)	
36 (24)	
40 (28)	BLANKS
44 (2C)	
48 (30)	
52 (34)	
56 (38)	
60 (3C)	
64 (40)	
68 (44)	
72 (48)	
76 (4C)	

Fields:

Device name	contains the name of a pseudo or physical device.
Number of pages/lines	contains the number of lines or pages to be spaced according to the function id.

E.12 END COMMAND

Command mnemonic: END

Command identifier (byte 54 of message header): 21(15)

Record format:

The Data Record is not used for this command.

Command function identifier (byte 55 of message header):

X'01' = IMMEDIATE

X'02' = DELAYED

E.13 FLUSH COMMAND

Command mnemonic: FLUSH

Command identifier (byte 54 of message header): 10(A)

Command function identifier (byte 55 of message header):

X'10' = FLUSH fd

X'11' = FLUSH fd, DELETE

X'20' = FLUSH dev:

Record format:

4014

0 (00)	VOLUME	
4 (04)	FILENAME	
8 (08)	FILENAME	
12 (C)	EXTENSION	ACCOUNT
16 (10)	DEVICE	
20 (14)	BLANKS	
24 (18)		
28 (1C)		
32 (20)		
36 (24)		
40 (28)		
44 (2C)		
48 (30)		
52 (34)		
56 (38)		
60 (3C)		
64 (40)		
68 (44)		
72 (48)		
76 (4C)		

Fields:

Volume	contains the volume name for the file to be flushed.
Filename	contains the name of the file to be flushed.
Extension	contains the filename extension for the file to be flushed.
Account	contains the account number of the file to be flushed.
Device	contains the name of the device to be halted.

E.14 FORM COMMAND (INCLUDE)

Command mnemonic: FORM (INCLUDE)

Command identifier (byte 54 of message header): 11(B)

Command function identifier (byte 55 of message header):

X'01' = include

Record format:

4015

0 (00)				
4 (04)	FORMNAME 1			
8 (08)				
12 (C)				
16 (10)	PAGE LENGTH 1	TOP MARGIN 1	BOTTOM MARGIN 1	PAGE WIDTH 1
20 (14)	FORMNAME 2			
24 (18)				
28 (1C)				
32 (20)				
36 (24)	PAGE LENGTH 2	TOP MARGIN 2	BOTTOM MARGIN 2	PAGE WIDTH 2
40 (28)	FORMNAME 3			
44 (2C)				
48 (30)				
52 (34)				
56 (38)	PAGE LENGTH 3	TOP MARGIN 3	BOTTOM MARGIN 3	PAGE WIDTH 3
60 (3C)	FORMNAME 4			
64 (40)				
68 (44)				
72 (48)				
76 (4C)	PAGE LENGTH 4	TOP MARGIN 4	BOTTOM MARGIN 4	PAGE WIDTH 4

Fields:

Formname	contains the names of forms to be included in the spooler environment.
Page length	contains default page lengths for the associated form.
Top margin	contains the top margin length for the associated form.
Bottom margin	contains the bottom margin length for the associated form.
Page width	contains the horizontal page size for the associated form.

E.15 FORM COMMAND (REMOVE)

Command mnemonic: FORM (REMOVE)

Command identifier (byte 54 of message header): 11(B)

Command function identifier (byte 55 of message header):

X'02' = to REMOVE

Record format:

4063

0 (00)	
4 (04)	
8 (08)	FORMNAME 1
12 (C)	
16 (10)	
20 (14)	
24 (18)	FORMNAME 2
28 (1C)	
32 (20)	
36 (24)	
40 (28)	FORMNAME 3
44 (2C)	
48 (30)	
52 (34)	
56 (38)	FORMNAME 4
60 (3C)	
64 (40)	
68 (44)	
72 (48)	FORMNAME 5
76 (4C)	

Fields:

FORMNAME N

contains the name of a form to be removed from the spooler environment.

