SPERRY UNIVAC
COMPUTER SYSTEMS

PUBLICATIONS REVISION

90/30, 90/30B, 90/40

Operations Handbook

Operator Reference

UP-8072 Rev. 4

This Library Memo announces the release and availability of "SPERRY UNIVAC® 90/30, 90/30B, 90/40 Data Processing Systems Operations Handbook Operator Reference", UP-8072 Rev. 4. This is a Standard Library Item (SLI).

This revision reflects the current status of the OS/3 operating system, including changes and enhancements.

The standard operating procedures have been reorganized for ease of reference. Commands and messages are grouped according to the function they provide. For job processing, the commands and messages are described according to the status of the job within the system.

The standard operating procedures contain the following new information:

- Procedures for initializing a spooled job
- Procedures for scheduling jobs for execution, including:
  - Ability to specify only new jobs or only currently residing (old) jobs within a scheduling queue
  - Ability to specify only remotely entered jobs or only locally entered jobs
- Procedures for controlling physical I/O control system error logging
- Procedures for saving the accumulated console logs
- Enhanced procedures for utilizing the spooling services
- Enhanced procedures for utilizing the system utility services

The system definition has been revised to include the 8413 Diskette subsystem and the 8424 and 8425 Disk subsystems.

In addition, new command formats and minor technical changes have been included.

Destruction Notice: This revision supersedes and replaces "SPERRY UNIVAC 90/30, 90/30B, 90/40 Data Processing System Operations Handbook for Operators" UP-8072 Rev. 3, released on Library Memo dated May, 1978. Please destroy all copies of UP-8072 Rev. 3 and/or its Library Memo.

Additional copies may be ordered by your local Sperry Univac Representative.

LIBRARY MEMO ONLY	LIBRARY MEMO AND ATTACHMENTS	THIS SHEET IS
Mailing Lists BZ, CZ and MZ	Mailing Lists 20, 21, 75 and 76 (Cover and 158 pages)	Library Memo
		RELEASE DATE: June, 1979

SPERRY UNIVAC 90/30, 90/30 B, and 90/40 Data Processing Systems

Operations Handbook

Operator Reference

SPERRY UNIVAC

SPERRY UNIVAC
90/30, 90/30 B, and
90/40 Data Processing Systems
Operations Handbook
Operator Reference

This document contains the latest information available at the time of preparation. Therefore, it may contain descriptions of functions not implemented at manual distribution time. To ensure that you have the latest information regarding levels of implementation and functional availability, please contact your local Sperry Univac representative.

Sperry Univac reserves the right to modify or revise the content of this document. No contractual obligation by Sperry Univac regarding level, scope, or timing of functional implementation is either expressed or implied in this document. It is further understood that in consideration of the receipt or purchase of this document, the recipient or purchaser agrees not to reproduce or copy it by any means whatsoever, nor to permit such action by others, for any purpose without prior written permission from Sperry Univac.

Sperry Univac is a division of the Sperry Rand Corporation.

This document was prepared by Systems Publications using the SPERRY UNIVAC UTS 400 Text Editor. It was printed and distributed by the Customer Information Distribution Center (CIDC), 555 Henderson Rd., King of Prussia, Pa., 19406.

PRINTED IN U.S.A.

PAGE STATUS SUMMARY

ISSUE: UP-8072 Rev. 4

Part/Section	Page Number	Update Level
Cover		
Title Page		
PSS	1	
Preface	1, 2	
Contents	1 thru 6	
1	1 thru 16	
2	1, 2	
3	1 thru 9	
4	1 thru 33	
5	1 thru 31	
6	1, 2	
7	1 thru 34	
Appendix A	1, 2	
Appendix B	1 thru 6	
Index	1 thru 11	
User Comment Sheet		

Part/Section	Page Number	Update Level

Part/Section	Page Number	Update Level

Preface

This handbook is designed to instruct and guide the operator in the procedures required to operate the SPERRY UNIVAC 90/30, 90/30 B, and 90/40 Systems under control of the SPERRY UNIVAC Operating System/3 (OS/3). Its intended audience is the operator with a basic knowledge of data processing operations but without experience on SPERRY UNIVAC systems.

One other document relating to the operation of the 90/30, 90/30 B, and 90/40 systems under control of OS/3 is the system messages programmer/operator reference, UP-8076 (current version). This reference describes all the system messages you could encounter while operating the 90/30, 90/30 B, or 90/40 data processing system and the appropriate responses, when necessary.

This operations handbook is organized as follows:

- SECTON 1. SYSTEM DEFINITION

  Briefly describes the minimum and maximum hardware configurations for the 90/30, 90/30 B, and 90/40 systems, along with a brief description of the OS/3 software available for these systems.

- SECTION 2. SYSTEM TURN-ON AND TURN-OFF PROCEDURES

  Provides system and special turn-on and turn-off procedures.

- SECTION 3. SYSTEM INITIALIZATION PROCEDURES

  Provides system procedures for control storage load and initial program load.

- SECTION 4. JOB PROCESSING PROCEDURES

  Provides job processing procedures performed by the system operator.

- SECTION 5. SPOOLING SERVICES

  Describes spooling service operations, need for spooling, and how it is accomplished. Procedures provided include the use of general spooling commands and messages, input reader commands, output writer commands and messages, and spool file processing aids.

- SECTION 6. INTEGRATED COMMUNICATIONS ACCESS METHOD (ICAM) PROCEDURES

  Describes the ICAM communications tasks, which include the commands required to load ICAM message control program modules, and the message instructions required by the operator to facilitate communications.

- SECTION 7. SYSTEM UTILITY SERVICES

  Describes the system utility symbiont (SL$$SU) provided by OS/3 to initialize tapes, disks, and diskettes; copying card, tape, disk, or diskette files and volumes; and many other routine utility functions.

- APPENDIX A. MESSAGE AND COMMAND FORMAT CONVENTIONS

  Describes the conventions used to illustrate the message and command formats presented in this handbook and other aids deemed necessary to help you perform your job.

- APPENDIX B. OPERATING PROCEDURES FOR THE 9200/9300 SERIES SUBSYSTEM

  Describes the operation of the 9200/9300 processor when used as a subsystem (remote station) to the 90/30, 90/30 B, or 90/40 systems.

Contents

6. INTEGRATED COMMUNICATIONS ACCESS METHOD (ICAM) PROCEDURES

7. SYSTEM UTILITY SERVICES

INDEX

USER COMMENT SHEET

FIGURES

TABLES

1. System Definition

1.1. GENERAL

The SPERRY UNIVAC 90/30, 90/30 B, and 90/40 Data Processing Systems are general-purpose, disk-oriented computers designed to function in many different data processing environments with equal operating efficiency. This efficiency is achieved through the use of the SPERRY UNIVAC Operating System/3 (OS/3), a multiprogramming software system specifically designed to make maximum use of the capabilities of the system hardware. The 90/30, 90/30 B, and 90/40 systems have a similar outward appearance. Figure 1–1 illustrates one version of the 90/30 system. Different versions of the systems are assembled from system configurations (1.2).

Figure 1—1. 90/30 System

1.2. SYSTEM CONFIGURATION

Diagrams of the basic 90/30 system and its expanded hardware options are presented in Figures 1–2 and 1–3, respectively. The basic and expanded hardware configurations for the 90/30 B system are presented in Figures 1–4 and 1–5, respectively. The basic and expanded hardware configurations for the 90/40 system are presented in Figures 1–6 and 1–7, respectively.