E.16 FORM COMMAND (MOUNT, MOUNT VERIFY)

Command mnemonic: FORM (MOUNT, MOUNT/VERIFY)

Command identifier (byte 54 of message header): 11(B)

Command function identifier (byte 55 of message header):

X'00' = to display

X'04' = MOUNT

X'14' = MOUNT/VERIFY

Record format:

4016

0 (00)	
4 (04)	
8 (08)	FORMNAME
12 (C)	
16 (10)	DEVICE 1
20 (14)	DEVICE 2
24 (18)	DEVICE 3
28 (1C)	DEVICE 4
32 (20)	DEVICE 5
36 (24)	DEVICE 6
40 (28)	DEVICE 7
44 (2C)	DEVICE 8
48 (30)	DEVICE 9
52 (34)	DEVICE 10
56 (38)	DEVICE 11
60 (3C)	DEVICE 12
64 (40)	DEVICE 13
68 (44)	DEVICE 14
72 (48)	DEVICE 15
76 (4C)	NUMBER OF PAGES

Fields:

Formname	contains the name of the form mounted on the specified devices.
Device n	contains the name of the devices to have the specified formname mounted.
Number of pages	specifies the number of pages to be output for alignment verification.

E. 17 GLOBAL COMMAND

Command mnemonic: GLOBAL

Command identifier (byte 54 of message header): 12(C)

Command function identifier (byte 55 of message header):

X'00' = to display

X'03' = to specify

Record format:

4017

0 (00)	PSEUDO DEVICE NAME
4 (04)	
8 (08)	
12 (C)	
16 (10)	
20 (14)	
24 (18)	
28 (1C)	BLANKS
32 (20)	
36 (24)	
40 (28)	
44 (2C)	
48 (30)	
52 (34)	
56 (38)	
60 (3C)	
64 (40)	
68 (44)	
72 (48)	
76 (4C)	

Field:

Pseudo device name

contains the name of the global pseudo device.

E.18 GROUP COMMAND

Command mnemonic: GROUP

Command identifier (byte 54 of message header): 13(D)

Command function identifier (byte 55 of message header):

X'00' = to display

X'01' = INCLUDE

X'02' = REMOVE

Record format:

4018

0 (00)	PSEUDO DEVICE NAME
4 (04)	UNIT ID 1
8 (08)	
12 (C)	UNIT ID 2
16 (10)	
20 (14)	UNIT ID 3
24 (18)	
28 (1C)	UNIT ID 4
32 (20)	
36 (24)	UNIT ID 5
40 (28)	
44 (2C)	UNIT ID 6
48 (30)	
52 (34)	UNIT ID 7
56 (38)	
60 (3C)	UNIT ID 8
64 (40)	
68 (44)	UNIT ID 9
72 (48)	
76 (4C)	BLANKS

Fields:

Pseudo device name contains the name of the default pseudo device for the group.

Unit ID n contains terminal or task identifiers.

E.19 INPUT COMMAND (INCLUDE)

Command mnemonic: INPUT

Command identifier (byte 54 of message header): 14(E)

Command function identifier (byte 55 of message header):

X'01' = INCLUDE

Record format:

4019

0 (00)	
4 (04)	CONTROL TASK
8 (08)	DEVICE 1
12 (C)	DEVICE 2
16 (10)	DEVICE 3
20 (14)	DEVICE 4
24 (18)	DEVICE 5
28 (1C)	DEVICE 6
32 (20)	DEVICE 7
36 (24)	DEVICE 8
40 (28)	DEVICE 9
44 (2C)	DEVICE 10
48 (30)	DEVICE 11
52 (34)	DEVICE 12
56 (38)	DEVICE 13
60 (3C)	DEVICE 14
64 (40)	DEVICE 15
68 (44)	DEVICE 16
72 (48)	DEVICE 17
76 (4C)	DEVICE 18

Fields:

Control task

contains the name of the control task that will monitor the specified input devices.

Device number

contains the names of the physical devices assigned to input spooling.

E.20 INPUT COMMAND (REMOVE)

Command mnemonic: INPUT

Command identifier (byte 54 of message header): 14(E)

Command function identifier (byte 55 of message header):

X'02' = to REMOVE

Record format:

4062

0 (00)	DEVICE 1
4 (04)	DEVICE 2
8 (08)	DEVICE 3
12 (C)	DEVICE 4
16 (10)	DEVICE 5
20 (14)	DEVICE 6
24 (18)	DEVICE 7
28 (1C)	DEVICE 8
32 (20)	DEVICE 9
36 (24)	DEVICE 10
40 (28)	DEVICE 11
44 (2C)	DEVICE 12
48 (30)	DEVICE 13
52 (34)	DEVICE 14
56 (38)	DEVICE 15
60 (3C)	DEVICE 16
64 (40)	DEVICE 17
68 (44)	DEVICE 18
72 (48)	DEVICE 19
76 (4C)	DEVICE 20

Fields:

Device n

contains the name of an input device to be removed from the spooling environment.