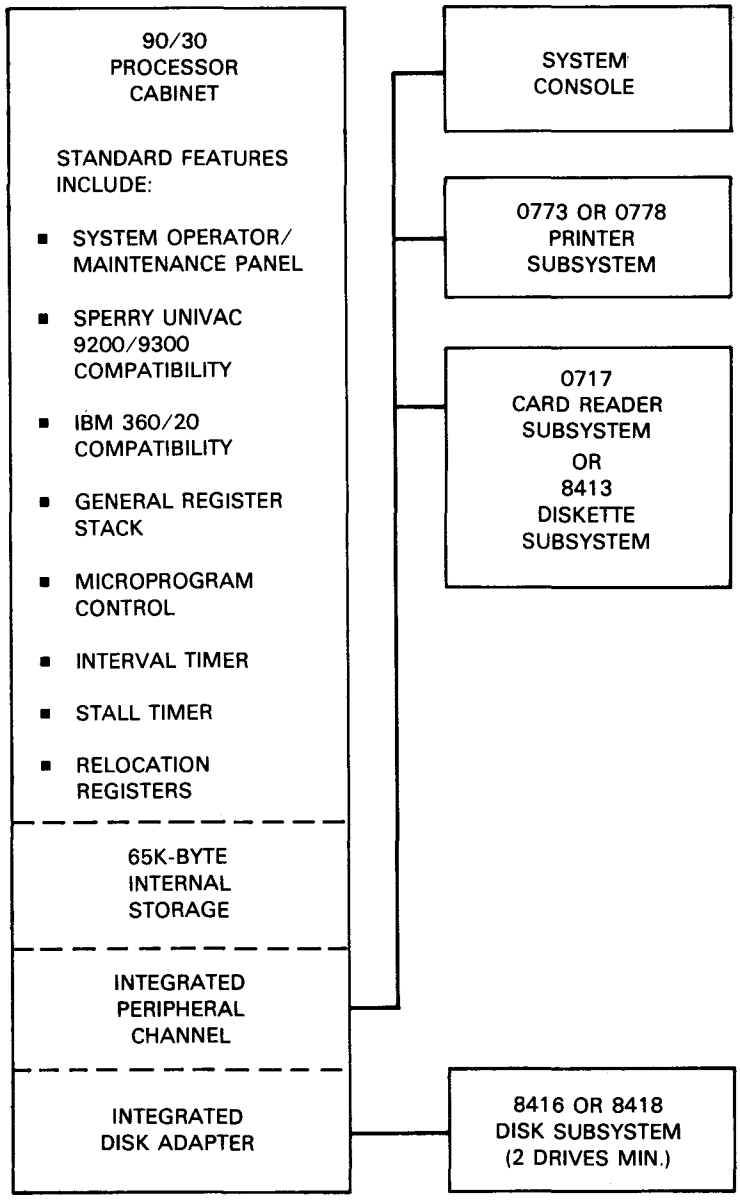


Figure 1—2. Basic 90/30 System Configuration

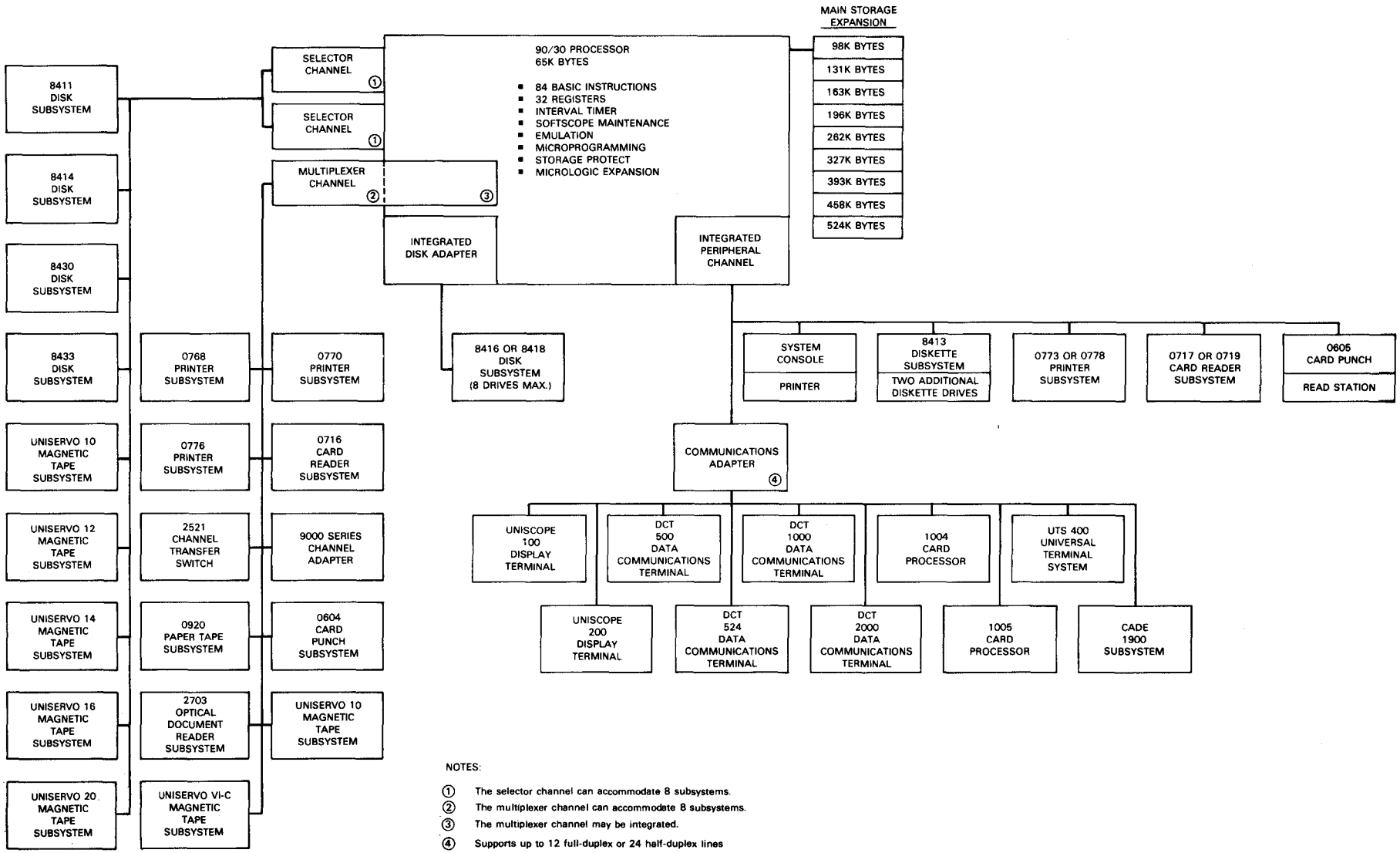


Figure 1—3. 90/30 System Configuration with All Hardware Options Shown

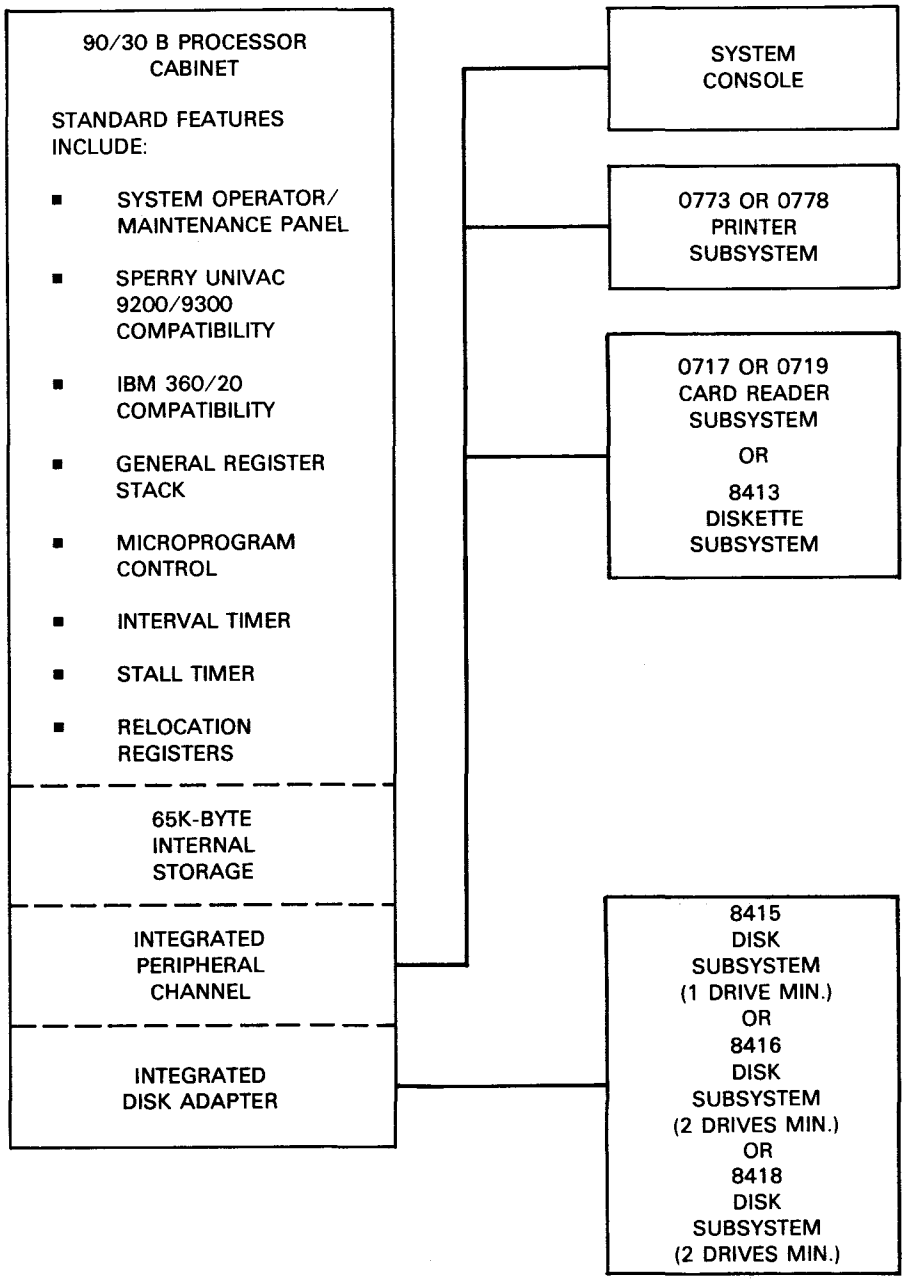


Figure 1—4. Basic 90/30 B System Configuration

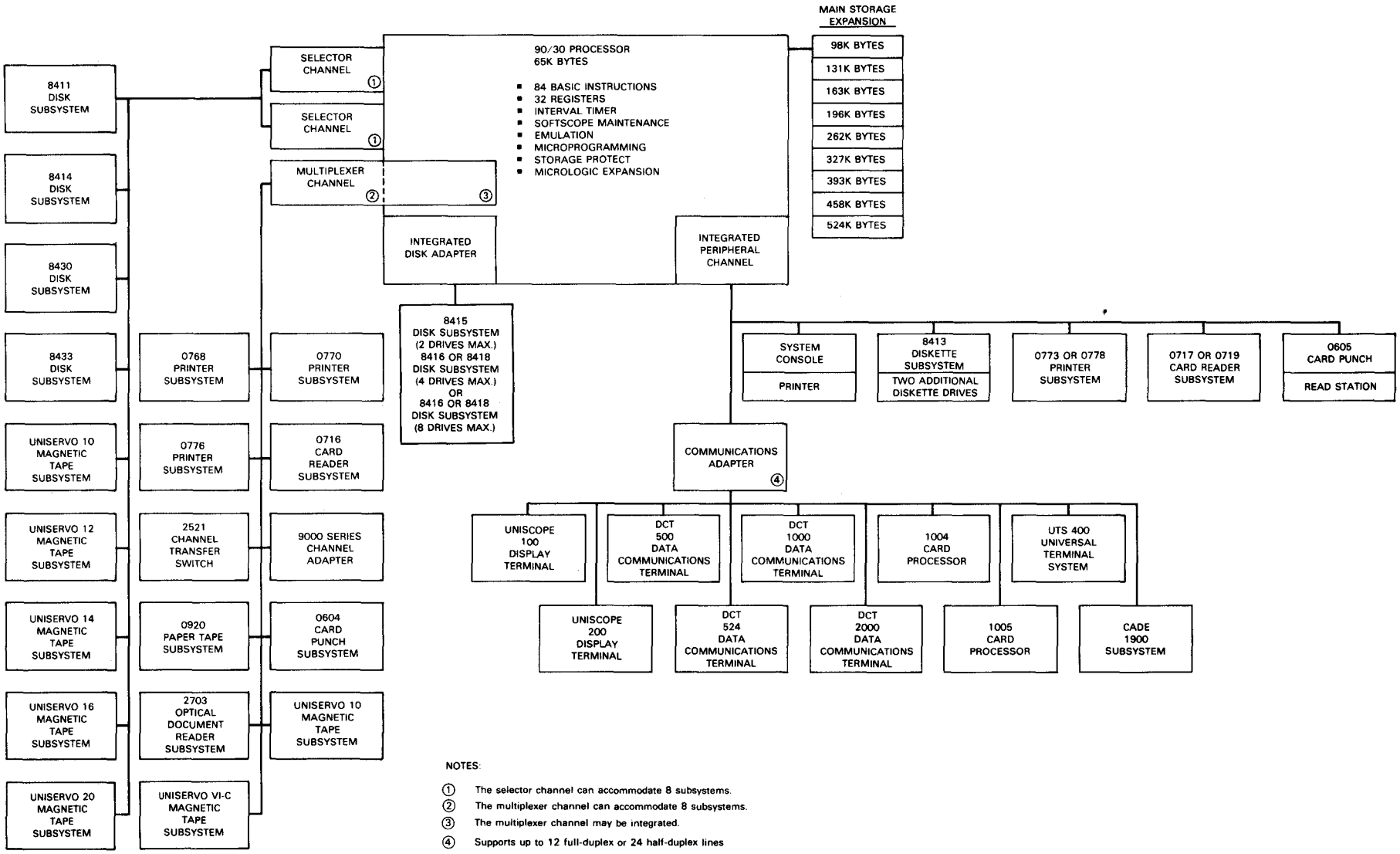


Figure 1—5. 90/30 B System Configuration with All Hardware Options Shown

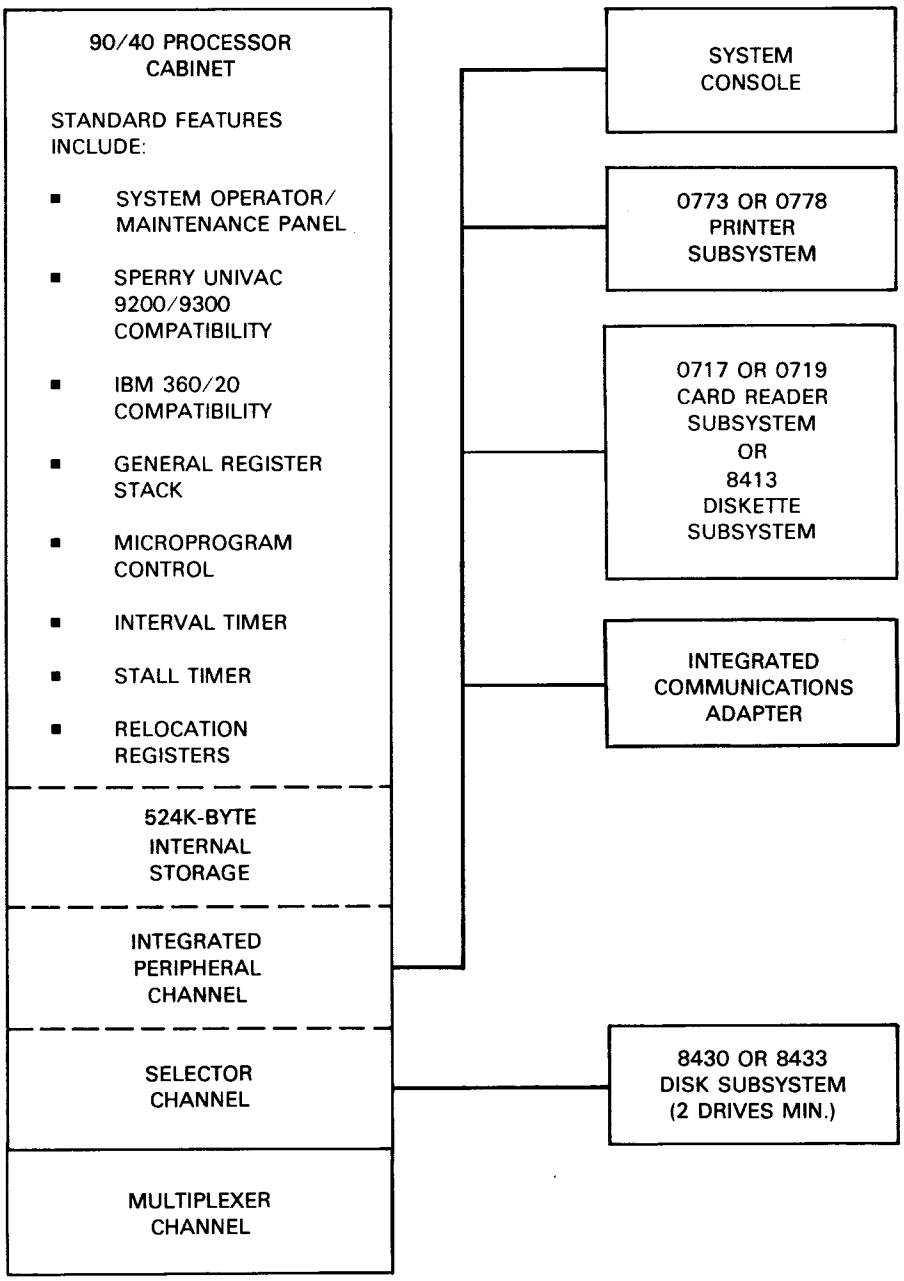


Figure 1—6. Basic 90/40 System Configuration

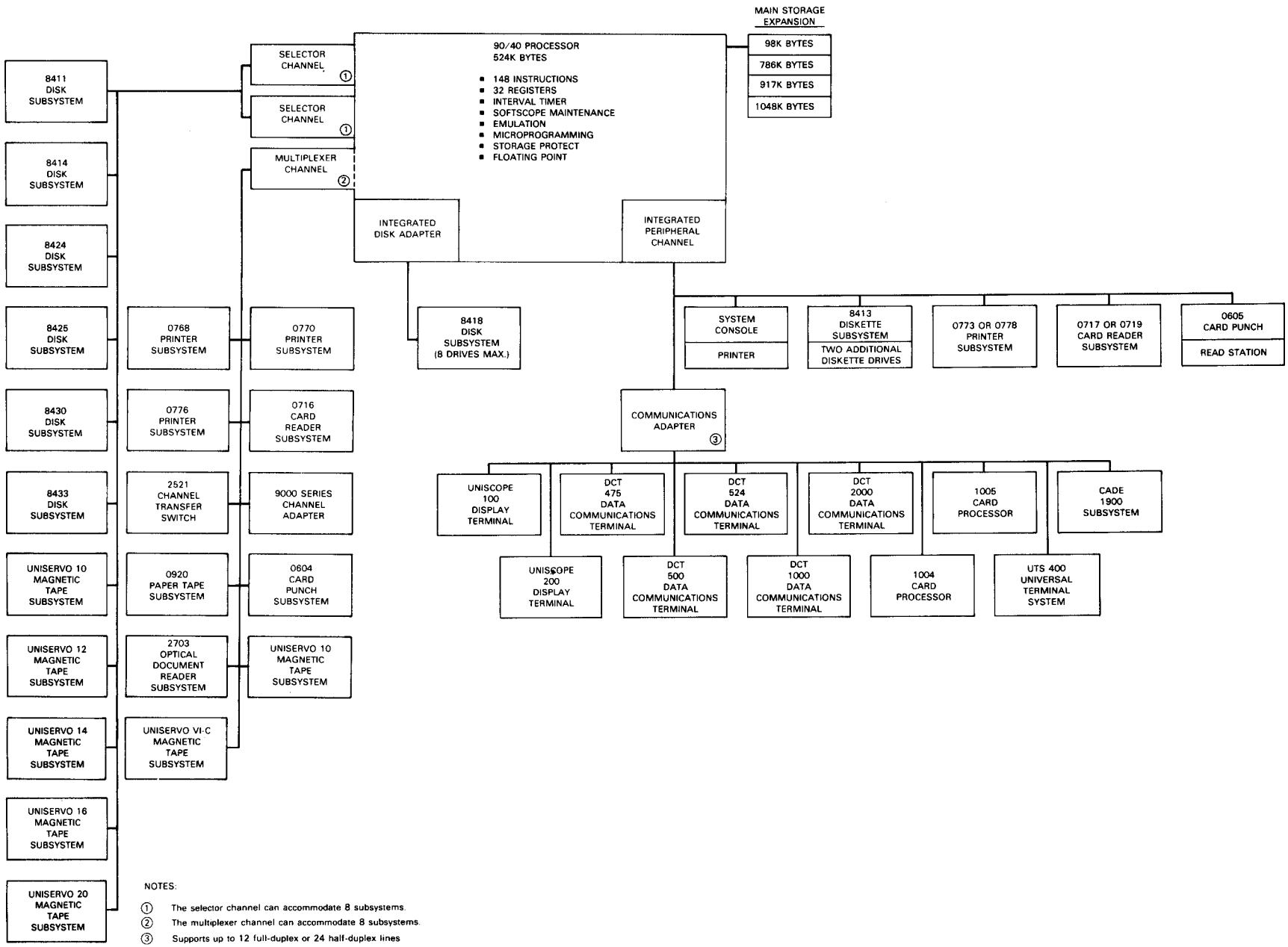


Figure 1—7. 90/40 System Configuration with All Hardware Options Shown

1.3. CENTRAL HARDWARE

The central hardware of a basic 90/30, 90/30 B, or 90/40 system consists of the processor cabinet and a system console, which is a modified UNISCOPE 100 Display Terminal. The central hardware for an expanded system may also include an external storage cabinet, an I/O expansion cabinet, a communications output printer (COP), and the processor cabinet optional features as shown in Figures 1-3, 1-5, and 1-7. All of the controls and input/output (I/O) channels required to process a job in the system are built into the hardware components. Enhanced system performance is obtained with the addition of the optional features mentioned.

A brief description of each of the functional components comprising the central hardware is provided in the following paragraphs.

1.3.1. Processor

The processor is a general-purpose, microprogram-controlled processor that includes the following:

- Basic instruction set (90/30, 90/30 B) or full instruction set (90/40)
- 32 general registers, 8 working registers, and 8 floating-point registers (optional)
- Interval timer
- Stall timer
- Integrated peripheral channel
- Relocation registers
- Integrated disk adapter
- Operator/maintenance panel
- 65K bytes main storage (expandable to 524K bytes) on the 90/30 and 90/30 B systems, or 524K bytes main storage (expandable to 1048K bytes) on the 90/40 system.
- Input/output control section

1.3.1.1. Main Storage

Main storage is of the semiconductor type with a 600-nanosecond half-word read/write cycle time for the 90/30 and 90/30 B systems. Main storage cycle time for the 90/40 system is 500 nanoseconds for half-word read/write. Main storage is constructed in modular form and is packaged as an internal part of the processor. Periodic refreshing of main storage is required to ensure data integrity. This refreshing of storage occurs automatically within the system. Power losses experienced by the system result in loss of all data in main storage.

1.3.1.2. Input/Output Control Section

The input/output control section initiates, directs, and monitors the transfer of data between main storage and the peripheral subsystems. After an I/O instruction is initiated, the data is transferred autonomously of other processor functions; i.e., the I/O and the processor operate concurrently. The I/O control section is the processor interface to the integrated peripheral channel, the integrated disk adapter, and the selector and multiplexer channels.

1.3.1.3. Integrated Peripheral Channel

The integrated peripheral channel (IPC) coordinates all information transfers between main storage and the integrated peripheral devices: system console, card reader, card punch, printer, diskette subsystem, and communications adapter. The IPC is a half-duplex channel that transfers commands, data, status, and sense information. Input/output activity is initiated by the processor upon issuance of a start I/O instruction to IPC. This instruction results in the transfer of a command to the control logic of a specific peripheral device. The command specifies the type of operation to be performed and is executed on an individual basis. The high transfer rate of the IPC permits simultaneous operation of all integrated peripherals.

1.3.1.4. Integrated Disk Adapter

The integrated disk adapter (IDA) acts as a combination channel and control unit.

On the 90/30 system, the IDA is designed to operate with a minimum of two and a maximum of eight 8416 or 8418 disk drive units, in any combination.

The 90/30 B system IDA operates with a minimum of one 8415 or two 8416 or 8418 disk drive units, and a maximum of:

- two 8415 and four 8416 or 8418 disk drive units; or
- eight 8416 or 8418 disk drive units.

In all cases, 8416 and 8418 units may be used in any combination.

The 90/40 system IDA is an optional feature because other disk subsystems are available to supply the two minimum disk drive units required. The IDA on the 90/40 system is designated to operate with a maximum of eight disk drive units.

1.3.1.5. Micrologic Expansion Feature

The micrologic expansion feature for the 90/30 and 90/30 B systems provides a repertoire of 64 additional instructions, four registers (each 64 bits long), and expanded control storage. It provides micrologic for execution of 44 floating-point instructions in both long and short, normalized and unnormalized formats, and micrologic for the execution of 20 additional nonprivileged instructions.

The 90/40 system has a repertoire of 148 basic instructions consisting of 135 nonprivileged and 13 privileged instructions. Also included are two privileged instructions used for storage protection and five emulation instructions (three valid in the 9200/9300 mode of operation and two valid in the 360/20 mode of operation).

1.3.1.6. Storage Protection Feature

The storage protection feature provides read/write protection on access to main storage and two additional privileged instructions (SSK, ISK). It protects up to 524,288 bytes of main storage for the 90/30 and 90/30 B systems, and up to 1,048,576 bytes of main storage for the 90/40 system.

1.3.1.7. Storage Expansion Feature

The storage expansion feature in the 90/30 and 90/30 B systems provides for increasing the size of main storage up to 524K bytes, by either 32K- or 64K-byte increments. In the 90/40 system, the storage expansion feature allows main storage to be increased up to 1048K bytes in 131K-byte increments.

1.3.1.8. Integrated Communication Adapter Feature

The integrated communication adapter feature provides for interfacing the IPC with a communication adapter. This feature is included in the basic 90/40 system configuration.

1.3.1.9. Integrated Multiplexer Channel Feature

The integrated multiplexer channel feature on the 90/30 and 90/30 B systems provides I/O capability between the processor and up to eight subsystems with a throughput rate of 83K bytes per second. The 90/30 and 90/30 B systems use a single multiplexer channel that may or may not be integrated, depending on the system configuration. If the system configurations include an I/O expansion cabinet, this feature cannot be used.

1.3.2. System Console

The system console provides the main interface for operator interaction with the processor. The system console is a modified UNISCOPE 100 Display Terminal, which accepts data from the keyboard of the console control unit, displays the data, and transfers the data to the integrated peripheral channel.

Data entered into the keyboard is displayed on the screen in a 64-character-per-line by 16-line format, providing a total display of 1024 characters. Displayable characters consist of the 64-character (including space) ASCII set plus control characters.

1.3.3. Communications Output Printer (COP)

The COP is a freestanding auxiliary output device for the system console. Capable of printing at a maximum rate of 30 characters per second, the COP can produce from one to six printed copies on edge-sprocketed forms 11 inches (27.9 cm) long and 3-5/8 inches (9.19 cm) wide to 14-7/8 inches (37.76 cm) wide. Operation is asynchronous. The COP requires only ac power connection and an interface connection to the system console.

1.3.4. External Storage Cabinet

External storage cabinets are used in early production models of the 90/30 system to extend the main storage capacity of their processor cabinets to 256K bytes. Processor cabinets up to and including serial number 746 can only house up to 128K bytes of main storage. When one of these early processors uses more than 128K bytes of main storage, the first 64K bytes are installed in the processor cabinet, and the remainder, up to 192K, is placed in an external storage cabinet.

All 90/30 systems using more than 256K bytes of main storage have serial number 747 or above. These newer processor cabinets can house up to 524K bytes of main storage and thus eliminate the need for an external storage cabinet.

The 90/30 B system can house up to 524K bytes without an external storage cabinet.

The 90/40 system can house up to 1024K bytes without an external storage cabinet.

1.3.5. I/O Expansion Cabinet

The I/O expansion cabinet for either system provides increased processor I/O capability by providing up to two selector channels and one multiplexer channel. Addition of these channels allows standard peripheral subsystems to operate with the system, in addition to the integrated peripheral subsystems. The basic 90/40 system configuration includes an I/O expansion cabinet with one selector channel and one multiplexer channel.

1.3.5.1. Selector Channels

Each selector channel controls the exchange of information between up to eight subsystems and processor main storage. The selector channels operate in the burst mode. (For example, one of eight possible subsystems retains control of the interface for the duration of its I/O operation. Simultaneously, other subsystems can be executing previously initiated operations that do not involve data transfer over the I/O interface.) The processor initiates all I/O operations to the selector channel and the specific subsystem connected to the channel. When the operation is successfully initiated, the channel maintains control of the data transfers between main storage and the subsystem independently of the processor. Upon completion of the I/O operation, the status of the channel and the subsystem is presented to the processor. One or two selector channels may be added to an expanded 90/30 or 90/30 B system configuration; one selector channel is included in the basic 90/40.

1.3.5.2. Multiplexer Channel

The multiplexer channel is similar in operation to the selector channel except that it operates in multiplexed mode. That is, the channel services several concurrently operating subsystems by assigning the input/output interface to a subsystem only long enough to transfer one or a few bytes of information. The multiplexer channel controls up to eight subsystems and initiates all input/output operations by issuing input/output instructions to a selected subchannel and subsystem. When the operation is successfully initiated, the multiplexer channel controls the flow of data between the main storage and the subsystem, independent of the processor. At the completion of the input/output operation, the status of the multiplexer channel is presented to the processor. One multiplexer channel, either integrated or with an I/O expansion cabinet (external) may be added to an expanded 90/30 or 90/30 B system configuration; one external multiplexer channel is included in the basic 90/40.

1.4. INPUT/OUTPUT SUBSYSTEMS

The I/O subsystems available for use with the 90/30, 90/30 B, and 90/40 systems include disk subsystems, magnetic and paper tape subsystems, high-speed printer subsystem, card reader and punch subsystems, an optical document reader subsystem, and a communications adapter.

1.5. ASSOCIATED HARDWARE MANUALS

The subsystems available for use with the system, along with their respective hardware references, are listed in Table 1–1. These references describe each subsystem from the hardware standpoint, giving the function, programming information, turn-on and turn-off procedures, use of the operating controls and indicators, recovery procedures, and directions for loading and unloading such subsystems as the disk units, magnetic tape units, card readers, and printers.

Table 1—1. System Hardware Documentation

Processor / Device	Document Number (current version)			
	General Description, UP-	Subsystem/ Programmer Reference, UP-	Operator Reference, UP-	Programmer/ Operator Reference, UP-
Processors			7781	
9200/9300	8547	8460	8459	
90/30, 90/30 B, 90/40		8052	8097 8522	
Integrated Peripheral Channel (IPC) 90/30, 90/30 B, 90/40		8041		
Card Punch				
0604		7772	7773	
0605	8192		8088	
Card Reader				
0716		7918	7921	
0717	8196		8089	
0719	8378	8431	8429	
Printer				
0768			7931	7688
0770		8016	7938	
0773	8191		8086	
0776	8354	8441	8250	
0778	(in process)		(in process)	
8541 (COP)	7939			
Magnetic Tapes				
VI-C				7644
10/14	8206	8205	8207	
12/16			7882	7661
U20			7956	
Disks				
8411	7605	7977	7802	
8414	7691	7977	7802	
8415				
8416				
8418		8362	8361	
8424			7983	
8425			8230	
8430		8344	8343	8450
8433		8344	8343	8590
Diskette				
8413	8463		8490	
Paper Tape				
0920	7595	7998	7830	
Optical Document Reader (ODR)				
2703	7710	7993	7994	

1.6. OPERATING SYSTEM

Operating System/3 (OS/3) is used with the 90/30, 90/30 B, and 90/40 systems. OS/3 (Figure 1-8) is composed of a group of major programs: supervisor, job control, data management, integrated communications access method, language processors, system service programs, emulators and transition aids, information management system, data base management system, and application programs.

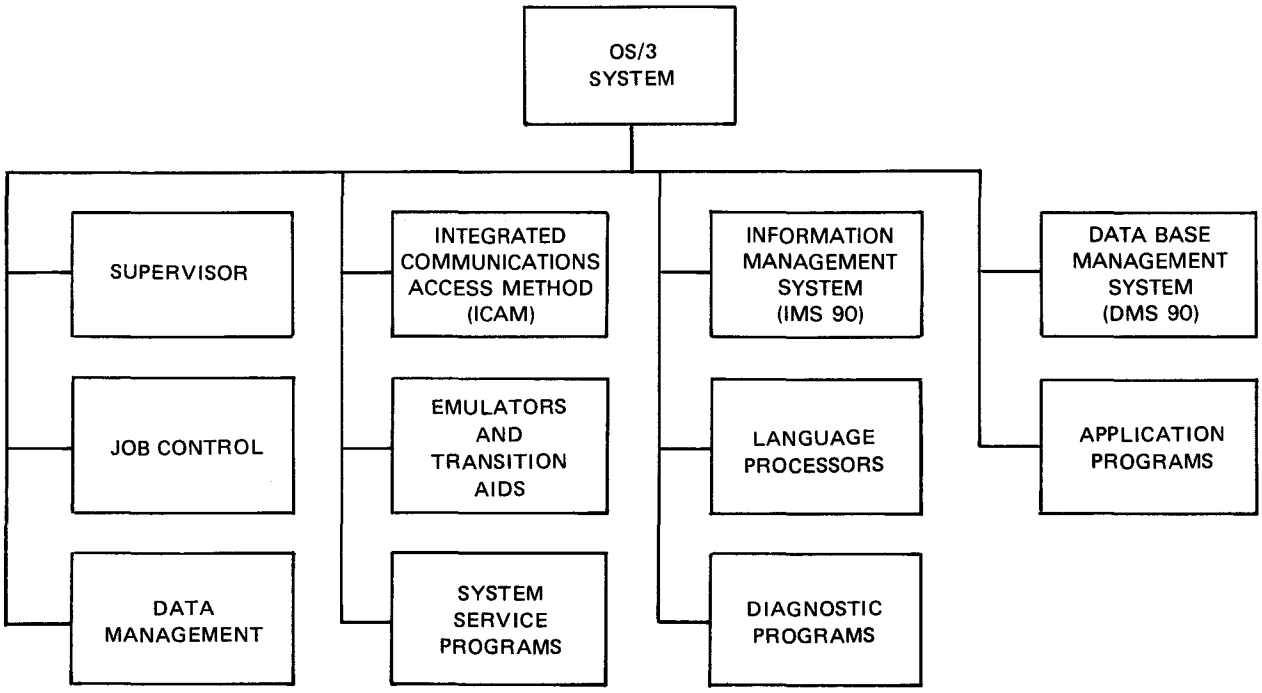


Figure 1—8. Operating System/3 Components

1.6.1. Supervisor

The supervisor is the part of the operating system that interfaces with the user programs to provide the necessary control for the optimum utilization of the system hardware and software. It controls the physical I/O operations, system resource allocation on a dynamic basis, task switching to achieve multitasking, hardware interrupt servicing, communications with the system operator, and interface to user programs with the system hardware. To accomplish this, the supervisor is composed of the following program elements:

- Interrupt control
- Priority control
- Transient control
- Physical I/O controls
- Resource allocation
- Task control
- Interrupt timer and day clock services

- Program management
- System console management
- File services
- Program error handling
- Cooperative/symbiont operations
- Physical input/output control system (PIOCS)
- Debugging aids

1.6.2. Job Control

Job control is a nonresident program of the operating system responsible for controlling the orderly initiation and termination of jobs within a multiprogrammed environment. The job control services are performed prior to execution of the initial job step of a job, during the transition between job steps, and at the conclusion of a job. Some of the services of job control are:

- Volume label and file label storage
- Job control stream file maintenance
- Job scheduling by priority
- Main storage allocation and reallocation
- Peripheral device assignment
- Program restart

The functions of job control are implemented by the programmer through the job control language or by the operator through the system console commands. These sequenced control statements form the control stream that defines a job's facility requirements and directs the execution of the job. The job control statements, through the job control stream, function as an interface between the programmer and OS/3.