E.21 INTERROGATE COMMAND

Command mnemonic: INTERROGATE

Command identifier (byte 54 of message header): 15(F)

Command function identifier (byte 55 of the message header):

- X'10' = QUEUE
- X'11' = QUEUE pd:
- X'12' = QUEUE form
- X'13' = QUEUE fd:
- X'20' = CONFIGURATION

Record format:

4020

0 (00)	VOLUME 1	
4 (04)	FILENAME 1	
8 (08)	FILENAME 1	
12 (C)	EXTENSION 1	ACCOUNT 1
16 (10)	VOLUME 2	
20 (14)	FILENAME 2	
24 (18)	FILENAME 2	
28 (1C)	EXTENSION 2	ACCOUNT 2
32 (20)	PSEUDO DEVICE	
36 (24)		
40 (28)		
44 (2C)		
48 (30)		
52 (34)		
56 (38)		
60 (3C)		
64 (40)		
68 (44)		
72 (48)		
76 (4C)		

Fields:

Volume 1	contains the volume name of the file for which information is to be displayed.
Filename 1	contains the name of the file for which information is to be displayed.
Extension 1	contains the name extension of the file for which information is to be displayed.
Account 1	contains the account number of the file for which information is to be displayed.
Volume 2	contains the volume name of the file for the formname to be displayed.
Filename 2	contains the name of a form about which information is to be displayed.
Extension 2	contains the name extension of the file with the formname information.
Account 2	contains the account number of the file with the formname information.
Pseudo device	contains the name of a pseudo device about which information is to be displayed.

E.22 PRIMARY COMMAND

Command mnemonic: PRIMARY

Command identifier (byte 54 of message header): 16(10)

Command function identifier (byte 54 of message header):

X'00' = to display

X'03' = to specify

Record format:

4061

0 (00)	CONTROL TASKNAME
4 (04)	
8 (08)	AUTHORIZATION IDENTIFIER (AID)
12 (C)	
16 (10)	BLANKS
20 (14)	
24 (18)	
28 (1C)	
32 (20)	
36 (24)	
40 (28)	
44 (2C)	
48 (30)	
52 (34)	
56 (38)	
60 (3C)	
64 (40)	
68 (44)	
72 (48)	
76 (4C)	

Fields:

Control taskname

contains the name of a control task.

Authorization Identifier
(AID)

contains the primary authorization
identifier.

E.23 PRINT COMMAND

Command mnemonic: PRINT

Command identifier (byte 54 of memory header): 17(11)

Command function identifier (byte 55 of message header):

X'03' = PRINT

Record format:

4022

0 (00)	VOLUME	
4 (04)	FILENAME	
8 (08)	FILENAME	
12 (C)	EXTENSION	15(F) ACCOUNT
16 (10)	RESERVED	
20 (14)	FORMNAME	
24 (18)	FORMNAME	
28 (1C)	RESERVED	
32 (20)	PSEUDO DEVICE	
36 (24)	OPTIONS	
40 (28)	NUMBER OF COPIES	
44 (2C)	PRIORITY	
48 (30)	BLANKS	
52 (34)		
56 (38)		
60 (3C)		
64 (40)		
68 (44)		
72 (48)		
76 (4C)		

Fields:

- Volume contains the volume name for the file to be printed.
- Filename contains the name of the file to be printed.
- Extension contains the extension of the file to be printed.

Account	contains the account number of the file to be printed.
Formname	contains the name of the form required for output.
Pseudo device	contains the name of the pseudo device the output is to be directed to.
Options	contains one or a combination of the following settings: Y'0001' = VFC (if NOVFC set) Y'0002' = NOVFC (if VFC set) Y'0004' = IMAGE (if NOIMAGE set) Y'0008' = NOIMAGE (if IMAGE set) Y'0010' = DELETE (if NODELETE set) Y'0020' = NODELETE (if DELETE set) Y'0040' = CHECKPOINT (if NOCHECKPOINT set) Y'0080' = NOCHECKPOINT (if CHECKPOINT set) Y'0100' = HOLD (if RELEASE set) Y'0200' = RELEASE (if HOLD set)
Number of copies	contains the number of copies to be printed.
Priority	contains the output priority of the file to be printed.

E.24 PRIORITY COMMAND

Command mnemonic: PRIORITY

Command identifier (byte 54 of message header): 18(12)

Command function identifier (byte 55 of the message header):

X'00' = to display

X'03' = to specify

Record format:

4023

0 (00)	PRIORITY
4 (04)	
8 (08)	
12 (C)	
16 (10)	
20 (14)	
24 (18)	
28 (1C)	
32 (20)	
36 (24)	
40 (28)	BLANKS
44 (2C)	
48 (30)	
52 (34)	
56 (38)	
60 (3C)	
64 (40)	
68 (44)	
72 (48)	
76 (4C)	

Fields:

Priority contains the default priority value.

E.25 PUNCH COMMAND

Command mnemonic: PUNCH

Command identifier (byte 54 of message header): 19(13)

Command function identifier (byte 55 of message header):

X'03' = PUNCH

Record format:

4024

0 (00)	VOLUME NAME	
4 (04)	FILENAME	
8 (08)	FILENAME	
12 (C)	EXTENSION	15(F) ACCOUNT
16 (10)	RESERVED	
20 (14)	RESERVED	
24 (18)	FORMNAME	
28 (1C)	RESERVED	
32 (20)	PSEUDO DEVICE	
36 (24)	OPTIONS	
40 (28)	NUMBER OF COPIES	
44 (2C)	PRIORITY	
48 (30)	BLANKS	
52 (34)		
56 (38)		
60 (3C)		
64 (40)		
68 (44)		
72 (48)		
76 (4C)		

Fields:

Volume	contains the volume name for the file to be punched.
Filename	contains the name of file to be punched.
Extension	contains the filename extension of the file to be punched.
Account	contains the account number of the file to be punched.
Formname	contains the name of the preprinted card required for output.
Pseudo device	contains the name of the pseudo device the output is to be directed to.
Options	contains one or a combination of the following settings: Y'0001' = VFC (if NOVFC set) Y'0002' = NOVFC (if VFC set) Y'0004' = IMAGE (if NOIMAGE set) Y'0008' = NOIMAGE (if IMAGE set) Y'0010' = DELETE (if NODELETE set) Y'0020' = NODELETE (if DELETE set) Y'0040' = CHECKPOINT (if NOCHECKPOINT set) Y'0080' = NOCHECKPOINT (if CHECKPOINT set) Y'0100' = HOLD (if RELEASE set) Y'0200' = RELEASE (if HOLD set)
Number of Copies	contains the number of copies to be punched.
Priority	contains the output priority of the file to be punched.

E.26 QUEUE COMMAND

Command mnemonic: QUEUE

Command identifier (byte 54 of message header): 20(14)

Command function identifier (byte 55 of the message header):

X'00' = to display

X'03' = to specify

Record format:

4025

0 (00)	VOLUME	
4 (04)	FILENAME	
8 (08)	FILENAME	
12 (C)	EXTENSION	15(F) ACCOUNT
16 (10)	BLANKS	
20 (14)		
24 (18)		
28 (1C)		
32 (20)		
36 (24)		
40 (28)		
44 (2C)		
48 (30)		
52 (34)		
56 (38)		
60 (3C)		
64 (40)		
68 (44)		
72 (48)		
76 (4C)		

Fields:

Volume	contains the volume name for the spool queue file.
Filename	contains the name of the file to be the queue file.
Extension	contains the file name extension for the spool queue file.
Account	contains the account number for the spool queue file.

E.27 SCTASK COMMAND

Command mnemonic: SCTASK

Command identifier (byte 54 of message header): 24(18)

Command function identifier (byte 55 of message header):

X'00' = to display

X'01' = to INCLUDE

X'02' = to REMOVE

Record format:

4027

0 (00)	CONTROL TASK
4 (04)	
8 (08)	SUBCONTROL TASK 1
12 (C)	
16 (10)	SUBCONTROL TASK 2
20 (14)	
24 (18)	SUBCONTROL TASK 3
28 (1C)	
32 (20)	SUBCONTROL TASK 4
36 (24)	
40 (28)	SUBCONTROL TASK 5
44 (2C)	
48 (30)	SUBCONTROL TASK 6
52 (34)	
56 (38)	SUBCONTROL TASK 7
60 (3C)	
64 (40)	SUBCONTROL TASK 8
68 (44)	
72 (48)	SUBCONTROL TASK 9
76 (4C)	

Fields:

Control task contains the name of the control task associated with the specified subcontrol tasks.