1.6.3. Data Management

Data management provides the interface between the hardware-oriented I/O facility and the user program. The data management facilities consist of logical input/output control stream (IOCS) modules, transient routines, declarative macro instructions, and imperative macro instructions.

1.6.4. Integrated Communication Access Method

The communication software necessary to support remote terminals or processors is controlled by two logical levels of software. These levels are:

- the communications physical input/output control system (CPIOCS) and the communication symbionts; and
- the message control program (MCP).

There are four user levels (interfaces) which communicate with the logical control levels via declarative and imperative macro instructions. These are:

- CPI – communications physical interface
- DDI – direct data interface
- STDMCP – standard GET/PUT interface
- TCI – transaction control interface

1.6.5. Emulators and Transition Aids

There are two emulation programs and one transition program that adapt the instruction repertoire and peripheral characteristics of existing systems to OS/3. The IBM 360/20 emulator allows existing 360/20 programs to be executed on the 90/30, 90/30 B, or 90/40 system. The 9200/9300 emulator allows existing 9200/9300 programs to be executed on the 90/30, 90/30 B, or 90/40 system. The IBM System/3 transition program allows existing System/3 programs to be executed on the 90/30, 90/30 B, or 90/40 system.

1.6.6. System Service Programs

The OS/3 system service programs make available to the system the means for sorting data into a specified order, merging data to facilitate processing, maintaining files on magnetic disk storage, linking output modules of language processors into executable programs, copying input cards, magnetic tapes, disk or diskette files to any other card, magnetic tape, disk or diskette, or printer device.

The major system service programs include:

- Data utilities
- Linkage editor
- System librarian
- Sort/merge

1.6.7. Language Processors

Four language processors are available with OS/3: assembler, COBOL, FORTRAN, and report program generator (RPG). All language processor input can be on punched cards, magnetic tape, or disk files; all output can be recorded on magnetic tape or disk files. All processor output is in a common system output format.

1.6.8. Diagnostic Programs

The diagnostic programs provided with OS/3 are hardware maintenance routines that can be executed concurrently with user programs. These programs are intended to be run as confidence tests by the system operator, and as diagnostic and maintenance tests by the customer engineer.

1.6.9. Application Programs

Application programs are those specialized programs that are available to a user but are not provided as part of the standard software package. These programs are directed towards handling problems distinctive to a particular user and include program evaluation and review techniques/critical path method analysis (PERT/CPM) and linear programming (LP).

1.6.10. Information Management System

The information management system (IMS 90) used with OS/3 is common to the 9000 series and facilitates access to information stored in data files. IMS 90 provides a terminal-oriented data retrieval and update capability for managerial and clerical personnel and, thereby, relieves them of needing to learn complex methods employed by programming personnel. IMS 90 is in the form of Sperry Univac-supplied application programs called UNIQUE, which require programming effort. Also, coding required for line and device handlers is provided.

1.6.11. Data Base Management System

The data base management system DMS 90 is a collection of system programs that support the development of integrated data bases. These programs provide for the description, initialization, creation, accessing, maintenance, backup, and recovery of data base. The languages used in the description and manipulation of DMS 90 data bases are derived from the CODASYL data base specifications. A data base may be accessed by batch application programs and communications application programs.

2. System Turn-on and Turn-off Procedures

2.1. SYSTEM TURN ON

To turn on your SPERRY UNIVAC 90/30, 90/30 B, or 90/40 System from a full power-off condition, proceed as follows:

1. Set the system circuit breakers (wall mounted) to the ON position.

2. Set the POWER ON/POWER OFF switch on the processor operator/maintenance panel to the POWER ON position. The POWER CYCLING indicator lights during the power-up sequence. All other indicators remain off.

   When the power-up sequence is complete, the POWER ON indicator on the operator/maintenance panel lights and all other indicators are extinguished. This condition indicates that operating power is applied to all the central hardware and online peripheral devices, excluding 0604 card punch subsystems, in the system.

   If the POWER CYCLING indicator, as well as the POWER ON indicator, on the operator/maintenance panel lights at the end of the power-up sequence, and all the other indicators remain extinguished, a stall condition in the power-up sequencing of the system has occurred. To remedy this condition, set the POWER ON/POWER OFF switch on the operator/maintenance panel to the POWER OFF position. Check the peripheral devices and central hardware circuit breakers and switches to ensure that they are set to their respective POWER ON positions. Then, set the POWER ON/POWER OFF switch to the POWER ON position.

   If, after this power-up sequence is completed, the POWER CYCLING indicator, as well as the POWER ON indicator lights, and all other system indicators remain extinguished, refer the situation to your Sperry Univac customer engineer.

3. Apply power to all the 0604 card punch subsystems in the system as described in 2.3.

2.2. SYSTEM TURN OFF

To turn off your 90/30, 90/30 B, or 90/40 system, proceed as follows:

1. Set the POWER ON/POWER OFF switch, on the processor operator/maintenance panel, to the POWER OFF position. All the indicators on the operator/maintenance panel light momentarily and then extinguish.

2. Turn off all the 0604 card punch subsystems in the system in accordance with the turn-off procedure in 2.3.

3. If the system is to remain unused for a period of time, set the system circuit breakers (wall mounted) to the OFF position.

2.3. SPECIAL TURN-ON/TURN-OFF PROCEDURES

The 0604 card punch must be turned on and turn off from its operator control panel since no remote power control is provided within this device. To power up the card punch, you must set the 1CB1 circuit breaker on the power control panel to its on (up) position, press the OFF LINE switch/indicator (OFFLINE indicator lights), and press the POWER MAN ON INTL/READY switch/indicator. Press the OFF LINE switch/indicator again (OFFLINE indicator extinguishes) to place unit online with the multiplexer channel. Use the *UP* function of the SET IO command (4.4.8) to place the card punch on the system's resources list for job allocation.

CAUTION

Place the card punch offline to the multiplexer channel prior to applying power with the POWER MAN ON INTL/READY switch/indicator. After circuit breaker 1CB1 is turned on, press the OFFLINE switch/indicator if that switch/indicator is extinguished. Power transients may interfere with proper operation of the processor if this procedure is not followed.

When power is applied to the unit, the following indicators on the operator control panel light:

- POWER MAN ON INTL/READY switch/indicator
- POWER MAN OFF AC/DC switch/indicator
- CLEAR HOPPER switch/indicator (remains lit until the operator loads cards into the hopper)

All other indicators should be extinguished.

Power is removed from the punch unit by first pressing the OFF LINE switch/indicator (OFF LINE portion of switch/indicator lights) and then pressing POWER MAN OFF switch/indicator on the operator control panel. The blowers are turned off by setting the 1CB1 circuit breaker to off (down). Use the *DOWN* function of the SET IO command (4.4.8) to remove the card punch from the system's resources list and prevent it from being allocated to a job while offline.

3. System Initialization Procedures

3.1. GENERAL

Initialization of the SPERRY UNIVAC 90/30, 90/30 B, or 90/40 System is a 2-step operation.

1. Loading and initializing control storage (3.2)

2. Loading and initializing the resident portion of the supervisor (3.3).

Whenever power is removed from the system, you must perform both these operations to initialize the system after power is reapplied. Whenever a nonrecoverable error occurs, only the supervisor need be reloaded and initialized. The only time control storage needs to be reloaded and initialized, after a nonrecoverable error occurs, is when you are unable to load and initialize the supervisor.

3.2. CONTROL STORAGE LOAD PROCEDURE

Whenever the 90/30, 90/30 B, or 90/40 system is turned on from a full power-off condition or the initial program load (IPL) operation cannot be completed successfully, control storage must be loaded and initialized. To load and initialize control storage, proceed as follows:

1. Place the system resident (SYSRES) disk pack containing the control storage code on a suitable disk drive unit and set the disk drive to the run state.

2. Perform the following operations at the operator/maintenance panel.

    a. Set the INHIBIT TIMER switch to its off (down) position.

    b. Set the INHIBIT PROC CHECK switch to its off (down) position.

    c. Set the HALT ON ERROR switch to its off (down) position.

    NOTE:

    The INHIBIT TIMER, INHIBIT PROC CHECK, and HALT ON ERROR switches are never set to the on position during normal system operation.

    d. Set the MODE SELECT switch to NORMAL.

    e. Set the INITIAL LOAD CONTROL switch to CONT STOR LOAD.

f. Set the DATA ENTRY CHANNEL NO. switch to the channel number on which the selected disk drive is connected. The numbers assigned to these channels are:

Channel	Assigned Number
Integrated disk adapter	3
Selector channel 1	4
Selector channel 2	6

g. Set the DATA ENTRY SUBCHANNEL NO. switch to the subchannel number assigned to the selected disk drive unit. The possible subchannel assignments are:

Subchannel	Assigned Number
Integrated disk adapter	0
Selector channel 1	0–7
Selector channel 2	0–7

h. Set the DATA ENTRY DEVICE NO. switch to the actual physical unit number assigned to the selected disk drive unit (0 through F).

i. Press the top portion of the SYSTEM RESET switch twice. The TEST MODE indicator lights.

j. Press the top portion of the RUN switch. The INITIAL LOAD indicator lights and remains lit until control storage has been successfully loaded (HPR STOP indicator lights).

If the INITIAL LOAD indicator fails to extinguish and the PROC CHECK or CONTROL STORAGE indicators light, control storage was not successfully loaded and steps 2i and 2j must be repeated.

NOTE:

Do not disturb any switch settings until control storage loading is complete.

If the proper indications cannot be obtained, try loading control storage from another disk drive unit. If the abnormal indications persist, refer the problem to your Sperry Univac customer engineer.

3.3. INITIAL PROGRAM LOAD PROCEDURE

Before the system can be used to run productive jobs, the resident portion of OS/3 (the supervisor) must be loaded into main storage and intialized. This operation can only be performed after control storage is loaded and initialized, and must be performed whenever control storage is loaded or a nonrecoverable error occurs.

To load and initialize the supervisor (the IPL procedure), proceed as follows:

CAUTION

Make certain the system is in the idle condition (no jobs are active) before you perform the IPL procedure. Otherwise, the SYSRES volume table of contents (VTOC) may be left in a nonrecoverable state requiring that a new SYSRES volume be generated.

1. Place the system resident (SYSRES) disk pack containing the initial program load (IPL) routine on a suitable disk drive unit and set the disk drive to the run state.

   NOTES:

   - *The control storage code and initial program load routine are stored on the same disk pack.*
   - *If an 8418 disk is being used for the SYSRES, the density setting on the drive must match the density of the recorded disk pack. The PROC CHECK indicator lights if the densities do not match.*

2. Set the controls and switches on the operator/maintenance panel to the following positions:

   a. Set the INHIBIT TIMER switch to its (down) position.

   b. Set the INHIBIT PROC CHECK switch to its off (down) position.

   c. Set the HALT ON ERROR switch to its off (down) position.

      NOTE:

      The INHIBIT TIMER, INHIBIT PROC CHECK, and HALT ON ERROR switches are never set to the on position during normal system operation.

   d. Set the MODE SELECT switch to NORMAL position.

   e. Set the INITIAL LOAD CONTROL switch to PROGRAM LOAD position.

   f. Set the DATA ENTRY CHANNEL NO. switch to the channel number on which the selected disk drive is connected. The numbers assigned to these channels are:

Channel	Assigned Number
Integrated disk adapter	3
Selector channel 1	4
Selector channel 2	6

   g. Set the DATA ENTRY SUBCHANNEL NO. switch to the subchannel number assigned to the selected disk drive unit. The possible subchannel assignments are:

Subchannel	Assigned Number
Integrated disk adapter	0
Selector channel 1	0–7
Selector channel 2	0–7

   h. Set the DATA ENTRY DEVICE NO. switch to the actual physical unit number assigned to the selected disk drive unit (0 through F).

   i. Press the top portion of the SYSTEM RESET switch. The TEST MODE indicator lights.

j. Press the top portion of the RUN switch. The INITIAL LOAD and RUN indicators light. After the initial program load routine is loaded into the system, the INITIAL LOAD indicator goes out; only the POWER ON, TEST MODE, and RUN indicators remain lit.

If either the PROC CHECK or HPR STOP indicators light, the IPL operation was unsuccessful and steps 2i and 2j must be repeated until the proper indications are obtained. If the PROC CHECK indicator continues to light, refer the problem to the Sperry Univac customer engineer. If the HPR STOP indicator continues to light, perform the following procedure to determine the cause of the error:

(1) Set the DISPLAY SELECT 1 switch to UPPER position.

(2) Set the LEGEND SELECT 1 switch to position 7.

(3) Read the first four sets of the DISPLAY 1 indicators (I0 through I15). HPR codes displayed by these indicators are listed and described in the OS/3 system messages programmer reference, UP-8076 (current version).

If possible, correct the cause of the HPR STOP, and then repeat steps 2i and 2j. If the HPR code identified a condition that cannot be corrected by the operator or if the HPR STOP persists, contact your Sperry Univac customer engineer.

After this phase of the IPL operation is successfully completed, proceed with step 2k.

NOTE:

The system automatically sets all devices or subsystems not online during IPL time to down. The devices or subsystems are not available for system use until they are identified as available by the operator via the SET IO command (4.4.8) or until a disk pack is mounted on them and they are initialized.

k. Set the INITIAL LOAD CONTROL switch to the OFF position. The TEST MODE indicator extinguishes and the following message appears on the system console screen:

IPL TO LOAD STANDARD SUPERVISOR UNLESS NEW NAME KEYED IN ⌐_____ {L C D}

NOTE:

At this point, the cursor (⌐) is at the point of entry for the type-in of the desired supervisor name (six characters). The final character after the comma is for special types of loading as follows:

L = *Special load for stand-alone programs*

C = *Supervisor control storage card read*

D = *Supervisor debug option (For details, see supervisor user guide, UP-8075, current version).*

l. If the standard supervisor is to be loaded into the system, press the TRANSMIT key on the system console. No keyin is required. If a different supervisor is to be loaded in, key in the name of the new supervisor and then press the TRANSMIT key. The following statements are now displayed on the system console. Respond to each statement as directed. After all statements are answered, again press the TRANSMIT key.

NOTE:

The cursor (¬) is initially positioned at the point of entry for the date (second statement). Upon completion of a keyin, the cursor is automatically positioned on the following line at the point where keyin is to begin. Lines requiring answers may be bypassed by pressing the RETURN key on the system console, thus selecting the displayed default value.

The MESSAGE WAITING key need not be pressed before initiating any keyin during this procedure.

OS/3 VERSION nn

Indicates the release version (nn) of the OS/3 system; no reply is required.

DATE? ({ YY/MM/DD / MM/DD/YY / DD/MM/YY }) _ _/_ _/_ _

Requires the date to be entered in the format shown. The format is a 2-digit configuration where:

DD=day
MM=month
YY=year

The date format displayed in this message is selected at SYSGEN time for operator convenience. The format of the date, as used by the system for all processing operations and output messages, is YY/MM/DD, regardless of the date format shown here.

TIME? (HH/MM/SS)_ _:_ _:_ _

Requires the time of day to be entered in hours, minutes, and seconds using 2-digit format.

RUN LIBS DEVICE ADDR? (DEFAULT=SYSGEN option)_ _ _

Questions whether the system job run library file YRUN is to be located on the disk volume specified during SYSGEN and identified as the DEFAULT value, or on another disk volume. If the run library is to be located on the DEFAULT volume, no keyin is required; press the RETURN key on the system console to position the cursor on the next line. If you wish to locate YRUN on a different volume, key in the device address of the disk unit containing the desired disk volume.

Remember, the volume identified as the YRUN volume must be online for the system to be operational. Should the specified volume not be online, a system message to mount the volume will appear on the system console.

RECOVER FILES?

Requests operator action on the following three statements:

JOB QUEUE (N, Y, H DEFAULT=N)

If the jobs previously filed in the job queue are to be retained for processing, key in Y; if they should be put into hold status, key in H. If they should be deleted from the queue, press the RETURN key to advance to the next statement.

ERROR LOG (N, Y DEFAULT=Y)

This message appears only if the error log option is configured in your system. Key in N to clear all accumulated errors in the error log and start a new error log file. Otherwise, if you wish to retain the present error log file and continue to list errors in the error log, press the RETURN key to advance to the next line.

SPOOL FILES (N, A, C, L, H DEFAULT=SYSGEN option)

This message appears only if the spooling option is configured in your system. It requests that you specify the level of recovery you desire for the spool file in your system. You specify the level of recovery by keying in one of the following responses:

N

Previously spooled input and output was processed before the system was turned off and the spool file is empty; therefore, no recovery is required.

A

All spooled subfiles are to be recovered when the spool file is reinitialized because previously spooled or output files in the spool file are to be processed. With this response, all spooled subfiles, whether complete or incomplete, are to be saved. This response (or H) must be specified to recover the system console log file (if configured into the system). Console messages that were not copied from main storage buffer to spool file are not recovered; copying is done only when buffer has been filled.

C

Only completed subfiles are to be recovered when the spool file is initialized.

L

Only the user log directory is to be recovered when the spool file is initialized.

H

May be selected only if the operator has taken a system dump of the previously loaded system at the operator/maintenance panel by simply pressing the SYSTEM RESET switch, then the RUN switch. (Refer to the system service programs manual, UP-8062 (current version) for further system dump information.) When the spool file is reinitialized, all spool subfiles (same as with A option), as well as system console messages that have been accumulated in main storage buffer, but not copied onto the spool file, are recovered; copying is done only when the buffer has been filled. This method of recovery (sometimes referred to as the hot start) should be used only if a system crash occurs. If there is a planned shutdown of the system and spool files are to be recovered at a later date, the operator should breakpoint the console log, then select the A, C, or L spool recovery option when the system is reinitialized.

If you wish to specify the same level of recovery that was specified during SYSGEN, simply press the RETURN key to advance to the next line.

SPOOLING DVC ADDR? (DEFAULT= { blank / vsn / SYSRES }) - - -

This message appears only if the spooling option is configured in your system. Further, if multivolume spooling was configured, it will appear once for each volume that the spool file can be on, as specified at SYSGEN time. Each of these messages requests that you identify the volume serial numbers of the disk volumes that are to contain the spool file. If the volume specified during SYSGEN, as identified in the message as the DEFAULT value, is to be used, press the space bar three times. If you wish to locate the spool volume being referenced on a different volume than that displayed in the message, key in the device address of the disk unit containing the desired disk volume. Repeat this procedure for each message displayed.

If no DEFAULT volume is identified (DEFAULT=blank), you have the option of identifying another volume for use by the spool file or limiting the spool file to those volumes already identified. To identify another volume, key in the device address of the disk unit containing the desired disk volume. To indicate that no more volumes are to be used, press the TRANSMIT key. This action will cause the message presented in step m to be displayed. Remember, all volumes identified as spool volumes at IPL time must be online for the system to be operational.

CAUTION

Be sure to respond to all the above statements requiring keyin before pressing the TRANSMIT key. Once the TRANSMIT key is pressed, the questions and answers are lost and the entire procedure must be restarted if the procedure was not performed correctly.

m. After responding to the requested information as required, press the TRANSMIT key on the system console. When the selected supervisor has been loaded and initialized, the following header message appears on the system console screen.

(1)_____(2)_____(3)_____(4)_____(5)_____(6)_____(7)_____

90/nn OS/3 version-no supnam COS-n yy/mm/dd hh:mm:ss

Message Description:

(1) through (7)
These seven numbers represent the numbers assigned to the seven jobs that can be concurrently run in the system. When a job is initiated, the number disappears and the job name takes its place. The position of the job name signifies its job number.

Example:

Three jobs are running in the system named A, B, and C.

Screen format:

A B C (4) (5) (6) (7)

NOTE:

The first IPL of a supervisor that is configured for spooling with the spool file being located on a selector channel disk subsystem (8411, 8414, 8424, 8425, 8430, or 8433) requires that the spool file be formatted. This formatting operation takes an appreciable amount of time (approximately 1 minute on an 8430 for the default specification of 50 cylinders) and is no cause for concern.

nn
Specifies your machine type (30 or 40).

version-no
Specifies the software version that has been loaded into the system.

supnam
Specifies the name of the supervisor that has been loaded into the system.

n

Specifies the amount of control storage that has been loaded into the system (1, 2, or 3K).

yy/mm/dd

Signifies the year, month, and day being used by the system.

hh:mm:ss

Signifies the time in hours, minutes, and seconds being used by the system.

n. If Y or H was selected for the JOB QUEUE message (refer to step l), the following message appears on the system console screen:

JOB QUEUE RECOVERED – n JOBS QUEUED

Message Description:

n JOBS QUEUED

Specifies the number of jobs in the scheduling queues.

o. If N was selected for the ERROR LOG message (refer to step l), the following message appears on the system console screen:

ERROR LOG NOT RECOVERED

When one job terminates, the next job to be run takes the place and job number of the terminated job on the system console screen heading.

If either the PROC CHECK or HPR STOP indicators (on the operator/maintenance panel) light during these phases (m, n, o) of the IPL operation, the preceding messages won't be displayed because the supervisor wasn't loaded or initialized properly, and you must repeat the IPL operation beginning with step 2e of this procedure.

If, after repeating this procedure, the PROC CHECK indicator remains lit, reload control storage and then repeat the IPL operation. If the HPR STOP indicator continues to light, perform the following procedure, at the operator/maintenance panel, to determine the cause of the error:

(1) Set the DISPLAY SELECT 1 switch to UPPER.

(2) Set the LEGEND SELECT 1 switch to position 7.

(3) Read the first four sets of the DISPLAY 1 indicators (I0 through I15). The possible HPR codes that can be displayed by these indicators during this phase of the IPL operation are listed and described in the OS/3 system messages manual, UP-8076 (current version). If one of the special supervisor initialization HPR codes is displayed, set the LEGEND SELECT 1 switch to position 6 and read the DISPLAY 1 indicators representing registers Z00 through Z15 to further identify the cause of the HPR stop. The first two hexadecimal digits displayed (Z00 through Z07) identify the HPR stop code (01 through 0F). The last two hexadecimal digits further identify the cause of the HPR stop. The meaning of the last two digits is also described in the OS/3 system messages manual, UP-8076 (current version).

If possible, correct the cause of the HPR stop and then retry the IPL operation. If the IPL operation still cannot be completed successfully or if the HPR code identified a condition that cannot be corrected, reload control storage and then retry the IPL operation.

If either the PROC CHECK or HPR STOP error condition persists, contact your Sperry Univac customer engineer.

After the header is displayed on the system console, the system is ready to process user jobs. Note, however, that any integrated disk units (8415's, 8416's, and 8418's) that were offline when the supervisor was initialized are not available for system use until they are identified as available by the operator via the SET IO command (4.4.8) or until a disk pack is mounted on them and they are initialized. Note also that the supervisor cannot distinguish between a low density 8418 disk unit and an 8416 disk unit. Thus, if a low density 8418 is placed online after system initialization, the following SET IO command must be keyed in to identify this fact to the supervisor:

SET IO,device-address,TYP,2002

Jobs requiring more than the available devices will be terminated with an R277 message. The operator may review device ready status with the MIX command (4.4.3) and change the status with the SET IO command.

4. Job Processing Procedures

4.1. GENERAL

After the header is displayed on the system console screen, the system is ready to process user jobs. To begin processing, place the input media (cards, diskette, or disk) containing the job control stream (JCS) to be processed on the input device and set the device to the run state. Then proceed to the system console to initiate running the job, using the system commands described in this section.

To perform confidence tests on the system and peripheral subsystems, refer to the OS/3 online diagnostics manual, UP-8512 (current version).

NOTE:

Refer to Appendix A for the conventions used to illustrate the commands and messages appearing in this handbook.

4.2. KEYIN PROCEDURE

Before keying in any message or command on the system console for transmittal to the system, the MESSAGE WAITING key on the system console must be pressed. Pressing this key notifies the system you want to send a message or command to it, automatically opens a line on the system console screen for display of the input command or message, gives the start of entry (▷) signal, and sets the cursor (¬) to the position where typing is to start.

After keying in the message or command, press the TRANSMIT key on the system console keyboard. This initiates transmittal of the message or command to the system. All commands are acted upon immediately or placed on a queue for future processing; they are never ignored or lost. Commands are placed on queue when insufficient main storage exists or a required device is not available. Queued commands are activated as soon as all requirements for execution are met. Therefore, commands keyed in twice are eventually executed twice. If a message or command is unacceptable, the system responds with a negative acknowledgment (NAK) error message indicating why. This error message appears in the last 12 character positions of the line of the unacceptable message or command and, if necessary, overwrites any message or command text that may be present in the last 12 character positions. A pair of blinking marker symbols (◹ and ◿) bracket each error message. The message or command must then be retyped in accordance with the information furnished by the error message, so that the job involved with the unacceptable message or command may be executed. The error messages that may be displayed are described in the system messages manual, UP-8076 (current version).

If console logging has been configured for the system, the console log messages are recorded in a spool file for later printing. Any console log messages not recorded will be identified by a single blinking character in the rightmost position of the console line containing the message. This may occur if, for example, the device containing the spool file is in error recovery and the operator does not want to halt processing.

4.2.1. System Command Characteristics

When a command is being typed in, there must be at least one space between the command and the first parameter, and commas between all parameters. The general format for these commands is:

Format:

```
▷ command [{(did)           }]   [command-parameters]
          [{([did],label)   }]
          [{(RDR,label)     }]
```

where:

▷
: Is the start of entry symbol (SOE) that must precede all lines. This symbol is automatically generated by OS/3 when the MESSAGE WAITING key is pressed.

command
: Is two to eight alphabetic characters that identify the system command to be processed. At least two characters must be supplied.

(did)
: Is a 3-character device address that identifies a device that is to be used when carrying out the command. A device address should be included when a particular peripheral device is to be specified or when no default option is provided.

  If a did is not entered, the first appropriate device will be used.

([did],label)
: When a diskette is used to serve the function of a card reader or card punch, a file identifier (label) is required to identify the specific file to be accessed on the diskette. The device address also must be included, unless the diskette is configured as the SYSRDR. The label, which may be a maximum of eight alphanumeric characters, is separated from the device address by a comma.

(RDR,label)
: Specifies that the device address to be used is the input reader spool file with the specified label. The label may be a maximum of eight alphanumeric characters and is separated from the RDR entry by a comma.

  Examples:

  RUN(010,DATANAM1) JOBNAM1
  RUN(,DATANAM1) JOBNAM1 (Valid if SYSRDR is configured as a diskette)

command-parameters
: Are optional positional parameters used to influence the effect of the command being issued. Each specified parameter must appear in its own position. Commas are used to separate positions. For example, the command format

```
RUN [{(did)           }] [{jobname              }] [,{PRE }]
    [{([did],label)   }] [{jobname (new-name)   }] [{HIGH}]
    [{(RDR,label)     }] [{(new-name)           }] [{NOR }]
```

  indicates that two positional parameters are associated with the RUN command; the comma separates the first from the second. If only the second parameter is to be specified, the command must be keyed in as follows:

```
RUN ,HIGH
```

As shown, the comma must be included to indicate the omission of the first parameter; otherwise, HIGH would be considered the name of the job to be run.

Also, if a parameter may take more than one form (as is the case with the first parameter in the RUN command), the punctuation marks (in this case parentheses) must be keyed in whenever shown in the format. For example, if a new-name is specified, the command could be keyed in as

```
RUN MYJOB(JOBA),PRE
```

or

```
RUN (TEMPNAME)
```

In either case, the parentheses must be included.

4.2.2. Message Characteristics

4.2.2.1. Output Messages

Output messages are displayed on the system console to provide the operator with information, to direct the operator to take some action, or to ask a question that requirs an operator response. The messages that may be output to the system console by the components of OS/3 are described in the system messages manual, UP-8076 (current version), together with their associated operator responses, when appropriate. The format of an output message is:

where:

▷

Is the start of entry symbol (SOE); must precede all lines.

j

Is a 1-digit job number that is assigned to each active job in the system. The numbers 1 through 7 are assigned to user jobs as they become scheduled for execution. The number assignments for user jobs are shown in the header area on the system console screen (Figure 4–1); as each job is assigned a job number, its job name replaces the number previously displayed. This assigned number is used in output messages to identify the job that transmitted the message and, in input messages, to identify the job that is to receive the message. The supervisor is always assigned the job number zero.

i

Is a 1-digit hexadecimal message number. Message numbers are consecutively assigned to output messages that are generated by user jobs beginning with the number 1 and ending with the letter F. Likewise, message numbers are consecutively assigned to output messages generated by the supervisor. These numbers are used together with the job numbers to explicitly identify each message in the system. When an output message requires a reply, the reply message must be prefixed with the job and message number of the message requesting the reply. Unsolicited input messages are identified by the message number zero. Thus, an unsolicited message to the supervisor has the prefix 00; and an unsolicited message to job number 1 has the prefix 10; job 2 has the prefix 20, etc.

?

Identifies an output message that must be responded to before the job that issued the message can continue. Output messages requiring replies are not rolled off the system console screen until they are answered.

△

Identifies an output message that requires no reply or operator action; information only. Input messages, solicited and unsolicited, may optionally include a space between the message number and message text.

*

Identifies an output message that requires that some action be taken by the operator. The job that generated the message has placed itself in a "yield" state. A GO command is required from the operator to reactivate the job.

`message-text`

Comprises the actual message content and is a maximum of 60 characters.

4.2.2.2. Solicited Input Messages

Solicited messages are those messages input by the operator in direct response to an output message that requires a reply (question mark immediately follows message number). The format for all solicited input messages is:

```
▷ ji△message-text
```

where:

`ji`

Identifies the job and message number of the message being replied to; Figure 4–1, line 3 is an example of a solicited message. The message soliciting the response is shown in line 2.

`message-text`

Is the actual reply message.

4.2.2.3. Unsolicited Input Messages

Unsolicited messages are those messages input by the operator that are not in direct response to an output message that requires a reply. The format for all unsolicited input messages is:

```
▷ jØ△[symbiont-name][(did)]△message-text
```

where:

`j`

Is the job number of the job you want to receive the unsolicited message.

`Ø`

Is the message number that must be used to identify the message as an unsolicited message.

`symbiont-name`

Is the 2-character alphanumeric name of the supervisor symbiont that is to receive the unsolicited message (the job and message numbers are 00). If a symbiont is not the recipient of the message, no symbiont name is required.

did

Is the address of the device being used or controlled by a specific copy of a symbiont in main storage. The symbiont having this device allocation receives the unsolicited message. If the specified device is not assigned to the symbiont, the unsolicited message is not acknowledged. The device-address must be enclosed in parentheses.

NOTE:

When you key in an unsolicited message to a symbiont, the system task control blocks (TCB) are searched to locate an active symbiont identified by the name specified in the message. When no address is specified, the unsolicited message is transferred to the buffer of the first symbiont encountered in the switch list that bears this name. If there is more than one copy of the same symbiont active, only the first will get the message. Figure 4—1, line 11, is an example of an unsolicited type-in to a symbiont.

message-text

Is the actual text of the message.

```
     PROGRAM1 (2)______(3)______PROGRAM4 PROGRAM5 (6)______PROGRAM7
1    ▷ 71 THIS IS A COMMENT FROM PROGRAM7
2    ▷ 12? ANSWER A QUESTION FROM PROGRAM1?
3    ▷ 12 THIS IS THE ANSWER TO PROGRAM1
4    ▷ 50 ACTIVATE USER ISLAND CODE FOR PROGRAM5
5    ▷ 43* MOUNT DEV=440 VSN=DSP614 LU=050 DEV=441 VSN=DSP633 LU=051
6    ▷ 44* MOUNT DEV=442 VSN=DSP554 LU=052 GO?
7    ▷ GO PROGRAM4
8    ▷ DISPLAY 140,7
9    ▷ 0A? THIS IS A QUESTION FROM THE 'DISPLAY' SYMBIONT
10   ▷ 0A THIS IS THE ANSWER TO THE 'DISPLAY' SYMBIONT'S QUESTION
11   ▷ 00 IO ACTIVATE IO SYMBIONT ISLAND CODE
12   ▷ CANCEL IO,S
13   ▷ DUMP PROGRAM5
14   ▷ 0B THIS IS A COMMENT FROM THE DUMP ROUTINE
15   ▷ END DUMP,PROGRAM5
```

Figure 4—1. Typical System Console Messages

The following paragraphs describe the operator procedures for entering commands, unsolicited messages, and solicited messages, according to the function required. Specific command and message formats are included, specifying the appropriate parameters and their order for that particular function.

4.3. JOB PROCESSING COMMANDS

Job processing commands enable the operator to:

- read job control streams into the system and assign scheduling priorities to them (job initialization);
- control jobs awaiting execution within the scheduling priority queues (schedule jobs);
- control jobs being executed (execute jobs); and
- stop jobs under execution (terminate jobs).

In the job processing commands that follow, scheduling priorities are defined as preemptive, high, or normal to specify in what order jobs are to begin execution. Jobs to be run are placed in one of the three scheduling priority queues:

- PRE (preemptive) Queue

  Contains jobs that are to be executed first, i.e., before any jobs assigned HIGH or NOR scheduling priority. If rollin/rollout is configured, a PRE job that is initiated for execution when sufficient main storage is not available may cause HIGH or NOR jobs being processed to be rolled out to make main storage space available for the PRE job. Rolled out jobs are rolled in and continue processing when main storage is again available.

- HIGH Queue

  Contains jobs that are to be executed before any jobs assigned a NOR scheduling priority. HIGH scheduling priority jobs are not executed unless the PRE queue is either empty or placed on hold.

- NOR (normal) Queue

  Contains jobs that are to be executed only when there are no jobs left in the PRE or HIGH queues or when the queues have been placed on hold. NOR scheduling priority is the default for a job control stream and for some of the job scheduling commands (4.3.2).

4.3.1. Job Initialization

A job control stream that is to be filed for future use or one that is to be processed immediately is read into the system using one of the job initialization commands (FILE, RUN/RV, and OCL). When a job is initialized for immediate processing, it is placed in a scheduling priority queue to await execution. This scheduling priority queue is normally specified by the programmer submitting the job, via the job control stream itself. However, the operator may override this specification by specifying another scheduling priority in the job initialization command.

When a job control stream is filed for future use, it is placed in the YJCL file.

4.3.1.1. Filing Job Control Streams (FILE)

Function:

The FILE command files jobs and JPROCs, read from an input device, in the permanent JCS library file (YJCS).

Format:

```
FILE [{(did)          }]
     [{([did],label)  }]
     [{(RDR,label)    }]
```

This command requires no positional parameters. If no device and label are identified, the first available card reader, as defined when the system was generated, is expected to contain the job control streams and/or JPROCs to be filed. If the job control stream is on a diskette, the label is required; if it is in the input spool file, RDR and label are required. (See 4.2.1.)

The jobs to be filed that originate from the card reader must terminate with a // FIN job control statement. When a diskette or spool file contains the jobs to be filed, the // FIN job control statement is not necessary.

4.3.1.2. Running Job Control Streams (RUN/RV)

Function:

The RUN/RV commands initiate the reading of a job control stream from either an input device or the YJCS file. The RV command is used only to initiate the reading of a job control stream that does not reside in a card reader, diskette, or spool file and that does not contain a // CR statement. The RUN command is used to initiate the reading of a job control stream that requires the use of an input device (i.e., card reader, diskette, or spool file). When the RUN command is issued, it is accepted only if an input device is available, whether or not one is needed by the job control stream being read. The RV command allows a job control stream to be initiated that does not require the use of an input device even when there are no input devices available in the system. Therefore, you must include a job name when you enter an RV command.

The operator should remember that when a system card reader is placed online, the RUN command to read cards in the hopper is initiated when the RUN switch on the card reader is pressed, or when the RUN command is keyed in at the system console. The RUN command should be initiated from either location, but not from both. Initiating a duplicate RUN command for the same job causes an error message to be displayed on the system console, because the second RUN command will not be accepted by the supervisor.

Format:

```
{RUN[{(did)          }]}                [{jobname            }]  [{PRE }]  [,key-1=val-1,...,
{    [{([did],label)  }]}  [{jobname(new-name)}]  ,  [{HIGH}]
{RV  [{(RDR,label)    }]}  [{(new-name)       }]     [{NOR }]
      key-n=val-n]
```

Command Code:

RUN
Used to initiate the running of a job control stream that requires the use of an input device. If a prefiled job control stream is to be run, positional parameter 1 must be specified. If no device and label are identified, the first available card reader, as defined when the system was generated, is expected to contain the job control stream to be run.

If the job control stream is on a diskette, the label is required; if it is in the input spool file, RDR and label are required. (See 4.2.1.)

RV
Used to initiate the running of a prefiled job control stream that does not require the use of an input device. You must specify a job name (positional parameter 1) when RV is used. The job stream to be run must not contain a // CR (read card reader) statement. For diskette and spool file input, the last // FIN job control statement is not necessary. // FIN statements that separate groups of card images read with // CR statements are still necessary.

Positional Parameter 1:

jobname
Identifies the name of the job to be read from YJCS and stored in a scheduling priority queue to await execution. The job name consists of one to eight alphanumeric characters.

jobname (new-name)
Assigns a new 1- to 8-character alphanumeric name to a job already stored in YJCS. The job identified by the *jobname* parameter is read from YJCS and stored in a scheduling priority queue under the name identified by the *new-name* parameter to await execution.

(new-name)
Assigns a new 1- to 8-character alphanumeric name to a job input from the card reader. The job is read and stored in a scheduling priority queue under the new name to await execution.

If omitted, the job is read and stored in a queue under the jobname identified in the // JOB statement in the job control stream.

Positional Parameter 2:

PRE
Specifies that the job is to be placed in the preemptive scheduling priority queue to await execution.

HIGH
Specifies that the job is to be placed in the high scheduling priority queue to await execution.

NOR
Specifies that the job is to be placed in the normal scheduling priority queue to await execution.

Positional Parameters 3 through n:

key-1=val-1,...,key-n=val-n
Are the keywords and their values, which may be used by the job being run. The keywords and their values must be supplied by the person requesting the job.

NOTE:

The total length of all the parameters specified in this command, from the first character of positional parameter 1 to the last character of the last keyword value specified, is limited to 60 characters.

4.3.1.3. Running IBM System/3 Operation Command Language Jobs (OCL)

The OCL command enables the operator to run an IBM System/3 job control stream in an OS/3 environment. When the OCL command is entered, the entire System/3 control stream is read and interpreted by the OCL processor. Once the OCL processor has verified that no syntax or sequence errors exist, the job is placed in a scheduling priority queue.

The jobs to be run that originate from the card reader must terminate with an OS/3 // FIN job control statement. For diskette and spool file input, the last // FIN job control statement is not necessary. // FIN statements that separate groups of card images read with // CR statements are still necessary. Several // FIN job control statements may be inserted to logically divide each independent task (or job) to utilize the multijobbing power of OS/3.

Format:

```
OCL [{(did)              }] [(new-name)] [,{PRE }]
    [{([did],label)      }]              [{HIGH}]
    [{(RDR,label)        }]              [{NOR }]
```

If no device and label are identified, the first available card reader, as defined when the system was generated, is expected to contain the job control stream to be run. If the job control stream is on a diskette, the label is required; if it is in the input spool file, RDR and label are required. (See 4.2.1.)

Positional Parameter 1:

(new-name)
: Enables the operator to rename the job to be run; otherwise, the job name will be taken from the // JOB statement or will be OCLnnnn by default, where nnnn is a decimal number from 0001 to 9999. A name of up to eight alphanumeric characters, within parentheses, can be assigned.

Positional Parameter 2:

PRE
: Specifies that the job is to be placed in the preemptive scheduling priority queue to await execution.

HIGH
: Specifies that the job is to be placed in the high scheduling priority queue to await execution.

NOR
: Specifies that the job is to be placed in the normal scheduling priority queue to await execution.

If positional parameter 2 is omitted, the scheduling priority assigned to the job via the job control stream is in effect.

4.3.2. Job Scheduling

A job is placed in a scheduling priority queue to await the availability of system resources (e.g., main storage, disk drive, printer, etc) necessary for that job to be executed. While waiting for these resources, the operator can exercise control over any specific job in a queue, all jobs in a specific queue, and all jobs in all queues, through the use of job scheduling commands. These commands allow you to:

- Defer jobs from being executed
- Permit jobs to be executed

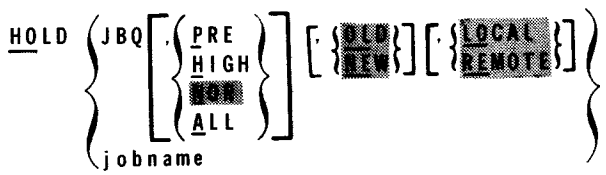
- Delete jobs from a queue
- Display contents of a queue
- Change a job's scheduling priority

4.3.2.1. Deferring Jobs Scheduled for Execution (HOLD)

Function:

The HOLD command permits the operator to defer the scheduling of all jobs in all queues, a specific job within a queue, or all jobs in a specific queue, as specified by command parameters. Scheduling remains deferred until the jobs are reactivated via the BEGIN command.

Format:



Positional Parameter 1:

JBQ
Specifies that the command applies to all jobs in a job scheduling priority queue, as further defined by positional parameters 2, 3, and 4.

jobname
Specifies that a particular job is to be deferred from being scheduled for execution. No further parameters are permitted.

Positional Parameter 2:

PRE
Specifies that the jobs in the preemptive scheduling priority queue are to be deferred.

HIGH
Specifies that the jobs in the high scheduling priority queue are to be deferred.

NOR
Specifies that the jobs in the normal scheduling priority queue are to be deferred.

ALL
Specifies that the jobs in all scheduling priority queues are to be deferred.

Positional Parameter 3:

OLD
Specifies that only jobs already in the scheduling priority queue defined in parameters 1 and 2 are to be deferred. Jobs subsequently entered in this queue are not to be deferrred.

NEW
Specifies that only jobs subsequently placed in the scheduling priority queue defined in parameters 1 and 2 are to be deferred. All existing jobs are still available for execution.

If omitted, both old and new jobs are to be deferred. This parameter may be interchanged with parameter 4.

Once a new job is placed in queue, it becomes an old job for any subsequent commands to defer or permit execution. Therefore, the NEW parameter is used with the HOLD command to defer new jobs entering a queue while the old jobs already residing in the queue remain unchanged and are still able to be scheduled for execution. Likewise, the OLD parameter is used with the HOLD command to defer old jobs already residing in the queue while the new jobs entering the queue are still able to be scheduled for execution. Once a new job enters a queue that is under the influence of a HOLD NEW command, a subsequent command to permit the old jobs in that queue to be scheduled for execution will also release the new job from its deferred status.

Positional Parameter 4:

LOCAL
Specifies that only locally entered jobs, as defined in parameters 1, 2, and 3, are to be deferred.

REMOTE
Specifies that only jobs entered remotely (i.e., from a remote batch terminal), as defined in parameters 1, 2, and 3, are to be deferred.

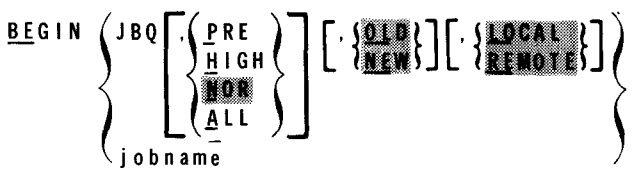
If omitted, both local and remote jobs are deferred. This parameter may be interchanged with parameter 3.

4.3.2.2. Scheduling Deferred Jobs (BEGIN)

Function:

The BEGIN command reinstitutes the scheduling for execution of a currently deferred job, job queue, or all jobs in all queues; as specified by command parameters. Jobs remain deferred by a HOLD command until a BEGIN command is entered to permit their rescheduling for execution.

Format:



Positional Parameter 1:

JBQ
Specifies that the command applies to all jobs in a job scheduling priority queue, as further defined by positional parameters 2, 3, and 4.

jobname
Specifies the particular job permitted to be scheduled for execution. No further parameters are allowed.