Subcontrol task contains the name of the subcontrol tasks.

E.28 SECONDARY COMMAND

Command mnemonic: SECONDARY

Command identifier (byte 54 in message header): 23(17)

Command function identifier (byte 55 of message header):

X'00' = to display

X'03' = to specify

Record format:

4028

0 (00)	TASKNAME
4 (04)	
8 (08)	PSEUDO DEVICE
12 (C)	
16 (10)	AUTHORIZATION IDENTIFIER
20 (14)	
24 (18)	
28 (1C)	
32 (20)	
36 (24)	
40 (28)	
44 (2C)	
48 (30)	BLANKS
52 (34)	
56 (38)	
60 (3C)	
64 (40)	
68 (44)	
72 (48)	
76 (4C)	

Fields:

- Taskname contains the name of a control task.
- Pseudo device contains the name of the default pseudo device.
- Authorization Identifier contains the secondary authorization identifier.

E.29 SPVOLUME COMMAND

Command mnemonic: SPVOLUME

Command identifier (byte 54 of message header): 25(19)

Command function identifier (byte 55 of message header):

X'00' = to display

X'01' = INCLUDE

X'02' = REMOVE

Record format:

4029

0 (00)	VOLUME 1
4 (04)	VOLUME 2
8 (08)	VOLUME 3
12 (C)	VOLUME 4
16 (10)	VOLUME 5
20 (14)	VOLUME 6
24 (18)	VOLUME 7
28 (1C)	VOLUME 8
32 (20)	VOLUME 9
36 (24)	VOLUME 10
40 (28)	VOLUME 11
44 (2C)	VOLUME 12
48 (30)	VOLUME 13
52 (34)	VOLUME 14
56 (38)	VOLUME 15
60 (3C)	VOLUME 16
64 (40)	VOLUME 17
68 (44)	VOLUME 18
72 (48)	VOLUME 19
76 (4C)	VOLUME 20

Fields:

Volume contains the name of volumes to be included or removed as spool volumes.

E.30 TRAILER COMMAND

Command mnemonic: TRAILER

Command identifier (byte 54 of message header): 26(1A)

Command function identifier (byte 55 of message header):

- X'10' = NONE
- X'20' = SINGLE
- X'40' = DOUBLE

Record format:

4030

0 (00)	DEVICE	
4 (04)	VOLUME	
8 (08)	FILENAME	
12 (C)		
16 (10)	EXTENSION	19(13) ACCOUNT
20 (14)	CHARACTER	
24 (18)		
28 (1C)		
32 (20)		
36 (24)		
40 (28)	BLANKS	
44 (2C)		
48 (30)		
52 (34)		
56 (38)		
60 (3C)		
64 (40)		
68 (44)		
72 (48)		
76 (4C)		

Fields:

Device	contains the name of the pseudo or physical device to which the TRAILER command pertains.
Volume	contains the volume name of the file that contains the trailer page specifications.
Filename	contains the name of the file that contains the trailer page specifications.
Extension	contains the filename extension of the file that contains the trailer page specifications.
Account	contains the account number for the file that contains the trailer page specifications.
Character	contains the character with which the letters of the trailer page are to be formed.

E.31 SUMMARY

Table E-1 provides a summary of prepared command information.

TABLE E-1 SUMMARY OF COMMAND IDENTIFIERS (BYTE 54)
AND COMMAND FUNCTION IDENTIFIERS (BYTE 55)
FOR PREPARED COMMANDS

COMMAND MNEMONIC	COMMAND IDENTIFIER (hexadecimal)	COMMAND FUNCTION IDENTIFIER (hexadecimal)	FUNCTION
ATTACH	1	3	To specify
BANNER	2	10 20 40	No banner page Single banner page Double banner page
BLOCK	3	0 3	To display To specify
CHANGE	4	3	To specify
CHECKPOINT	5	0 1 2	To display Turn on Turn off
CREATE	6	3	To specify
CTASK	7	0 1 2	To display To INCLUDE To REMOVE
DELETE	16	3	TO SPECIFY
DETACH	8	3	To detach
DEVICE	9	10 11 20 21 22 30 31 40 41 50	Forward space line Forward space page Backward space line Backward space page Rewind Free Restrict SUSPEND immediate SUSPEND delayed Continue
END	15	1 2	IMMEDIATE DELAYED
FLUSH	A	10 11 20	Flush fd Flush fd, DELETE Flush dev