Positional Parameter 2:

PRE
Specifies that the jobs in the preemptive queue are permitted to be scheduled for execution.

HIGH
Specifies that the jobs in the high queue are permitted to be scheduled for execution.

NOR

Specifies that the jobs in the normal queue are permitted to be scheduled for execution.

ALL

Specifies that the jobs in all queues are permitted to be scheduled for execution.

Positional Parameter 3:

OLD

Specifies that only jobs already in the scheduling priority queue defined in parameters 1 and 2 are permitted to be scheduled for execution.

NEW

Specifies that newly entered jobs to be placed in the scheduling priority queue defined in parameters 1 and 2 are permitted to be scheduled for execution.

If omitted, both old and new jobs are permitted to be scheduled for execution. (See 4.3.2.1.) This parameter may be interchanged with parameter 4.

Positional Parameter 4:

LOCAL

Specifies that only locally entered jobs, as defined in parameters 1, 2, and 3, are permitted to be scheduled for execution.

REMOTE

Specifies that only jobs entered remotely (i.e., from a remote batch terminal), as defined in parameters 1, 2, and 3, are permitted to be scheduled for execution.

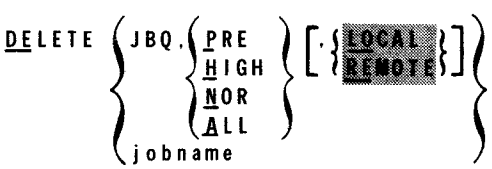
If omitted, both local and remote jobs are permitted to be scheduled for execution. This parameter may be interchanged with parameter 3.

4.3.2.3. Deleting Jobs from Scheduling Priority Queues (DELETE)

Function:

The DELETE command permits the operator to delete a specific job from a scheduling queue, a queue of jobs, or all jobs in all queues, as specified by the command parameters. Only those jobs residing in a scheduling priority queue, and thus waiting to begin execution, can be deleted.

Format:



Positional Parameter 1:

JBQ

Specifies that the command applies to all jobs in a job scheduling priority queue, as further defined by positional parameters 2 and 3.

jobname

Specifies that a particular job is to be deleted from being scheduled for execution. No further parameters are permitted.

Positional Parameter 2:

PRE
Specifies that the jobs in the preemptive queue are to be deleted.

HIGH
Specifies that the jobs in the high queue are to be deleted.

NOR
Specifies that the jobs in the normal queue are to be deleted.

ALL
Specifies that the jobs in all queues are to be deleted.

Positional Parameter 3:

LOCAL
Specifies that only locally entered jobs, as defined in parameters 1 and 2, are to be deleted.

REMOTE
Specifies that only jobs entered remotely (i.e., from a remote batch terminal), as defined in parameters 1 and 2, are to be deleted.

If omitted, both local and remote jobs are deleted.

4.3.2.4. Displaying Jobs in Scheduling Priority Queues (DISPLAY)

Function:

The DISPLAY command permits the operator to display the contents of any or all job scheduling queues on the system console screen; as specified by command parameters. All jobs within the requested queue are displayed; those jobs in a deferred status (HOLD command) are displayed with parentheses around the jobname. When all queues are requested, PRE is displayed first, followed by HIGH, and then NOR. Before a queue is displayed, a system output message is displayed specifying that the request was for LOCAL, REMOTE, or all jobs (QUEUED); the PRE, HIGH, or NOR queue display to follow; and whether or not a HOLD NEW LOCAL (HL) or HOLD NEW REMOTE (HR) command is currently in effect for that queue. If no jobs are found in the queue requested, a system output message is displayed stating that condition.

Format:

Positional Parameter 1:

JBQ
Specifies that the command applies to all jobs in a job scheduling priority queue, as further defined by positional parameters 2 and 3.

Positional Parameter 2:

PRE
Specifies that the jobs in the preemptive scheduling priority queue are to be displayed.

HIGH
Specifies that the jobs in the high scheduling priority queue are to be displayed.

NOR
Specifies that the jobs in the normal scheduling priority queue are to be displayed.

ALL
Specifies that the jobs in all scheduling priority queues are to be displayed.

Positional Parameter 3:

LOCAL
Specifies that only locally entered jobs in the specified queues are to be displayed.

REMOTE
Specifies that only jobs entered remotely (i.e., from a remote batch terminal) in the specified queues are to be displayed.

If omitted, both local and remote jobs are displayed.

4.3.2.5. Changing a Job Scheduling Priority (CHANGE)

Function:

The CHANGE command changes the scheduling priority of a specific job. If a deferred job is placed into a new (changed) scheduling priority queue, the job retains its deferred status. Likewise, if a job is placed into a queue that is under the influence of a HOLD NEW command, it too will become deferred. The job is put on the end of the new queue; last examined for scheduling for execution in that queue.

Format:

```
CHANGE jobname,{PRE }
               {HIGH}
               {NOR }
```

Positional Parameter 1:

jobname
Specifies the particular job that is to have its scheduling priority changed.

Positional Parameter 2:

PRE
Specifies that the job defined in positional parameter 1 is to be moved into the PRE scheduling priority queue.

HIGH
Specifies that the job defined in positional parameter 1 is to be moved into the HIGH scheduling priority queue.

NOR
Specifies that the job defined in positional parameter 1 is to be moved into the NOR scheduling priority queue.

4.3.3. Job Execution

When a job is being executed, the operator can control the processing of that job through the use of job execution commands. These commands allow you to:

- suspend a job under execution;
- restart a job that has been suspended; and
- raise or lower the switching priority level of a job being executed.

A job is never executing unless the job name is displayed in the top line of the system console screen. If a job is rolled out, an asterisk (*) is displayed next to the job name on this line. Any command or unsolicited message to that job will be rejected; reenter the message or command when the asterisk is no longer displayed.

4.3.3.1. Suspending a Job in Progress (PAUSE)

Function:

The PAUSE command suspends processing of a job. The command may be given at any time and job processing is suspended immediately. If the job is between job steps, PAUSE will take effect at the beginning of the next job step. The PAUSE command permits you to mount a new volume on a tape unit or disk drive, replace paper on the printer, or place more cards in the card reader. The suspended job is reactivated by the GO command.

Format:

```
PAUSE jobname
```

Positional Parameter:

jobname
Specifies the name of the job whose processing is to be suspended.

4.3.3.2. Activating a Suspended Job (GO)

Function:

The GO command reactivates a job suspended by the PAUSE command or by job control operations. Job control suspends processing of a job when it issues instructions to mount a new volume on a tape unit or disk drive. The GO command also is required as a response to a message from the system preceded by an asterisk (*).

Format:

```
GO jobname
```

Positional Parameter 1:

jobname
Specifies the job to be reactivated after execution had been temporarily suspended.

4.3.3.3. Changing a Job Switching Priority (SWITCH)

Function:

The SWITCH command changes the switching priority level for a job under execution. A job assigned a higher switching priority level has priority over lower switching priority level jobs for control of the central processor. If a job is changed to a higher switching priority level than another job currently being executed, the lower switching level job will often be processed slower then the high switching level job. The number of switching priority levels a job can be raised or lowered is governed by the number of switching priority levels established at system generation time. Switching priority levels are from 1 to n, where 1 is the highest priority level and n is the lowest. If the SWITCH command exceeds the upper or lower limit of these levels, the system automatically changes the number of levels the job can be raised or lowered, so that the job remains within the preset switching priority limits. When any job switching priority is changed, all tasks of that job retain the same switching priority relative to each other; therefore, if any task of a job exceeds the upper or lower switching priority limit, all the tasks of the job move only by the number of queue positions that the highest or lowest priority task can be moved within the switching priority limits. For this reason, the system may automatically reduce the number of queue positions that the job's switching priority may be changed.

Format:

```
SWITCH jobname,{+number-of-queue-positions}
               {—number-of-queue-positions}
```

Positional Parameter 1:

jobname
Specifies the name of the job whose task switching priority is to be changed.

Positional Parameter 2:

+number-of-queue-positions
Specifies the number of switching priority levels a job is to be raised.

—number-of-queue-positions
Specifies the number of switching priority levels a job is to be lowered.

4.3.4. Job Termination

Job termination commands permit the operator to terminate the processing of a job, or a symbiont or transient, as defined by the command parameters.

4.3.4.1. Cancelling a Job in Progress (CANCEL)

Function:

The CANCEL command immediately halts all processing of a job or symbiont. The CANCEL command may be issued at any time during processing of a job and results in the immediate termination of the job step being executed at the time the CANCEL command is given, plus any subsequent job steps scheduled for the job. The job run library file for the job also is deleted.

Format:

```
CANCEL {jobname[,{D}]}
       {       {N}}
       {symbiont,S }
```

Positional Parameter 1:

jobname
: Specifies the name of the job whose processing is to be immediately terminated and whose job run library file is to be deleted.

symbiont
: Specifies the 2-character name of the symbiont to be terminated.

Positional Parameter 2:

D
: Specifies that a dump is to be taken when the job is terminated, regardless of the dump option specified in the job control stream for the job.

N
: Specifies that no dump is to be taken when the job is terminated, regardless of the dump option specified in the job control stream for the job.

S
: Specifies that the name specified in positional parameter 1 is the name of a symbiont. A dump is always taken when a symbiont is terminated.

If omitted, the job control dump options remain in effect. Positional parameter 2 must be specified when a symbiont is being cancelled.

4.3.4.2. Stopping Execution of Symbiont or Transient (END)

Function:

The END command terminates execution of a system symbiont or transient routine. This command is usually used to end a dump.

Format:

```
END system-routine,{jobname  }
                  {symbiont,S}
```

Positional Parameter 1:

system-routine
: Specifies the name of the symbiont or transient routine whose execution is to be stopped.

Positional Parameter 2:

jobname
Specifies the job that is using the system routine whose execution you want stopped (i.e., stopping the execution of a dump for a particular job).

symbiont
Specifies the 2-character name of the symbiont that is using the system routine whose execution you want stopped (for example, stopping the execution of a dump for a particular symbiont).

Positional Parameter 3:

S
Specifies that the name specified in positional parameter 2 is the name of a symbiont.

4.3.4.3. Terminating a Job (STOP)

Function:

The STOP command terminates a specific job at the end of the currently executing job step. This command provides for orderly termination of the job.

Format:

STOP jobname

Positional Parameter 1:

jobname
Specifies the job whose processing is to be terminated in an orderly sequence.

4.4. SELECTED-OCCASION OPERATOR COMMANDS

During the course of processing a job, the operator may be required to enter system-oriented commands to obtain information or make changes not involved with the execution of a particular job. These commands enable you to:

- display or dump an area of main storage;
- display information on jobs being processed, current system I/O device status, and outstanding requests and commands;
- clear the system console screen of all but outstanding output messages;
- change the system time, date, or I/O configuration;
- control software-detected hardware error logging; and
- read the volume serial number of a mounted disk or tape volume.

The commands enabling you to perform these functions are described in this section. Commands required to control spooling services, data communications, and system utility functions are described in Sections 5, 6, and 7.

4.4.1. Displaying Portions of Main Storage (DISPLAY)

Function:

The DISPLAY command causes selected areas of main storage to be displayed at the system console.

Format:

```
DISPLAY address[,{jobnumber}]
                 {0       }
```

Positional Parameter 1:

address
: A hexadecimal number that identifies a job-relative address if a job number is specified in positional parameter 2; otherwise, an absolute address.

Positional Parameter 2:

jobnumber
: Specifies the job identification number of the job in main storage whose relative address is to be displayed.

0
: Specifies that the address entered in positional parameter 1 is an absolute address.

It should be noted that, after a DISPLAY command is issued and the display symbiont (DI) is loaded, the following output message appears on the system console:

```
▷ji? addr[JOB#n] contents-of-selected-addr Y,N,NXT?
```

where:

j
: Is the job identification number (1 through 7) assigned to the display symbiont (DI).

i
: Is a message number (1 through F).

addr
: Is the address of the main storage location being displayed, in hexadecimal.

JOB#n
: Identifies the address being displayed as a job-relative address and identifies the job region by its job number. If JOB#n is not displayed, the address being displayed is an absolute address.

contents-of-selected-addr
: Is the hexadecimal representation of the contents of the selected main storage address.

Y
: Is a message response code that causes the display symbiont to display the next sequential main storage location.

N
: Is a message response code that causes the display symbiont to be terminated.

NXT

Is a reminder to the operator that another nonconsecutive main storage location may be displayed without recalling the display symbiont, simply by responding to this output message with the solicited input message:

```
▷ji addr[,{jobnumber}]
          [{■       }]
```

where:

j
: Is the job identification number (1 through 7) assigned to the display symbiont (DI).

i
: Is a job-relative message number (1 through F).

addr
: Is the main storage location to be displayed.

jobnumber
: Is a decimal number from 1 to 7 that identifies the main storage address being keyed in as a new job-relative address. This must be included each time a new address is specified.

■
: Identifies the main storage address being keyed in as an absolute address.

When no more addresses are to be displayed, the operator should terminate the display symbiont by responding to the last display message with the solicited input message:

```
▷ji N
```

Otherwise, the last message output generated by it will continue to be displayed on the system console.

4.4.2. Initiating a Main Storage Dump (DUMP)

Function:

The DUMP command causes a specific job step region in main storage, which is currently being executed, to be dumped to a printer for printing. At completion of the dump, the dumped job step is lost; however, the remaining job steps of the specified job, if any, are returned to job control for execution.

If the DUMP command is keyed in while the files associated with a job are open, a dump is not performed on the job step, but instead a cancel function (4.3.4.1) is performed to terminate the job.

Format:

```
DUMP[(did)] {jobname     }
            {symbiont,S}
```

If no device is identified, the first available printer, as defined when the system was generated, is used.

Positional Parameter 1:

jobname
Specifies the job whose job step region in main storage is to be dumped to the system printer.

symbiont
Specifies the 2-character symbiont name whose region in main storage is to be dumped to the system printer.

Positional Parameter 2:

S
Specifies that positional parameter 1 is the name of a symbiont, rather than a job name.

4.4.3. Displaying System Information (MIX)

Function:

The MIX command displays tables of different aspects of system information.

Format:

```
MIX { DA                          }
    { VI                          }
    { SQ                          }
    { SI                          }
    { DS[,device-address]         }
```

Positional Parameter 1:

DA
Causes the following information to be displayed on the system console screen:

- Job names of all jobs being processed
- Job numbers of all jobs being processed
- Regions of main storage that are allocated to jobs
- Device numbers of devices allocated to jobs being processed
- Priorities of jobs being processed
- Any variance of the storage key from the job key or variance in protection mode within a user job (e.g., a user job should be in read/write mode protect; if a region is in write protect mode only, that region's start-end addresses will be displayed)

VI
Displays device identification of devices having a mounted volume and the volume serial number of each volume.

SQ
Displays a list of outstanding symbiont requests, including unprocessed queued operator commands. (These are console commands that could not be processed immediately.)

SI

Displays system information including: supervisor name, release-id, date, time, RES device address, RUN device address, RDR device address, SPOOL device addresses, total available main storage, COS-size.

DS

Causes the following information to be displayed on the system console screen for each device in the system if no device is specified in positional parameter 2, or for only the device specified in positional parameter 2:

- The device address of each device
- Whether the device is up or down (UP or DO)
- Whether the device is available or not (Y or N)
- Whether the device is sharable or not (Y or N)
- The job numbers of all the jobs to which the device is allocated
- The job numbers of all the jobs for which the device is reserved
- If device is a diskette, will show 'IN USE' if currently being used

Positional Parameter 2:

device-address

This parameter is used in conjunction with the DS parameter to identify a particular device or group of devices for which status information is desired. One, two or three characters may be used. A 3-character key-in specifies the exact device to be displayed. If two characters are keyed in, the group of devices whose device addresses start with those two characters will be displayed. A 1-character key-in may be used to display information about all devices that have addresses beginning with the character keyed. If this parameter is omitted, the status of all the devices in the system is displayed.

Examples:

MIX DS,4	Displays any configured devices from 400 to 499
MIX DS,40	Displays any configured devices from 400 to 409
MIX DS,400	Displays device 400 only

4.4.4. Reconstructing Console Display (REBUILD)

Function:

The REBUILD command clears all information from the system console, then restores the first line on the system console and rewrites all outstanding question and action request output messages on the screen. All displays other than unanswered questions and action requests are lost.

Format:

REBUILD

Positional Parameters:

No positional parameters are required for the REBUILD command.

4.4.5. Setting Simulated Day Clock (SET CLOCK)

Function:

The SET CLOCK command resets the time of day in the system-simulated day clock. This should normally be done at midnight to change from 24:00:00 to 00:00:00.

Normally, the date and time are changed automatically at midnight of each day. If this function was excluded at system generation, the operator would use the SET CLOCK command to reset the day clock and the SET DATE command to reset the date.

Format:

```
SET CLOCK,hh:mm:ss
```

Positional Parameter 1:

CLOCK
: Specifies that the simulated day clock is to be set to the time specified in positional parameter 2.

Positional Parameter 2:

hh:mm:ss
: Specifies the time to which the simulated day clock is to be set, as follows:

  hh
  : Specifies hours (00 through 99).

  mm
  : Specifies minutes (00 through 59).

  ss
  : Specifies seconds (00 through 59).

4.4.6. Setting Date Field (SET DATE)

Function:

The SET DATE command resets the calendar date in the system information block date field. This should normally be done at midnight, in conjunction with the SET CLOCK command (4.4.5), which resets the time of day.

Normally the date and time are changed automatically at midnight of each day. If this function was excluded at system generation, the operator would use the SET DATE command to reset the date and the SET CLOCK command to reset the day clock.

Format:

```
SET DATE,yy/mm/dd[,yyddd]
```

Positional Parameter 1:

DATE
: Specifies that the calendar date in the system information block date field is to be changed to the date specified in positional parameter 2.

Positional Parameter 2:

yy/mm/dd
Specifies the date to which the calendar date in the system information block date field is to be changed, as follows:

yy
Specifies year (00 through 99).

mm
Specifies month (01 through 12).

dd
Specifies day (01 through 31).

Positional Parameter 3:

yyddd
Specifies the ordinal date, where yy is the year (00 through 99) and ddd is the day of year (001 through 366). This date is maintained in a separate part of the system information block and is used by data management routines that check the label fields.

If omitted, this field in the system information block is set to the ordinal date corresponding to the yy/mm/dd date specified in positional parameter 2.

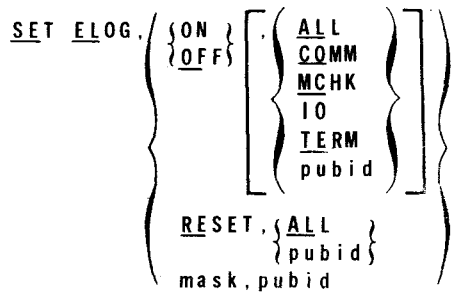
4.4.7. Setting Error Log (SET ELOG)

Function:

The SET ELOG command is used after IPL to control physical I/O control system (PIOCS) error logging into the YELOG file on SYSRES. Software-detected hardware errors to be logged can be specified by the operator through the use of the SET ELOG command and command parameters. Communications, I/O device, machine check, and I/O termination record error logging can be turned on or off when you enter specified combinations of the command and parameters. The IPL procedure automatically turns on error logging and all error logging functions. Any changes to this all-on condition that you enter are lost when the system is reloaded.

SET ELOG is also used to redefine the characteristics of a loggable error for a specified device. The mask required to define an error that can be logged is determined by device type and included at SYSGEN. The masks are automatically loaded through the IPL procedure. Should your Sperry Univac customer engineer ask you to change the characteristics of a mask, enter the command with the actual hexadecimal characters of the new mask inserted as the value of the mask parameter, and with the pubid specified.

Format:



Positional Parameter 1:

ELOG
Specifies that the PIOCS error logging processor is to be set to some condition.

Positional Parameter 2:

ON
Specifies that the function of logging errors is to be turned on. When parameter 3 is included, error logging is turned on for that parameter specification only.

OFF
Specifies that the function of logging errors is to be turned off. When parameter 3 is included, error logging is turned off for that parameter specification only.

RESET
Specifies that error sense bytes are to be reset to their original IPL specification. The pubid or ALL specifications of parameter 3 must be used with RESET.

mask
Represents a 1- to 4-hexadecimal-character sense byte mask used to determine if an I/O error can be logged. Mask can only be used with the pubid specification of parameter 3. When a mask is entered with the SET ELOG command, the new mask replaces the mask set at IPL for the pubid identified. The new mask remains until RESET is entered or IPL occurs. The mask is left-justified and zero-filled.

Positional Parameter 3:

ALL
Is valid for all parameter 2 specifications except mask and has the following functions:

ON
Specifies that all loggable errors (i.e., communications, machine check, I/O device, and I/O termination record error logging) are to be logged for all devices in the system.

OFF
Specifies the same as ON except that all error logging is off.

RESET
Specifies that all device sense bytes are to be reset to their original specification.

COMM
Specifies that communications error logging is to be turned on or off as directed by parameter 2.

MCHK
Specifies that machine check error logging is to be turned on or off as directed by parameter 2.

IO
Specifies that all I/O device error logging is to be turned on or off as directed by parameter 2.

TERM
Specifies that all I/O termination record error logging is to be turned on or off as directed by parameter 2.

pubid
Represents one, two, or three hexadecimal characters specifying a channel, subchannel or control unit, and device, respectively. Pubid can be used with any parameter 2 specification.

- If pubid is specified as one character, all devices on that channel are directed by the parameter 2 specified.
- If pubid is specified as two characters, all devices on that channel and subchannel or control unit are directed by the parameter 2 specified.
- If pubid is specified as three characters, a specific device on that channel and subchannel is directed by the parameter 2 specified.

If omitted, the error logging condition for each parameter 3 remains unchanged, i.e., turned on or off as previously set. Thus, if you omit parameter 3 and enter the command SET ELOG,OFF, the error logging function is turned off (suspended) while the logging condition for parameter 3 specifications remains as previously set. If you omit parameter 3 and enter the command SET ELOG,ON, the error logging function resumes, but only for those parameter 3 specifications previously turned on.