COMMAND MNEMONIC	COMMAND IDENTIFIER	COMMAND FUNCTION IDENTIFIER	FUNCTION
FORM	B	0	To display
		1	INCLUDE
		2	REMOVE
		4	MOUNT
		14	MOUNT VERIFY
GLOBAL	C	0	To display
		3	To specify
GROUP	D	0	To display
		1	INCLUDE
		2	REMOVE
INPUT	E	0	To display
		1	INCLUDE
		2	REMOVE
INTERROGATE	F	10	QUEUE
		11	PD
		12	FORM
		13	FD
		20	Configuration
PRIMARY	10	0	To display
		3	To specify
PRINT	11	3	To print
PRIORITY	12	0	To display
		3	To specify
PUNCH	13	3	To punch
QUEUE	14	0	To display
		3	To specify
SCTASK	18	0	To display
		1	To INCLUDE
		2	To REMOVE
SECONDARY	17	0	To display
		3	To specify
SPVOLUME	19	0	To display
		1	To INCLUDE
		2	To REMOVE
TRAILER	1A	10	No trailer page
		20	Single trailer page
		40	Double trailer page

INDEX

/@INPUT control statement	2-1		
/@SUBMIT control statement	2-3		
A			
Acceptance messages			
function of	5-2		
Acceptance response message			
format	D-10		
AID. See authorization			
identifier.			
ASSIGN command	3-1		
ATTACH command	4-4		
	4-8		
	6-2		
prepared format	E-2		
service record format	D-20		
Attention messages	D-59		
function of	5-3		
SVC1 and 7	D-61		
types of	D-59		
Authority message format	D-65		
Authority messages			
function of	5-3		
Authorization identifiers			
function of	5-10		
primary AID	5-10		
secondary AID	5-10		
Authorization levels			
associated spooler			
commands	5-9		
definition of	1-6		
function of	5-8		
primary	1-6		
	5-1		
relation to task types	5-1		
secondary	1-6		
	5-1		
user	1-6		
	5-1		
B			
BANNER command	4-14		
	6-4		
prepared format	E-2		
BANNER pages			
contents of	4-13		
BLOCK command	2-2		
	2-3		
	3-1		
	5-9		
	6-7		
prepared format	E-5		
service record format	D-22		
C			
CHANGE command			4-5
			4-17
			5-9
			6-8
prepared format			E-6
CHECKPOINT command			4-14
			6-10
prepared format			E-8
service record format			D-24
CHECKPOINT feature			
turning off			4-15
use of			4-14
Command comparison			
SPL/32 vs. OS/32 Spooler			C-1
Command messages			5-5
			D-8
function of			5-2
Command summary			A-1
Communication with a task			
losing			D-2
terminating			D-2
Communication with spooler			
establishing			D-2
Console interface			
conventions			1-12
Control tasks			5-1
communication with the			
spooler			5-2
definition of			1-6
displaying			5-4
function			5-2
in the spooling			
environment			5-4
message types			5-2
normal termination of			D-2
removing			5-4
spooler communication			
with			5-3
CREATE command			6-12
prepared format			E-9
Creating pseudo devices			
explicitly			4-3
implicitly			4-3
Creating the spool queue			
example			3-4
explicitly			3-4
implicitly			3-4
CTASK command			5-4
			6-13
prepared format			E-10
service record format			D-26
D			
DELETE command			4-7
			6-15
prepared format			E-11

DETACH command	4-7	GLOBAL command (Continued)	
prepared format	6-16	prepared format	E-23
DEVICES command	E-12	service record format	D-30
prepared format	4-6	Global priority	4-16
Device contention	4-8	changing	4-16
Dialogue termination	6-17	Global pseudo device	
Displaying queue entries	E-13	definition of	4-2
according to preprinted	1-2	GROUP command	6-27
form requests	D-3	prepared format	E-23
by assigned pseudo device		service record format	D-32
by particular file	3-7		
displaying all	3-7	H	
information	3-8	Hardware requirements for	
displaying spooler		SPL/32	1-8
configuration	3-9		
Displaying spooler	3-8	I, J, K	
environment configurations	3-8	I/O device utilization	1-2
		Identification pages	
E		banner pages	4-13
END command	1-13	trailer pages	4-13
prepared format	6-20	use of	4-13
End control statement	E-15	Individual command service	
Ending a spooler task	2-4	response	D-20
Ending conventions for START	1-13	Initializing SPL/32	1-9
command file		INPUT command	2-1
Establishing input spooling	1-11		2-5
Exception message error codes	2-1	service record format	6-29
Exception message-message	D-56	INPUT command (INCLUDE)	D-34
related data		prepared format	E-25
Exception message-message	D-56	INPUT command (REMOVE)	
test		prepared format	E-26
Exception response messages	D-56	Input spooling	
error codes	5-5	control statements	2-1
function of	D-54	definition of	1-3
	D-56	ending	2-4
	5-3	establishing	2-1
		introduction	2-1
		removing devices	2-5
F		Input spooling control	
FLUSH command	4-9	statements	
prepared format	6-21	/@INPUT	2-1
FORM command	E-16	/@SUBMIT	2-3
service record format	4-11	ending	2-4
FORM command (INCLUDE)	6-23	stacking card files	2-4
prepared format	D-28	syntactically incorrect	2-2
FORM command (MOUNT/MOUNT			2-4
VERIFY)		INTERROGATE command	3-5
prepared format	E-18	functional discussion of	6-31
FORM command (REMOVE)		prepared format	E-27
prepared format	E-21	service record format	D-36
FORM command (REMOVE)		Interrogating the spool queue	3-5
prepared format	E-20		
		L	
G		Linking SPL/32	1-10
General attention message		Loading SPL/32	1-9
field entries	D-64		
GLOBAL command	4-2		
	6-26		

M,N	
Memory requirements for SPL/32	1-8
Message formats	D-8
acceptance response	D-10
command message	D-8
request for mnemonic table	D-12
service response message	D-14
SPL/32 to task	D-14
task to SPL/32	D-8
Message handling	
attention messages	D-59
authority message format	D-65
communication with spooler	D-2
dialogue termination	D-3
exception response messages	D-54
general attention message field entries	D-64
message formats	D-8
message header	D-4
message overview	D-1
message recovery procedures	D-68
message routing	D-3
mnemonic message format	D-64
response routing	D-3
uniform spooler control interface	D-1
Message header	D-4
Message overview	D-1
Message recovery procedures	D-68
error status	D-69
recovery information	D-69
Message routing	D-3
Message summary	B-1
Mnemonic message format	D-64
Mnemonic messages function of	5-3
O	
OID. See origin identifiers.	
Origin identifiers function of	5-10
Output devices	
altering the status of	4-8
control of	4-8
Output priority	
altering	4-17
assignment of	4-16
global priority	4-16
user priority	4-16
Output spooling	
definition of	1-3
features of	4-1
global pseudo device	4-2
introduction	4-1
pseudo device	4-2

P	
Physical devices	
removing from spooling environment	4-10
Preprinted forms	
associating with a physical device	4-11
characterizing	4-11
naming	4-11
removing from spooling environment	4-12
use of	4-10
verify the alignment of	4-12
Primary AID	
function of	5-10
Primary authorization level	5-1
function of	5-8
PRIMARY command	5-8
prepared format	6-33
service record format	E-29
PRINT command	D-42
prepared format	4-5
service record format	6-35
Print locations	E-30
establishing unique defaults	4-6
PRIORITY command	4-16
prepared format	6-37
service record format	E-32
Pseudo devices	D-44
associating with physical devices	4-2
creating	4-4
definition of	4-3
deleting from spooling environment	1-3
directing output to physical devices	4-2
limiting default access to	4-7
optional output parameters	4-6
PUNCH command	4-5
prepared format	4-5
service record format	6-38
service record format	E-33
Q	
QUEUE command	3-4
prepared format	6-40
service record format	E-35
service record format	D-46
R	
Request for mnemonic table messages	D-12
function of	5-2