When the ELOG file is nearly full (only 200 more I/O errors can be recorded), ELOG informs the operator of the near-full condition. When the message "LOG FILE IS NEARLY FULL" appears, the operator should run the system-supplied ONUERL job using the RUN/RV command to print the contents of the ELOG file.

NOTE:

In order to run ONUERL, the disk volume that contains the YRUN file and the SYSRES disk volume must be for similar devices (both 8414 disks, for example).

The ONUERL job control stream permits the operator to execute the ONUERL program with parameter options either preset (default) or overridden by // PARAM cards. Enter the following command to execute the ONUERL program with default conditions for parameter options:

```
RV ONUERL
```

Enter the following command to execute the ONUERL program with overriding // PARAM cards:

```
RUN ONUERL,,CARD=YES
```

The ONUERL program automatically turns error logging off as the program begins, then turns logging on automatically when the program terminates (unless you override the ON/OFF parameter). When you use overriding // PARAM cards for one or more parameters, the parameters are entered via the card reader, one parameter per card. A // FIN card must follow the last parameter card. The ONUERL parameters, including operator options and preset default conditions, are provided in Table 4–1.

Table 4—1. ONUERL Program Parameters (Part 1 of 2)

```
// PARAM ESUMFIL={INIT}
                 {YES }
                 {NO  }
```

INIT
Formats the YESUM summary file, which is used for error analysis.

YES
Permits summary file to be updated for duration of program execution.

NO
Turns off summary file updating for duration of program execution.

Table 4—1. ONUERL Program Parameters (Part 2 of 2)

```
// PARAM OPEN={BEGIN  }
              {CURRENT}
              {PRIOR  }
```

BEGIN
Starts retrieving error data at oldest record and includes all records. (Must be entered if ELOGDMP=YES is specified.)

CURRENT
Starts retrieving error data at the next record following point where most recent retrieval terminated.

PRIOR
Starts retrieving error data at the same record the most recent retrieval started.

```
// PARAM ELOGDMP={YES}
                 {NO }
```

YES
Dumps all error data from YELOG; to be used for troubleshooting. (OPEN=BEGIN must be entered when this option is used.)

NO
Turns off the dump option.

```
// PARAM ELOG={ON }
              {OFF}
```

ON
Permits error logging to continue during ONUERL program execution.

OFF
Turns off error logging during ONUERL program execution. (At program termination, logging is returned to the condition it was set prior to executing ONUERL.)

```
// PARAM LOG-ID={SYSGEN-specified-installation-name-and-number}
                {installation-name-and-number               }
                {NONE                                       }
```

SYSGEN-specified-installation-name-and-number
Fills the appropriate error log report header space with the SYSGEN-specified installation name and serial number.

installation-name-and-number
Fills the report header space with a different installation name and serial number.

NONE
Fills the error log report header space for installation name and number with spaces.

NOTES:

1. In the // PARAM LOG-ID statement, the error log report header space designated for the installation name and serial number is 38 characters total in length – the first 34 characters are reserved for the installation name; the last 4 characters are reserved for the installation serial number.

2. ESUMFIL=INIT must be executed as soon after SYSGEN as possible, in order to format write the summary file YESUM.

The ELOG file can become completely full only if:

- the ONUERL program was not initiated when the "LOG FILE IS NEARLY FULL" message was displayed; or

- the ONUERL program was run with LOG=ON specified, thus allowing the file to become full.

If the ELOG file becomes full, ELOG asks the operator if he wants to turn off error logging or wraparound to the beginning of the log file. When the message

LOG FILE IS FULL W(RAP) OR O(FF)

appears, the operator responds using the solicited message format:

```
Ø i {O}
    {W}
```

where:

i
Identifies the console message being answered.

O
Turns off the PIOCS error logging processor.

W
Informs the PIOCS error logging processor to wrap around the log file and continue logging.

If printing of the ELOG file has been initialized previously (with error logging set to ON) you should respond with the letter W.

If printing has not been initiated, enter O to turn off logging, and then initiate the printing of the ELOG file as previously directed.

The summary report obtained as a result of running the ONUERL program provides a comprehensive listing of all errors contained in the YELOG file, as defined by the program parameters. The main body of the report consists of a single line for each error log entry. The entries are sorted chronologically by channel device number. The report also includes a summary of total I/O count and sense byte errors per device, and a listing of machine check errors.

A sample ONUERL summary report is illustrated in Figure 4–2.

```
ONUERL-ERROR LOG REPORT      _________________________________          SERIAL# ____ yy/mm/dd hh.mm  PAGE    1

OS/3 VER. 6/0, REV.  x, DATE yy/mm/dd, TIME hh.mm.ss, FLAGS xxxxxxxx, CHARACTERISTICS xxxx

SYSRES xxx, PRINTER xxx, READER xxx, $Y$ELOG xxx, MAIN STORAGE SIZE xxxxxxxx, USER MEMORY SIZE xxxxxxxx



***** UNISCOPE 100 CONSOLE *****
      CHAN. DVC ADDR.: 00 00
      FEATURES :
      PRINTER, 24x80 SCREEN
      LGERMSK: xxxx, LGEMSK: xxxx



DEVICE: 8430, CHDV: xxxx, FEATURE BYTES: xxxx, PHYSICAL ADDR: x, DATE yy/mm/dd

                                          (*=BCW)       CCW / BCW                DISK ADR DV/SC
  TIME   JOB NAME  VSN   RT R/U OPR I/O'S * CCAAAAAAFFBBBBBB CCAAAAAAFFBBBBBB CCC HH RR STATS. EMSK SNS BYTES 0-5
hh.mm.ss ________ ______ __ xxx  x   xxxxx  ________________ ________________ ___ __ __ __ __  ____ ____ ____ ____
8430 SSB 6-23: ____ ____ ____ ____ ____ ____ ____ ____ ____

   .
   .
   .

DEVICE: U-16, CHDV: xxxx, MODE: xx, DATE yy/mm/dd

                                          (*=BCW)       CCW / BCB                   XPCT ACT. DV/SC
  TIME   JOB NAME  VSN   RT R/U OPR I/O'S * CCAAAAAAFFBBBBBB CCAAAAAAFFBBBBBB MODE BLCK BLCK STATS. EMSK SNS BYTES 0-5
hh.mm.ss ________ ______ __ xxx  x  xxxxx   ________________ ________________  __  ____ ____ __ __  ____ ____ ____ ____
   .
   .
   .

  TIME   JOB NAME START   SIZE  SIO MACH CHK OLD PSW PGM CHK OLD PSW  RELOC RG SUP R-0  SUP R-1
hh.mm.ss ________ ______ ______  _  ________________ ______________  ________ ________ ________
```

Figure 4—2. Sample Error Log Summary Report

The description of each summary report heading follows:

DEVICE TYPE
A 4-character identification of device type assembled into the PUB at supervisor generation time.

DEVICE ADR
A combination address consisting of the channel for this I/O (cochannel or primary) and the control unit device address from the PUB.

DATE, TIME
The date and time the error log was written to the error file.

JOB NAME
The name of the job to which the device was allocated.

VSN
The volume serial number of the volume mounted on the device.

RT
Record Type: 20 = I/O Error, 31 = Operator Response, 40 = Termination

RCVRY
The number indicates the number of retries the supervisor initiated in trying to recover from the error. The letter following the retry number indicates the error's disposition.

R = recovered by supervisor.

U = not recovered by supervisor; error passed to user program.

VALID I/O's
Contains the number of valid I/O's prior to the error. It indicates the frequency of errors for the device.

ERROR BCW
The error or failing buffer control word. (Entire BCW, indicated by an * preceding BCW.)

CCAAAAAA: CC = command code, AAAAAA = address

FFBBBBBB: FF = flag, BBBBBB = count

For a description of the BCW and CCW fields, see MH2737, SPERRY UNIVAC Processor Type 3029-02, -03 Functional Analysis and Servicing. On selector channels, two CCWs will be displayed.

DISK
If the failing device was a disk and the first CCW was a seek, the cylinder (CCC) and the head (HH) identifications are recorded. If, in scanning the chain, a search-equals command is recognized, the record number (RR) is recorded.

DVC/SC STATS
The device and subchannel status recognized when the error recovery sequence was completed.

SENSE BYTES
The bytes recognized when the first unit check of the error recovery sequence was initiated. Up to six sense bytes are displayed for all devices except the 8430 disk drives, which display 24 sense bytes, on the following line.

MODE
On tape devices, normal and current tape mode will be displayed.

XPCT BLCK
Expected block number

ACT BLCK
Actual block number.

The MODE, EXPCT BLCK, and ACT BLCK headings, along with CCCHHRR, will not be displayed on unit record devices.

Machine Check Error Display
MACH CHK OLD PSW – PSW at time of error

PGM CHK OLD PSW – PSW at time of error

RELOC RG – Contents of the relocation register

SUP R-0 – Contents of supervisor

SUP R-1 – Registers 0 and 1

4.4.8. Setting Up Physical Unit Block (SET IO)

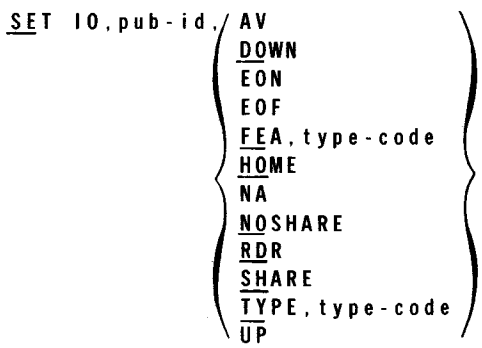
Function:

The SET IO command allows you to set specific bits in the physical unit blocks (PUB), which define operational characteristics and assignments of I/O devices. All devices or subsystems must be set *DOWN* before attempting operation on the device, such as forms loading, changing ribbon, etc. This is required if the processor is to continue operation with other peripheral devices while the subject device is undergoing isolated operations. Before a procedure is performed or when power is turned off for the device, key in SET IO, pub-id, DOWN. After the offline procedure is completed or after turning on power independently of the processor, key in SET IO, pub-id, UP.

NOTE:

The system automatically sets all devices or subsystems not online during IPL time to not available. After IPL, any attention interrupt from a device causes that device to be set available.

Format:



Positional Parameter 1:

IO
Specifies that the change is to be made in the PUB specified by positional parameter 2.

Positional Parameter 2:

pub-id
Specifies a 1- to 4-character numeric field identifying the PUB to be changed (device addresses are usually attached at a visible location on the device cabinet). The following are the PUB identifiers:

0000	integrated peripheral channel (console)
0001	integrated peripheral channel (card reader)
0002	integrated peripheral channel (printer)
0003	integrated peripheral channel (card punch)
0100 through 0177	multiplexer channel
0300 through 0307	integrated disk adapter
0400 through 0477	selector channel 4
0600 through 0677	selector channel 6

Positional Parameter 3:

AV
Specifies that the device identified by positional parameter 2 is available for assignment to user jobs.

DOWN
Specifies that the device identified in positional parameter 2 is not to be considered for assignment to user jobs.

EON
Specifies that error logging is to be turned on for the specified device.

EOF
Specifies that error logging is to be halted for the specified device.

FEA
Allows modification of the feature bytes of the device specified in positional parameter 2. The type code in positional parameter 4 is the new feature bytes.

HOME
Synchronizes the operating system with the physical paper position of an 0768, 9200, or 9300 printer during a home operation.

NA
Specifies that the device identified by positional parameter 2 is not available for assignment to user jobs, although the device is physically up.

NOSHARE
Forbids allocation of the device specified in positional parameter 2 to more than one program simultaneously.

RDR
Assigns the new pub-id specified in positional parameter 2 as the system card reader.

SHARE
Permits the device specified in positional parameter 2 to be shared by more than one program simultaneously.

TYPE
Allows modification of the type bytes of the device specified in positional parameter 2. The type code specified in positional parameter 4 describes the modification.

UP
Specifies that the device identified in positional parameter 2 is to be made available for assignment to user jobs.

Positional Parameter 4:

type-code
A 1- to 4-character field specifying the device, its options, and features desired.

4.4.9. Reading a Mounted Volume Serial Number (AVR)

Function:

The AVR command initiates reading the volume serial number of a premounted prepped disk pack or magnetic tape volume and storing it in the device physical unit block. This command is required when a disk pack or magnetic tape is mounted on a unit that does not have an attention interrupt capability (i.e., UNISERVO VI-C Magnetic Tape Subsystems).

Format:

```
AVR device-addr[,device-addr][,device-addr]
```

Positional Parameters 1 through 3:

device-addr
Specifies the addresses of the devices on which the volumes to be recognized are mounted.

5. Spooling Services

5.1. GENERAL

When card readers, printers, or card punches (paper peripherals) are required by a job, they generally must be dedicated to the job at the point where requested. The serial nature of these devices does not lend itself to multijobbing. Because the speed of these devices is far less than the speed of the other resources in the system, they drastically impede throughput if allowed to tie up the system. To eliminate this problem, OS/3 provides automatic spooling (simultaneous peripheral operations online) that buffers the input/output information to and from disk. This permits the card and print devices to operate at their rated speeds while the user jobs are processed, without regard to device contention. In OS/3, spooling requires no changes in any user programs.

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

- General spooling
- Input reader
- Output writer

These commands are divided into two types:

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

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

- save the spool log subfiles for each job processed in the system;
- produce a job accounting report; and
- use the subfiles for your own programs.

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

5.2. GENERAL SPOOLING COMMANDS AND MESSAGES

General spooling commands allow you to:

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

The formats for these commands may include:

- spool file directories (Table 5–1) to identify one of the input or output directories that make up the spool file; and
- spool file modifiers (Table 5–2) to further identify the subfiles that are being referenced.

Table 5—1. Spool File Directories

Directory	Subfile Function
LOG	Subfile input or output is on designated log file.
PRINT	Subfile output is to designated printer.
PUNCH	Subfile output is to designated card punch or diskette.
RBPIN	Remote batch processing input is from card reader.
RBPPR	Remote batch processing output is to printer.
RBPPU	Remote batch processing output is to card punch.
RDR	Subfile input is from designated card reader or diskette.

NOTE:

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

Table 5—2. Spool File Command Modifiers

Modifier	Subfile Identification
ACCT={acctno / *}	Account number of up to four characters
CART={cartridge-id / *}	Print cartridge identification of up to eight characters
DEV=(768 / 770 / 773 / 776 / 778 / 9300 / *)	Any device of the type designated that is available. The designation 9300 refers to a SPERRY UNIVAC 9300 System that is being used as a 90/30 system peripheral device.
FILE={filename / *}	File name of up to eight characters
FORM={formname / *}	Form name of up to eight characters
ID=remote-id	Remote file identification of up to six characters (output writer BX function only)
JOB={jobname / *}	Job name of up to eight characters
LBL=labelname	Label name of up to 8 characters for diskette, up to 17 characters for card reader
STEP=stepno	Job step number of three characters (left-justified with zeros)
VOL=volno	Volume number of up to six characters (for diskette only)

NOTE:

An asterisk (*) specifies that the output writer is to group the contents of the spool file according to the modifier type specified, and process the groups on a first-in/first-out basis. This provides a complete spool file listing, grouped according to the modifier specified.

5.2.1. Displaying the Status of Spool Subfiles (DISPLAY)

Function:

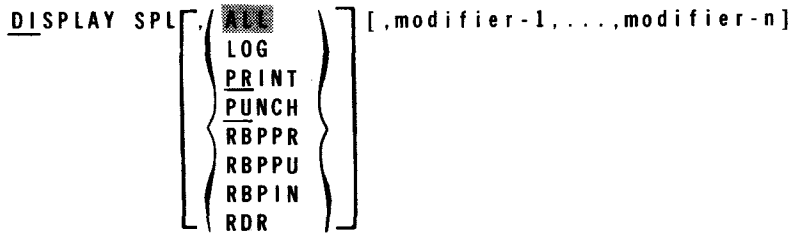
The DISPLAY command causes all requested information for either currently active or complete subfiles to be displayed on the system console. If there are no spool subfiles in the requested directory, a system message that the spool file is empty is displayed.

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

Format 1:

DISPLAY ACT [,(ALL / PRINT / PUNCH / RBPPR / RBPPU)] [,modifier-1,...,modifier-n]

Format 2:



Format 3:

DISPLAY CNSLG

Positional Parameter 1:

ACT
: Indicates that only the currently active subfiles are being displayed.

SPL
: Indicates that only the completed subfiles are being displayed.

CNSLG
: Indicates that only the console log is being displayed.

Positional Parameter 2:

Identifies one of the spool file directories permitted, as described in Table 5–1.

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

Positional Parameter 3:

modifier-1,...,modifier-n
: Further identify the subfiles being referenced, as described in Table 5–2. Modifiers should not be used with format 3. If any are entered, the command is ignored.

Solicited message entries (4.2.2.2) are required to direct and terminate the display when ACT or SPL is specified with the DISPLAY command. If DISPLAY SPL is used, the display function displays the number, if any, of subfiles queued (ready for processing), in hold, and in progress (being processed by the output writer). The following message is also displayed.

DI02 SPOOL FILE DETAILS? ***Y,N,Q,H,I,S,SQ,SH,SI***

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

Y Displays all spool details

N Does not continue display

Q Displays all queued files

H Displays all files in hold

I Displays all files currently being processed by output writers

S Abbreviated display of all files

SQ Abbreviated display of all queued files

SH Abbreviated display of all files being held

SI Abbreviated display of all files in progress

Depending on the response, the following displays are produced:

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

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

- If the response was Y, Q, H, or I and the RDR directory was specified, the following appears on the console screen:

  DI09 DEVICE-TYPE RDR TOTAL-CARDS nnnnn [VOL xxxxxx]
  DI10 LABEL XXXXXXXXXXXXXXXXX CONTINUE? ***Y,N***

  NOTE:

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

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

  DI11 JOB=jobname PROG=programname FORM=formname { PAGES / CARDS / LINES } = nnnnn

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

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

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

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

Examples:

1. Operator keyin:

   DI SPL,ALL

   Function requested:

   Display all closed subfiles in the spool file.

2. Operator keyin:

   DI SPL,PRINT

   Function requested:

   Display all closed subfiles in the print directory.

3. Operator keyin:

   DI SPL,PUNCH,JOB=XYZ

   Function requested:

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

4. Operator keyin:

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

   Function requested:

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

5. Operator keyin:

   DI ACT

   Function requested:

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

6. Operator keyin:

   DI CN

   Function requested:

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

5.2.2. Deleting Closed Spool Subfiles (DELETE)

Function:

To delete closed spool subfiles from the spool file. Only closed subfiles will be deleted; subfiles that are currently being processed are not deleted.

Format:

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

Positional Parameter 1:

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

Positional Parameter 2:

Identifies one of the spool file directories permitted, as described in Table 5–1. Entering LOG does not affect logs destined for a remote device. If ALL is entered, the RBPPR, RBPPU, and logs destined for a remote device are not affected.

Positional Parameter 3:

modifier-1,...,modifier-n
Further identifies the subfiles being referenced, as described in Table 5–2.

When the DELETE SPL command is entered, the system displays the message

DE01 xxx SPOOL FILES DELETED

where xxx is the number of spool files deleted.

Examples:

1. Operator keyin:

   DE SPL,ALL,JOB=ABC

   Function requested:

   Delete all closed subfiles associated with job ABC.

2. Operator keyin:

   DE SPL,PRINT,JOB=XYZ

   Function requested:

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

5.2.3. Placing Spool Subfiles on Hold (HOLD)

Function:

To place active and closed spool subfiles in a hold condition. Subfiles in this condition are not available for processing by an output writer or job control symbiont until they are subsequently released by the operator using the BEGIN command (5.2.4).

Format 1:

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

Format 2:

```
HOLD {ACT}[,(LOG  )]
     {SPQ} [(PRINT)]
           [(PUNCH)]
           [(ALL  )]
```

Positional Parameter 1:

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

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

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

Positional Parameter 2:

Identifies one of the spool file directories permitted, as described in Table 5–1. Entering LOG does not affect logs destined for a remote device. If no parameter is specified or ALL is entered, the RBPPR, RBPPU, and logs destined for a remote device are not affected.

Positional Parameter 3:

modifier-1,...,modifier-n
: Further identifies the subfiles being referenced, as described in Table 5–2. Modifiers should not be used with format 2. If any are entered, the command is ignored.

When the HOLD SPL or HOLD SPQ commands are entered, the system displays the message

HO01 xxx SPOOL FILES HELD

where xxx is the number of spool files held.

Examples:

1. Operator Keyin:

   HO SPL,PRINT,CART=48-SCI

   Function requested:

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

2. Operator keyin:

   HO SPL,RDR,LBL=TEST

   Function requested:

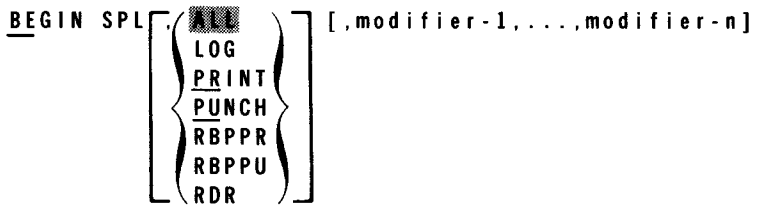
   Hold all closed subfiles in the reader directory that contain a label named TEST.

5.2.4. Releasing Spool Subfiles (BEGIN)

Function:

Removes the hold status from active and closed subfiles previously placed in the hold condition by the HOLD command (5.2.3), and releases a file placed in hold condition by the // SPL card. This function causes the output writer to be loaded automatically in burst mode according to the parameters entered in the BEGIN SPL or BEGIN SPQ commands.

Format 1:



Format 2:

Positional Parameter 1:

SPL

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

ACT

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

SPQ

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

Positional Parameter 2:

Identifies one of the spool file directories permitted, as described in Table 5–1. Entering LOG does not affect logs destined for a remote device. If no parameter is specified or ALL is entered, the RBPPR, RBPPU, and logs destined for a remote device are not affected.

Positional Parameter 3:

modifier-1,...,modifier-n

Further identifies the subfiles being referenced, as described in Table 5–2. Modifiers should not be used with format 2. If any are entered, the command is ignored.

When the BEGIN SPL or BEGIN SPQ commands are entered, the system displays the message

BE01 xxx SPOOL FILES RELEASED

where xxx is the number of spool files released.

Examples:

1. Operator keyin:

   BE SPL,PRINT,CART=48-SCI

   Function requested:

   Release all the subfiles in the print directory that require the use of a 48-character scientific print cartridge and that are currently in the hold condition. This command countermands the HOLD command presented as example 1 in 5.2.3.

2. Operator keyin:

   BE SPL,RDR,LBL=TEST

   Function requested:

   Release all the subfiles in the reader directory that contain a label named TEST and are currently in the hold condition. This command countermands the HOLD command presented as example 2 in 5.2.3.

5.2.5. Breakpointing Active Spool Subfiles (BRKPT)

Function:

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

The operator should use this function when warning messages indicating the spool file is nearly depleted are displayed on the console.

NOTES:

1. *After the breakpoint has been taken for printer or punch subfiles, the operator should call in an output writer in burst mode.*

2. *After the breakpoint has been taken for the console log subfile, the output writer is brought in automatically. The output writer recognizes the console log subfile and prints it.*

3. *Diskette subfiles may not be breakpointed.*

4. *After the BR command has been entered, a message telling the operator whether the breakpoint was successful will appear on the system console.*

Format 1:

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

Format 2:

```
BRKPT CNSLG
```

Positional Parameter 1:

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

I
: Indicates that the subfile is to be breakpointed immediately.

CNSLG
: Indicates that the console log subfile is to be breakpointed.

Positional Parameter 2:

Identifies one of the spool file directories permitted, as described in Table 5–1. For a punch subfile, either P or I may be entered. A directory should not be used with format 2; if entered, the command is ignored.

Positional Parameter 3:

JOB=jobname
: Identifies the name of the job in the subfile to be breakpointed.

Positional Parameter 4:

modifier-1,...,modifier-n
: Further identifies the subfiles being referenced, as described in Table 5–2. Modifiers should not be used with format 2. If any are entered, the command is ignored.

Examples:

1. Operator keyin:

   BR I,PU,JOB=ABC

   Function requested:

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

2. Operator keyin:

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

   Function requested:

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

3. Operator keyin:

   BR CN

   Function requested:

   Breakpoint the console log subfile.

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

Function:

The SET SPL,BURST/NBURST command is used to change the spooling criterion or subcriterion established for processing the subfiles in the spool file. Basically, subfiles can be processed in either a burst or nonburst mode. Burst mode processing allows closed and breakpointed files within subfiles to be processed by spooling symbionts before the job they are associated with has terminated. Nonburst mode processing requires that a job be terminated before any of the files associated with the job can be output by an output writer symbiont. Operating in the burst mode, therefore, permits the printed and punched output produced by a single job to be split between two or more like output devices. This tends to speed up throughput for a job, but also creates a confusion factor for the operator if more than one job is being run in the system. On the other hand, operating in the nonburst mode may slow down throughput, but ensures that all like output associated with one job is produced on one device (e.g., all print files are printed on the same printer).

When a SET SPL,BURST/NBURST command is issued, the system spooling criterion is changed immediately and all future output writers loaded by the spooling system will function accordingly. Previously loaded output writers, however, continue to function using the criterion established for them either where they were loaded or via operator command (5.4). This SET command should not be confused with the BURST and NBURST functions of the output writer commands.

Format:

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

Positional Parameter 1:

SPL
Indicates that the SET command is directed to the system spooler.

Positional Parameter 2:

BURST
Specifies that the spooler should operate in the burst mode.

NBURST
Specifies that the spooler should operate in the nonburst mode.

Positional Parameter 3:

modifier
Further identifies the subfiles being referenced, as described in Table 5-2.

5.2.7. Changing the System Accumulation of Log Files (SET SPL,DUMP/ENDDUMP)

Function:

This command is used to change the method of processing the log subfiles produced in the system. The initial method of processing is specified at SYSGEN time.

As each job in the system is terminated, its associated spool log subfile is closed and automatically output to a high-speed printer as soon as an output writer becomes available in the system. The log subfile is then either deleted from the spool LOG subfile or saved for future processing, depending on the SYSGEN option selected. This command permits you to override the SYSGEN selected option.

Format:

SET SPL, { **DUMP** / **ENDDUMP** }

Positional Parameter 1:

SPL
Indicates the SET command is directed to the system spooler.

Positional Parameter 2:

DUMP
Specifies that log files are to be accumulated for later routing to a magnetic tape or disk file.

ENDDUMP
Specifies that log file accumulation is to end.

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

Function:

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

Format:

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

Positional Parameter 1:

SPL
Indicates the SET command is directed to the system spooler.

Positional Parameter 2:

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

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

5.2.9. Changing the Job Log File Print Specifications (SET SPL,NOACT/NOLOG/NOPRINT/PRINT)

Function:

To select what job log records, if any, are to be printed at the end of each job. The initial job log print specification is established at SYSGEN time.

Format:

```
SET SPL,(NOACT  )
        (NOLOG  )
        (NOPRINT)
        (PRINT  )
```

Positional Parameter 1:

SPL
Indicates the SET command is directed to the system spooler.

Positional Parameter 2:

NOACT
Indicates that job accounting records are not to be printed.

NOLOG
Indicates that job log records are not to be printed.

NOPRINT
Indicates that both job accounting and log records are not to be printed.

PRINT
Indicates that both job accounting and log records are to be printed.

NOTE:

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

5.2.10. Changing the Console Log File Specifications (SET SPL,CNSLG)

Function:

This command allows the operator to turn the console log function on or off, to delete or retain the console log for SYSLOG accumulation, and to print or not to print the console log.

Format:

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

Positional Parameter 1:

SPL
Directs the SET command to the system spooler.

Positional Parameter 2:

CNSLG
Further directs the command to the spooler console log file.

Positional Parameter 3:

ON
Indicates that the console log function is to be turned on.

OFF
Indicates that the console log function is to be turned off.

Positional Parameter 4:

RETAIN
Indicates that the console log is to be retained for SYSLOG accumulation.

DELETE
Indicates that the console log is to be deleted from SYSLOG accumulation.

Positional Parameter 5:

PRINT
Indicates that the console log is to be printed.

NOPRINT
Indicates that the console log is not to be printed.

NOTES:

1. *If positional parameter 3, 4, or 5 is omitted, the console log file remains in the same condition or mode for that parameter that it was in before the command.*

2. *A blinking marker symbol (◥) in the rightmost position of the console line indicates that the message was not written to the console log.*

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

Function:

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

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

Format:

```
SET SPL,{TEST  }
        {NOTEST}
```

Positional Parameter 1:

SPL
: Indicates the SET command is directed to the system spooler.

Positional Parameter 2:

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

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

5.3. INPUT READER COMMANDS

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

5.3.1. Loading an Input Reader (IN)

Function:

This command causes the operating system to load an input reader symbiont that will transfer a card or diskette file into the RDR spool directory.

Card data files to be spooled must be preceded by a // DATA job control statement (for details, see the job control user guide, UP-8065, current version). The cards are read into the spool file until a // FIN or another // DATA card is detected.

When diskette data files are to be spooled, the // DATA and // FIN job control statements should not be used. If any are entered, they are included as records in the data file.

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

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

Format 1 (card reader):

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

NOTE:

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

Format 2 (diskette):

```
IN([did],label) [RETAIN]
```

NOTE:

If the data file is on a diskette, format 2 must be used (4.2.1).

Positional Parameter 1:

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

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

RETAIN
: Indicates that the spool file is to be retained after the file has been processed. This parameter is only to be used for single-volume input.

5.4. OUTPUT WRITER COMMANDS AND MESSAGES

Output writers are loaded under three conditions:

1. Automatically by the system, as required, up to the number of output devices available for their use. The output writers are set in either burst or nonburst mode, according to the mode established at SYSGEN time.

2. Automatically by the operator when the BEGIN spooling command is entered. The output writers are automatically set to burst mode by this command.

3. Manually by the operator under certain conditions, such as the system set in nonburst mode, in order to have files printed. Some of these conditions are:

   - warm start (i.e., recovering files at IPL) if set to nonburst at SYSGEN;
   - BR function if system is in nonburst mode;
   - STOP or HALT function previously entered from system console and printing of files for the job is incomplete; and
   - IN function when specifying tape input.

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

5.4.1. Loading an Output Writer (PR/PU/PD)

Function:

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

Format:

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

Command Code:

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

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

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

Positional Parameter 1:

(did)
Designates the address of the output device to be used by the output writer.

If the address specified by did is that of a tape unit, all output files processed by the output writer being called are output to tape (not valid with PD).

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

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

Positional Parameter 2:

`function-code`

Identifies the mode of operation and processing criteria under which the output writer is to function. The various function codes that may be specified in this parameter location are listed and described in Table 5–3. If omitted, the output writer is loaded in the mode (nonburst/burst) indicated by SYSGEN.

Positional Parameter 3:

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

Further identify the subfiles being referenced, as described in Table 5–2.

NOTE:

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

Table 5—3. Output Writer Function Codes (Part 1 of 2)

Function	Meaning
{BURST} {BX} [modifier-1,..., modifier-n]	Output writer is to function in the burst mode, as optionally further qualified by modifiers 1 through n. The modifiers that may be specified are listed and described in Table 5–2. (STEP, LBL, and VOL are not used with BU and BX.) Burst mode processing is described in 5.2.6. If BX is entered with modifiers, the output writer terminates after processing all files that satisfy the modifiers. If BU is entered, the output writer requests another function if more files exist that do not satisfy the modifiers.
BYPASS	Output writer is to discontinue processing the current subfile and continue processing on the next subfile. The bypassed subfile can be restarted later.
COPIES,nnn	Specifies the number of copies the output writer is to produce of each subfile it processes. The number of copies that may be specified is from 0 to 255. If zero is specified, the subfile is closed when the current processing of the file is completed. This function may not be used with PD.
DELETE	Delete subfile currently being processed from spool file and proceed with the next subfile ready to be processed.
DEVICE[,did]	Indicates that the output writer is to change the device it is currently using to print (punch) its output. If a new device (did) is identified in the function code, that device is allocated to the output writer, and the current device is deallocated. If a new device is not identified, a device having the same characteristics as the current device will replace the current device. If the new device specified is a tape unit, all subsequent output files are written to tape, as long as the same copy of the output writer remains in main storage.
DISPLAY	Output writer is to display the status of subfile being processed on the system console.
HALT	Output writer to terminate after current subfile is processed. If multiple copies of the current subfile were requested, the remaining copies will be produced when the output writer is reloaded.

Table 5—3. Output Writer Function Codes (Part 2 of 2)

Function	Meaning
HOLD	Output writer is to place the subfile it is processing in the hold state and process the next subfile. Subfiles in the hold state are not available for processing until they are released by a BEGIN command (5.2.4).
INPUT,did	Directs the output writer to accept input from the tape unit (did) identified in the function code. This function may not be used with PD.
NBURST	Output writer is to function in the nonburst mode. If this function code is received while the output writer is processing a subfile, it does not take effect until the subfile is completed. Nonburst mode processing is described in 5.2.6.
RESTART[,nnn / ,PAGE,nnnn / ,CARD,nnnn]	Restarts processing of the currently active subfile from a number of pages or cards. If no number is specified, the output writer is to restart processing from the beginning of the subfile. If only nnn is entered, the file processing will be restarted nnn pages or cards back from the current position of the file. If PA or CA is entered with nnnn, the file will be positioned back to the page or card identified by nnnn. This function may not be used with PD.
SKIP{,nnnn / ,PAGE,nnnn / ,CARD,nnnn}	Directs the output writer to skip n number of pages or cards. If only nnnn is entered, the file will be positioned nnnn pages or cards from the current position. If PA or CA is entered with nnnn, the file will be positioned forward to the page or card identified by nnnn. After positioning, a request will be made for an additional function. This function may not be used with PD.
STOP[,PAGE]	Directs the output writer to stop processing. If STOP is entered without the PAGE operand, the output writer terminates immediately. If STOP is entered with the PAGE operand, the output writer terminates after printing the complete current page. If a subfile is being processed when this function code is received, the subfile is closed but not deleted. When the subfile is accessed by another output writer, it will be processed from the point at which it was closed by the previous output writer.

NOTE:

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

Examples:

1. Operator keyin:

   PR BX,FO=*

   Function requested:

   The output writer using the printer is loaded to begin processing in burst mode, to group input by form name, and to terminate when all form name files have been processed.

2. Operator keyin:

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

Function requested:

The output writer using the printer, is loaded to begin processing all remote files associated with an ID of ABCDEF, which was created by job MYJOB. ID permits output writer to print or punch all files (logs not processed) that have a remote ID equal to the ID entered by the operator.

5.4.2. Directing the Operation of an Output Writer (Messages)

Function:

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

Formats:

- Unsolicited Messages

```
ØØ {PR}[(did)]function-code
   {PU}
   {PD}
```

where:

ØØ

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

```
{PR}[(did)] function-code
{PU}
{PD}
```

Is the message text that is used to communicate with and identify the desired output writer. As shown, it is similar to the command format used to load an output writer (5.4.1). The difference between them is that the device identification, function code parameters, and modifiers are optional in the command format; but the device identification is optional here, the function code is required, and modifiers are not permitted. The did parameter here is used to identify the output writer that is to receive the instructions being sent in the function code portion of the message. A function code is required because the output writer being communicated with is already loaded. The function codes that can be specified here are the same as those that can be specified in the command format, and are described in Table 5–3.

- Solicited Messages

```
Øi function-code
```

where:

Øi

Is the job and message number used to reply directly to an output writer in the solicited mode of transmission. The number i corresponds to the number of the message that was transmitted by the output writer that requested the reply.

function-code
Is the message text that is used to instruct the requesting output writer of the function to be performed. The function codes that may be specified are the same as those described in Table 5–3.

Examples:

1. Operator keyin:

   00 PR(002) STOP

   Function requested:

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

2. System console display:

   05? ENTER OUTPUT WRITER FUNCTION

   Operator keyin:

   05 BURST

   Function requested:

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

5.5. SPOOL FILE PROCESSING AIDS

A special system service program (SL$LOG) and two canned job control streams (DUMPLOG and DUMPLOGT) allow you to dump the job log subfiles from the spool LOG file to a SYSLOG file created on disk or tape. Once in SYSLOG, the subfiles are available for processing by user programs and by a job log report program (JOBLOG) that produces a job accounting report based on the contents of the SYSLOG file.

The SL$LOG program with the DUMPLOG and DUMPLOGT job control streams also allow you to dump the retained console logs from the spool LOG file to a user tape or disk file. This accumulated console log file is available for processing by a user program once it is dumped.

The following paragraphs describe job log subfiles, how they are saved, how the job log accounting reports are produced, and how the accumulated console log is saved.

5.5.1. Saving Job Log Subfiles (SL$LOG Program)

All 90/30 and 90/40 systems that are generated with the spooling option specified (SPOOLING=INPUT, OUTPUT, or REMOTE) maintain a spool log subfile for each job processed in the system. Each log subfile contains a copy of:

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

As each job is terminated, its associated spool log subfile is closed and normally output to a high-speed printer as soon as an output writer becomes available in the system. This copy of the job log subfile (Figure 5–1) is normally given to the person that submitted the job for his debugging and bookkeeping purposes. If the system is operating with the accumulate system log files option in effect (the system was generated with the SYSLOG=YES parameter specified, or the SET SPL,DUMP command was issued by the operator after the system was initialized), the job log subfile is marked as having been printed, and then saved to allow job accounting and other bookkeeping chores to be accomplished at the "installation level".

If the system is operating with the accumulate system log files option suppressed (the system was generated with the SYSLOG=NO parameter specified or the SET SPL,ENDDUMP command was issued by the operator after the system was initialized), the job log subfile is deleted from the spool log file immediately after it is processed by the output writer. The job log subfile is no longer available for any further processing.

If log subfiles are being saved, the SL$LOG program can be used to transfer them to either a user tape or disk file to make them available for further processing by user programs. The SL$LOG program allows for dumping log records only, accounting records only, or both log and accounting records. Snap and cancel dump records are ignored, as they can serve no purpose from a bookkeeping standpoint. After the transfer operation takes place, SL$LOG effectively erases the closed log subfiles from the spool LOG file, returning the file space to the system spooler for its reuse.

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

The SL$LOG program is executed using the RUN command with canned job control streams, as described in the paragraphs that follow. A typical job log subfile printout is shown in Figure 5–1.

5.5.1.1. Using Tape for the Job Log SYSLOG File (RUN DUMPLOGT)

Function:

This command establishes a SYSLOG file on tape and transfers the job log records from the system spool LOG file into this file. After the transfer operation is completed, the output tape will be rewound to its load point.

Format:

Positional Parameter 1:

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

Positional Parameter 2:

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

Keyword Parameters:

D=ACT
Indicates that accounting records only are to be transferred.

```
// LOAD CASEY,F2                                                                                                                   L 00:01:29
//PRNTR1 SPL ,,,,,P,NOHDR                                                                                                          L 00:01:32
// RUN                                                                                                                             L 00:01:33
025 TEST CARD FOR OCL PROG                                                                                                         L 00:01:33
// LOAD CASEY,F2                                                                                                                   L 00:01:34
//PUNCH SPL ,,5,,,MYFORMS                                                                                                          L 00:01:35
//PUNCH SPL HOLD                                                                                                                   L 00:01:35
// RUN                                                                                                                             L 00:01:35
055 TEST CARD FOR OCL PROG                                                                                                         L 00:01:35
// FIN                                                                                                                             L 00:01:35
AC01  JOB OCL0002    ACCT. NO.            ASSIGNED MEMORY=00008192 BYTES (PLUS 003072 BYTE PROLOGUE)     00/00/00                 A 00:01:40
JC07  USING   DEV=FFF TYPE=PRNTR     DEV=FFF TYPE=PRNTR                                                                            L 00:01:43
JC08  USING   DEV=FFF TYPE=PUNCH                                                                                                   L 00:01:43
JC01  JOB OCL0002  EXECUTING JOB STEP CASEY000 #001 00:01:43                                                                       L 00:01:43
AC10  LFD - PRNTR   , FORM NAME - STAND1   , COPIES - 001, PAGES - 00000000, STEP =001                                             A 00:01:48
AC11  STEP #001  (CASEY000) USED 00004866 BYTES    ELAPSED WALL CLOCK TIME=00:00:04.494    TOTAL SVC CALLS=00000213               A 00:01:48
AC12       TERM CODE=000       SWITCH-PRIORITY=10    CPU TIME USED          =00:00:00.610    TRANSIENT CALLS=00000010               A 00:01:48
AC19                DEVICE EXCP'S      303=00000080  PRT=00000002                                                                  A 00:01:48
JC07  USING   DEV=FFF TYPE=PRNTR     DEV=FFF TYPE=PRNTR                                                                            L 00:01:50
JC08  USING   DEV=FFF TYPE=PUNCH                                                                                                   L 00:01:51
JC01  JOB OCL0002  EXECUTING JOB STEP CASEY000 #002 00:01:51                                                                       L 00:01:51
AC10  LFD - PRNTR   , FORM NAME - STAND1   , COPIES - 001, PAGES - 00000000, STEP =002                                             A 00:01:56
AC11  STEP #002  (CASEY000) USED 00004866 BYTES    ELAPSED WALL CLOCK TIME=00:00:04.665    TOTAL SVC CALLS=00000212               A 00:01:56
AC12       TERM CODE=000       SWITCH-PRIORITY=10    CPU TIME USED          =00:00:00.593    TRANSIENT CALLS=00000010               A 00:01:56
AC19                DEVICE EXCP'S      303=00000080  PRT=00000002                                                                  A 00:01:56
JC07  USING   DEV=FFF TYPE=PRNTR     DEV=FFF TYPE=PRNTR                                                                            L 00:01:58
JC08  USING   DEV=FFF TYPE=PUNCH                                                                                                   L 00:01:59
AC21  JOB TOTALS     USED 00004866 BYTES      TOTAL ELAPSED WALL CLOCK TIME=00:00:20.605    TOTAL JOB SVC CALLS=00000425          A 00:02:01
AC22                                          WALL CLOCK TIME OF ALL STEPS =00:00:09.159    JOB TRANSIENT CALLS=00000020          A 00:02:01
AC23                            .             TOTAL CPU TIME OF ALL STEPS  =00:00:01.203    TOTAL JOB EXCP'S    =00000164          A 00:02:01
JC02  JOB OCL0002   TERMINATED NORMALLY              00:02:01                                                                      L 00:02:01
```

Figure 5—1. Typical Job Log Subfile Printout

D=LOG
Indicates that log records only are to be transferred.

If the D keyword parameter is omitted, both log and accounting records will be transferred.

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

V=SYSLOG
Identifies SYSLOG as the volume serial number to be used.

C=Y
Indicates that checkpoint records are desired.

C=N
Indicates that no checkpoint records are desired.

5.5.1.2. Using Disk for the Job Log SYSLOG File (RUN DUMPLOG)

Function:

This command establishes a SYSLOG file on disk and transfers the job log records from the system spool LOG file into this file.

Format:

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

Positional Parameter 1:

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

Positional Parameter 2:

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

Keyword Parameters:

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

D=ACT
Indicates that accounting records only are to be transferred.

D=LOG
Indicates that log records only are to be transferred.

If the D keyword parameter is omitted, both log and accounting records will be transferred.

5.5.1.3. Adding Job Log Subfiles to the SYSLOG File on Disk (RUN DUMPLOG)

Function:

This command adds additional job log subfiles to an existing SYSLOG file.

Format:

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

Positional Parameter 1:

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

Positional Parameter 2:

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

Keyword Parameters:

D=ACT
Indicates that additional accounting records only are to be transferred.

D=LOG
Indicates that additional log records only are to be transferred.

If the D keyword parameter is omitted, additional log and accounting records will both be transferred.

5.5.1.4. Reinitializing the Job Log SYSLOG File on Disk (RUN DUMPLOG)

Function:

This command reinitializes (effectively erases) the SYSLOG file and transfers new job log subfiles from the spool LOG file to the SYSLOG file. Thus, the current contents of the SYSLOG file are overwritten with new spool LOG file job log records.