Response routing spooler to task	D-3	SPL/32 commands (Continued)	
		DETACH	6-16
		DETACH prepared format	E-12
		DEVICE	6-17
		DEVICE prepared format	E-13
		END	6-20
		END prepared format	E-15
		FLUSH	6-21
		FLUSH prepared format	E-16
		FORM	6-23
		FORM prepared format	E-18
		GLOBAL	6-26
		GLOBAL prepared format	E-23
		GROUP	6-27
		GROUP prepared format	E-24
		INPUT	6-29
		INPUT prepared format	E-25
		INTERROGATE	6-31
		INTERROGATE prepared format	E-27
		prepared summary	E-43
		PRIMARY	6-33
		PRIMARY prepared format	E-29
		PRINT	6-35
		PRINT prepared format	E-30
		PRIORITY	6-37
		PRIORITY prepared format	E-32
		PUNCH	6-38
		PUNCH prepared format	E-33
		QUEUE	6-40
		QUEUE prepared format	E-35
		SCTASK	6-42
		SCTASK prepared format	E-37
		SECONDARY	6-44
		SECONDARY prepared format	E-39
		SPOOLFILE	6-45
		SPVOLUME	6-48
		SPVOLUME prepared format	E-40
		summary	A-1
		TRAILER	6-50
		TRAILER prepared format	E-41
		SPL/32 commands prepared format	
		FORM (MOUNT/MOUNT VERIFY)	E-22
		FORM (REMOVE)	E-20
		GLOBAL	E-23
		GROUP	E-23
		INPUT (INCLUDE)	E-25
		INPUT (REMOVE)	E-26
		INTERROGATE	E-26
		PRIMARY	E-29
		PRINT	E-30
		PRIORITY	E-32
		PUNCH	E-33
		QUEUE	E-35
		SCTASK	E-37
		SECONDARY	E-39
		SPVOLUME	E-40
		summary of prepared commands	E-43
		TRAILER	E-41
		SPL/32 requirements	1-8
		hardware	1-8
		memory	1-8
		software	1-8
S			
SCTASK command	6-42		
prepared format	E-37		
service record format	D-48		
Secondary AID			
function of	5-10		
Secondary authorization level	5-1		
function of	5-8		
SECONDARY command	5-8		
prepared format	6-44		
service record format	E-39		
Service response message format	D-50		
buffer set	D-15		
individual command			
service	D-20		
Service response message formats	D-14		
ATTACH	D-20		
BLOCK	D-22		
CHECKPOINT	D-24		
CTASK	D-26		
FORM	D-28		
GLOBAL	D-30		
GROUP	D-32		
INPUT	D-34		
INTERROGATE	D-36		
PRIMARY	D-42		
PRIORITY	D-44		
QUEUE	D-46		
SCTASK	D-48		
SECONDARY	D-50		
SPVOLUME	D-52		
Service response messages function of	5-5		
Software requirements for SPL/32	1-8		
SPL/32			
initialization	1-9		
loading procedures	1-9		
SPL/32 command message summary	B-1		
SPL/32 commands	6-1		
ATTACH	6-2		
ATTACH prepared format	E-2		
BANNER	6-4		
BANNER prepared format	E-3		
BLOCK	6-7		
BLOCK prepared format	E-5		
CHANGE	6-8		
CHANGE prepared format	E-6		
CHECKPOINT	6-10		
CHECKPOINT prepared format	E-8		
CREATE	6-12		
CREATE prepared format	E-9		
CTASK	6-13		
CTASK prepared format	E-10		
DELETE	6-15		
DELETE prepared format	E-11		

PERKIN-ELMER
Technical Systems Division

D O C U M E N T A T I O N C H A N G E N O T I C E

The purpose of this documentation change notice (DCN) is to provide a quick and efficient way of making technical changes to manuals before they are formally updated or revised. The manual affected by these changes is:

48-056 F00 R00 SPL/32 Spooler System Administration Reference
 Manual

- Page 1-7

After the last paragraph in Section 1.5.1, please insert:

NOTE

At this time, the Perkin-Elmer Network (PENnet) and SPL/32 are incompatible. Users who want the capabilities of PENnet must configure their system with OS/32 Spooler support. See the OS/32 System Generation/32 (SYSGEN/32) Reference Manual.

- Page B-1

Before the first message, please insert the following:

CANNOT ASSIGN DISPLAY MESSAGE FILE

The display message file (SPLDISPL.MES) cannot be assigned.

- Page B-1

Immediately after the above inserted message, please insert:

CORRUPTED QUEUE

The Spooler cannot read the spool queue because the queue entry links have been corrupted.

- Page B-4

After the sixth message, please insert:

OS DOES NOT SUPPORT SPL32

The operating system was configured without support for SPL/32. Re-sysgen the operating system to include SPL/32 support.

- Page D-59

Message code 69. Please change this message to read:

OS DOES NOT SUPPORT SPL32

- Page D-59

After message code 75, please insert the following messages:

77 CORRUPTED QUEUE

78 CANNOT ASSIGN DISPLAY MESSAGE FILE