Format:

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

Positional Parameter 1:

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

Positional Parameter 2:

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

Keyword Parameters:

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

D=ACT
Indicates that accounting records only are to be transferred.

D=LOG
Indicates that log records only are to be transferred.

If the D keyword parameter is omitted, both log and accounting records will be transferred.

5.5.2. Producing Job Accounting Reports (JOBLOG Program)

Once your job log SYSLOG file has been created by SL$LOG, you can execute JOBLOG to produce a job accounting report. JOBLOG uses the OS/3 independent sort/merge routine to sort the SYSLOG file. You can execute JOBLOG using the RUN command with a canned job control stream.

5.5.2.1. Running JOBLOG Using Tape Input (RUN JBLOGT)

Function:

This command sorts tape file input to produce a job accounting report.

Format:

```
RUN JBLOGT[,,V={vol-ser-no}  ,S={A}]
              {SYSLOG    }     {C}
```

Positional Parameter 1:

JBLOGT
Indicates that the job accounting file input to JOBLOG will be on tape.

Positional Parameter 2:

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

Keyword Parameters:

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

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

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

S=B

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

S=C

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

Example:

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

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

5.5.2.2. Running JOBLOG Using Disk Input (RUN JBLOG)

Function:

This command sorts disk file input to produce a job accounting report.

Format:

```
RUN JBLOG[,,V={vol-ser-no|RES},L={vol-ser-no|SYSLOG},S={A|B|C}]
```

Positional Parameter 1:

JBLOG

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

Positional Parameter 2:

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

Keyword Parameters:

V=vsn

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

V=RES

Identifies SYSRES as the volume containing the job accounting file.

L=vsn

Identifies the file name of the job accounting file.

L=SYSLOG

Identifies SYSLOG as the name of the job accounting file.

S=A

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

S=B

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

S=C

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

Example:

The job accounting file is named SYSLOG and is residing on the SYSRES volume. You need the report by accounting number and job name. You use the following command:

```
RUN JBLOG
```

5.5.3. Saving the Accumulated Console Logs (SL$LOG Program)

All 90/30 and 90/40 operating systems that are generated with spooling and with the console log file SYSGEN parameter (CONSOLOG) specified maintain a record of all messages written to and read from the console. This console log file is part of the spool LOG file and can be accessed by the operator and by the SL$LOG program in a manner similar to that for the job log files. Console logs are accumulated when either the SYSGEN parameter RETAINLOG has been specified or when the operator enters the SET SPL,CNSLG command with the RETAIN parameter included. You can also use the SET SPL,CNSLG command to turn the console log file on or off and to print or not print the console logs. When the console logs have been accumulated, the SL$LOG program allows you to dump them to a user tape or disk file. This output is used as input to a user-written program.

Usually you will execute SL$LOG to dump the job log file so it can be used as input to the JOBLOG program. However, when you also want to dump the accumulated console logs, perform the following steps in the sequence shown to ensure the integrity of the output.

1. Execute the SL$LOG program to dump the job log records.
2. Execute the JOBLOG program to produce a job accounting report.
3. Execute the SL$LOG program to dump the accumulated console logs.
4. Execute a user-written program to process the console log output.

The SL$LOG program is executed by using the RUN command with canned job control streams, as described in the paragraphs that follow.

5.5.3.1. Using Tape for the Console Log SYSLOG File (RUN DUMPLOGT)

Function:

This command establishes a SYSLOG file on tape and transfers the accumulated console log records from the system spool LOG file into this file. After the transfer operation is completed, the output tape will be rewound to its load point.

Format:

```
RUN DUMPLOGT,,D=CON[,V={vsn   }][,C={Y}]
                      {SYSLOG}     {N}
```

Positional Parameter 1:

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

Positional Parameter 2:

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

Keyword Parameters:

D=CON
Indicates that the accumulated console log records only are to be transferred.

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

V=SYSLOG
Identifies SYSLOG as the volume serial number to be used.

C=Y
Indicates that checkpoint records are desired.

C=N
Indicates that no checkpoint records are desired.

5.5.3.2. Using Disk for the Console Log SYSLOG File (RUN DUMPLOG)

Function:

This command establishes a SYSLOG file on disk and transfers the accumulated console log records from the system spool LOG file into this file.

Format:

```
RUN DUMPLOG,,F=ALLOC,D=CON
```

Positional Parameter 1:

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

Positional Parameter 2:

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

Keyword Parameters:

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

D=CON
Indicates that the accumulated console log records only are to be tranferred.

5.5.3.3. Adding Console Log Subfiles to the SYSLOG File on Disk (RUN DUMPLOG)

Function:

This command adds additional console log subfiles to an existing SYSLOG file.

Format:

```
RUN DUMPLOG,,D=CON
```

Positional Parameter 1:

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

Positional Parameter 2:

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

Keyword Parameter:

D=CON
Indicates that additional console log records only are to be added.

5.5.3.4. Reinitializing the Console Log SYSLOG File on Disk (RUN DUMPLOG)

Function:

This command reinitializes (effectively erases) the SYSLOG file and transfers a new accumulated console log file from the spool LOG file to the SYSLOG file. Thus, the current contents of the SYSLOG file are overwritten with new spool LOG file console log records.

Format:

```
RUN DUMPLOG,,F=INIT,D=CON
```

Positional Parameter 1:

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

Positional Parameter 2:

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

Keyword Parameters:

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

D=CON
Indicates that the accumulated console log records only are to be transferred.

7. System Utility Services

7.1. GENERAL

The system utility symbiont (SL$$SU) is a multipurpose utility that allows you to perform many different functions using cards, tapes, disks, or diskettes. Table 7–1 breaks down the different functions to the media associated with them.

Table 7—1. System Utility Functions (Part 1 of 2)

Function Code	Function Performed
	Card Functions
CC	Reproducing cards punched in Hollerith code
CC96	Reproducing 96-column cards
CCB	Reproducing cards punched in binary and Hollerith code
CCS	Reproducing and resequencing source programs
CS96	Reproducing and resequencing source programs contained on 96-column cards
CT	Writing card to tape in unblocked format
CT96	Writing 96-column cards to tape in unblocked format
CTR	Writing card to tape in blocked format
CP	Listing cards
CP96	Listing 96-column cards in character format
CH	Listing cards containing compressed mode
CH96	Listing 96-column cards in vertical hexadecimal format
JCP	Punching cards from the system console

Table 7—1. System Utility Functions (Part 2 of 2)

Function Code	Function Performed
Tape Functions	
TT	Copying a tape to another tape
TH	Printing a tape in character and hexadecimal format
THR	Printing a tape in character, hexadecimal, deblocked format
TP	Printing a tape containing only standard characters
TPR	Printing a tape in character and deblocked format
TRS	Locating a specific record on tape
TRL	Changing existing records on tape
TC	Punching cards from tape
INT	Prepping a tape
FSF	Forward space to a specific file
BSF	Backward space to a specific file
FSR	Forward space to a specific record
BSR	Backward space to a specific record
WTM	Writing tape marks
REW	Rewinding a tape
RUN	Rewinding a tape with interlock
ERG	Erasing a portion of a tape
Disk Functions	
DD	Printing a disk in unblocked format
SD	Printing a disk in split cylinder mode
DDR	Printing a disk in reblocked format
SDR	Printing a disk in split cylinder/deblocked format
VTP	Printing the volume table of contents of a disk
DID	Changing volume serial number (VSN) of a disk
Diskette Functions	
DD	Printing a diskette in unblocked format
VTP	Printing the data set labels of a diskette
DID	Changing volume serial number (VSN) of a diskette

NOTE:

If XXX is entered in place of the function code, all function codes are displayed.

7.2. SYSTEM UTILITY COMMANDS AND MESSAGES

Function:

The system utility symbiont is loaded by the SU/TU command to the system. The SU and TU symbionts can be used interchangeably for all functions. However, it is recommended that the TU symbiont be used for tape operations since TU increases the buffer size for all selector channel tapes from 8189 to 32,767 bytes.

The function required can be included as a parameter with the SU/TU command. A spooling parameter can also be entered with the command, if spooling is configured in your system. When the command alone is entered and the symbiont is initiated, the function required is entered as a solicited message. In either case, any additional control of the symbiont is entered as a solicited message. Unsolicited messages are used only to terminate the symbiont or current symbiont functions on certain occasions.

Format:

```
{SU}  [function-code[,{Y}]]
{TU}                 {N}
```

Positional Parameter 1:

`function-code`

Specifies the appropriate 2- or 3-character function code. (Function codes are shown in Table 7–1.) If omitted, the symbiont displays a message requesting that you enter a function.

All possible function codes that are recognized by either symbiont can be displayed on the system console by entering XXX in place of the function code. Following this display, the symbiont requests that you enter the required function code.

Positional Parameter 2:

Y

Specifies that the system utility output is to be spooled.

N

Specifies that the system utility output is not to be spooled.

This parameter is entered only if spooling is configured in your system. When Y is specified or taken as the default condition, the output writer automatically prints or punches any spooled output at the end of every SU function.

When the symbiont is loaded, the following message is displayed:

0i SYSTEM UTILITY SYMBIONT LOADED

where:

i

Is a 1-digit hexadecimal message number (1–F) consecutively assigned to output messages generated by the supervisor.

If the command entry includes a function code, the symbiont completes the requested function, then displays an ENTER REQUIRED FUNCTION message to allow you to either terminate the symbiont or request another function.

If the function code is omitted, the symbiont displays the ENTER REQUIRED FUNCTION message to allow you to enter a function.

The ENTER REQUIRED FUNCTION message is displayed as follows:

- With spooling:

  0i? ENTER REQUIRED FUNCTION AND SPOOL OPTION [,Y/N] DEFAULT=Y

- Without spooling:

  0i? ENTER REQUIRED FUNCTION

The required function is initiated by the following keyin:

- With spooling:

  0i function-code, {Y / N}

- Without spooling:

  0i function-code

Each time the symbiont completes a requested function, it transmits the ENTER REQUIRED FUNCTION message to allow you to either terminate the symbiont or request another function.

If your system supports spooling, and the spool option is incorrectly entered, (i.e., a character other than Y or N is entered), the following message is displayed:

0i? IS {PRINTED / PUNCHED} OUTPUT TO BE SPOOLED FROM SU Y/N

Enter the required spooling option with the keyin:

0i {Y / N}

To terminate the symbiont, you reply to the ENTER REQUIRED FUNCTION with the end-of-job keyin:

0i EOJ

The symbiont then terminates and displays the message:

0i SYSTEM UTILITY SYMBIONT ENDED

An unsolicited entry can also be made to end the symbiont by pressing the MESSAGE WAITING key on the system console and keying in one of the following:

00 SU EOJ
00 TU EOJ

This permits you to terminate the symbiont before it has completed a function.

To terminate only the current function of the symbiont, key in one of the following unsolicited entries:

00 SU END
00 TU END

NOTE:

When message replies are keyed in incorrectly or the reply cannot be honored, the symbiont requests the information to be keyed in again. If no determination can be made on why the keyed input is not accepted, use the unsolicited entry (above) to terminate the current function or to terminate the symbiont.

7.3. CARD FUNCTIONS

All the card functions that can be performed are described in this section. All input card files must be terminated by a card with the words END OF DATA punched in columns 1 through 11.

Following are the procedures to perform the card functions.

1. As described in 7.2:

   a. Enter the SU symbiont command.

   b. Enter the appropriate function code either as a command parameter or as a solicited message response to the ENTER REQUIRED FUNCTION message.

   c. Enter the spooling option; otherwise, default is Y (applicable only if spooling is configured).

2. If a card file is being read, place it in the card reader designated the system reader (SYSRDR). If this card reader is unavailable, the first available card reader is assigned to the symbiont, causing the following message to be displayed on the system console:

   0i USE READER did

   where:

   did
   Is the address of the card reader assigned to read the input file.

   If no card readers are available, the function is aborted and the following message is displayed on the system console:

   0i NO READER AVAILABLE

   Likewise, if the required output device is not available, the function is aborted and the following message is displayed on the system console:

   0i NO { PUNCH / TAPE / PRINTER } AVAILABLE

If the required devices are available, the operation will continue for each function code, as described in the following paragraphs.

7.3.1. Reproducing Cards Punched in Hollerith Code (CC)

You use the CC function code to reproduce cards in 80 x 80 format containing the Hollerith code. All job control cards, even the /*, can be reproduced by using this function. You must submit an END OF DATA card with your input deck indicating the end of file to the symbiont.

7.3.2. Reproducing 96-Column Cards (CC96)

You use the CC96 function code to read 96-column cards and punch 80-column cards. Columns 81-96 of the input cards are truncated. The input card file must be terminted with an END OF DATA card.

7.3.3. Reproducing Cards Containing Binary Data (CCB)

You use the CCB function to reproduce cards containing binary data in addition to the Hollerith code. Again, you must submit an END OF DATA card as the last card in your input deck. When punching column binary, the output must not be spooled.

7.3.4. Reproducing and Resequencing Source Programs (CCS)

You use the CCS function code to reproduce and resequence an assembler (BAL), COBOL, or RPG source language program. For a BAL program, the program name can be up to three characters in length; COBOL can be up to eight characters in length; and RPG can be up to six characters in length. If you supply a name less than the amount of characters permitted, the name is left-justified and space-filled. You must submit an END OF DATA card as the last card of your source program. Tables 7-2 through 7-4 show the formats of the source programs being reproduced and resequenced.

Table 7—2. Assembler Format

Column	Description
1-72	Source statement
73-75	3-character program name
76-80	5-character sequence number; first number is 00010. Succeeding numbers are incremented by 10.

Table 7—3. COBOL Format

Column	Description
1-3	Page sequence number; starts at 001 and is incremented by every 20 lines (cards)
4-6	Line sequence number; starts at 010 and is incremented by 10 for every line up to 200
7-72	Source statement
73-80	8-character program name

Table 7—4. RPG Format

Column	Description
1–2	Page number sequence; starts at 01
3–5	Line number sequence; starts at 010 and is incremented by 10 for every line up to 200
6–74	Source statement
75–80	6-character program name

7.3.5. Reproducing and Resequencing Source Programs Contained on 96-Column Cards (CS96)

You use this function code to reproduce and resequence your source program when it is contained on 96-column cards. CS96 reads 96-column cards and punches and sequences 80-column cards. Columns 81–96 of the input cards are truncated. Sequencing is the same as that described for the CCS function code. An END OF DATA card must follow the source program to indicate end-of-file.

Procedure:

For the message:

0i? XXX DECK TYPE

Key in:

0i BAL
If the source program is in assembly language.

0i COB
If the source program is in COBOL.

0i RPG
If the source program is in RPG.

SU will print the message:

0i DECK NAME n CHARACTERS

Followed by the message:

0i? XXX

where:

n

Is the number of characters printed for the source deck program and is the number of X's that appears in the second message. The permissible numbers of characters for a source deck name are:

- 8 characters for a COBOL source deck;
- 6 characters for an RPG source deck; and
- 3 characters for a BAL source deck.

Key in the correct characters that are used as the program (deck) name for output cards. If less than the specified number of characters is provided, the data given is left-justified and space-filled to the right.

7.3.6. Writing Cards to Tape in Unblocked Format (CT)

You use the CT function code to write cards to an unblocked tape in 80 x 80 format. You must submit an END OF DATA card in your input deck indicating the end of file whereby two tape marks are then written on your tape.

Procedure:

Copies a deck of cards onto an unlabeled output tape in 80 x 80 format. No leading tape marks are written, but two tape marks are written when the END OF DATA card is read. Each data block is 80 bytes long. You specify the density and mode in which the tape is written at execution time. The tape is not rewound at either the beginning or the end of this function.

For the message:

0i? CUUMMB OUTPUT TAPE B=BLK CNT

You respond with the device, mode setting, and block count characteristics of the tape output devices (as described in 7.4.1).

7.3.7. Writing 96-Column Cards to Tape in Unblocked Format (CT96)

You use the CT96 function code to copy 96-column cards to tape in 96 x 96 format. An END OF DATA card must be in your input deck indicating the end of file whereby two tape marks are then written on your tape.

Procedure:

Copies a deck of cards onto an unlabeled output tape in 96 x 96 format. No leading tape marks are written but two tape marks are written when the END OF DATA card is read. Each data block is 96 bytes long. You specify the density and mode in which the tape is written at execution time. The tape is not rewound at either the beginning or the end of this function.

For the message:

0i? CUUMMB OUTPUT TAPE B=BLK CNT

You respond with the device, mode setting, and block count characteristics of the tape output devices (as described in 7.4.1).

7.3.8. Writing Cards to Tape in Blocked Format (CTR)

You use the CTR function code to write cards to a tape in blocked format. The blocking factor is in the range of 1 to 100.

Procedure:

Same as the CT function except tape blocks written are 80 x f long, where f is the blocking factor (maximum blocking factor is 100). If the last block is not full, the length of the last block is 80 x r, where r is the number of records in the last block.

For the message:

0i? BLK FACTOR

You respond with blocking factor 1 to 100.

For the message:

0i? CUUMM OUTPUT TAPE B=BLK CNT

You respond with the device and mode setting of the tape output device (as described in 7.4.1).

For the message:

0i? OUTPUT EXCEEDS ALLOCATED BUFFER

You must reenter the function and blocking factor because the factor supplied was greater than 100 and the function was terminated.

7.3.9. Listing Cards (CP)

You use the CP function code to list cards in 80 x 80 format on the printer. You must submit an END OF DATA card as the last card in your input. Only printable punch configurations are printed; however, any standard punch configuration is accepted. The function is useful when listing your job control cards.

7.3.10. Listing 96-Column Cards in Character Format (CP96)

You use the CP96 function code to list your 96-column cards in character format on the printer. The processing constraints for this function are the same as those described for the CP function code.

7.3.11. Listing Cards Containing Compressed Mode (CH)

You use the CH function code to list cards containing the compress mode (hexadecimal characters) and the standard characters. Again, you must have an END OF DATA card as the last card in your input deck.

7.3.12. Listing 96-Column Cards in Vertical Hexadecimal Format (CH96)

You use the CH96 function code to list your 96-column cards in character and vertical hexadecimal format. An END OF DATA card is required to terminate the function.

7.3.13. Punching Cards from the System Console (JCP)

Punches job control or data cards entered through the system console. Up to 60 columns may be entered in reply to the scale message. If more than 60 columns are needed, position the cursor under the 0 of 60 in the scale message and transmit. Another message requesting 20 more characters will appear. If a card having a blank in column 1 is required, key in a right parenthesis instead of a blank for column 1. The symbiont replaces a right parenthesis in column 1 with a blank. The right parenthesis is needed because all messages received by SU are returned left-justified. To terminate the JCP function, key in END and transmit immediately.

7.4. TAPE FUNCTIONS

The tape functions that can be performed by the symbiont are described below. It is recommended you use the TU symbiont because of the increase in buffer size for selector channel tapes.

7.4.1. Tape Addressing

The tapes to be used for a TU function must be identified using the tape unit ID. Communication by you is initiated by the following message:

```
cuummb
```

where:

cuu
Is device ID (channel and unit).

mm
Is mode setting of the tape. If mm is blank, 00, or not entered (e.g., response is 102), the SYSGEN mode settings are assumed. (Refer to OS/3 job control user guide, UP-8065 (current version).

b
Is block count characteristics of the tape. If blank or omitted, the tape is assumed to not have a block count. If b is entered, the tape is assumed to have a block count.

NOTES:

1. *The block count specification is not necessary for tape functions used to position a tape (e.g., FSF).*
2. *The record number (REC) printed by the system utility for the TRS function is relative to the beginning of scan. For other tape functions, it is relative to where printing begins. The REC number does not correspond to the 3-byte block number on block numbered tapes.*
3. *If the TU symbiont is specified, the maximum block size for selector channel tape units is increased from 8189 to 32,767 bytes. The block size for multiplexer tape units remains at 8189 bytes.*

7.4.2. Tape Error Processing

If an error is encountered on an input tape, control is turned over to a tape error correction routine, where communication is established with you to determine whether the error should be ignored or bypassed or the function can be terminated.

If an error is ignored, the record is processed as is. If an error is bypassed, the input tape is reread before returning to the active function; therefore, no processing is performed on the error block.

The messages:

0i TAPE ERROR ON INPUT TAPE

0i? B-BYPASS, I-IGNORE, OR E-END FUNCTION

inform you that a tape error has occurred and one of the following options may be replied:

0i BYPASS (This reply is not applicable during a TRL function.)

0i IGNORE

0i END

7.4.3. Tape Operating Instructions

To perform a tape function, proceed as follows:

1. As described in 7.2:
   a. Enter the TU symbiont command.
   b. Enter the appropriate function code either as a command parameter or as a solicited message response to the ENTER REQUIRED FUNCTION message.
   c. Enter the spooling option; otherwise, default is Y (applicable if spooling if configured).
2. Place the input tape volume on an available tape unit, and identify the tape unit to the TU symbiont by keying in its unit address as described in 7.4.1.

If the required devices are available, the operation will continue for each function code, as described in the following paragraphs.

7.4.3.1. Copying a Tape to Another Tape (TT)

You use the TT function code to copy from one tape to another tape of the same or different device type. You can copy either labeled or unlabeled tapes in blocked or unblocked format. If you are copying standard label tapes, the file marks are used as controlling devices. The first file mark on the tape includes the header record; the second file mark includes all of your data; and the third file mark includes the trailer record. Therefore, three file marks make up one complete file.

Procedure:

For the messages:

0i? CUUMMB-INPUT TAPE B=BLK CNT

Respond with the tape unit ID, mode setting, and block count characteristics of the input tape (refer to 7.4.1).

0i? CUUMMB-OUTPUT TAPE B=BLK CNT

Respond with the tape unit ID, mode setting, and block count characteristics of the output tape.

0i? #FILES 1-99

Respond with the number of files or file marks to be copied. The number must be from 1 through 99. See previous description of TT for number of files on a standard labeled tape.

0i? END OF VOLUME? Y OR N

This message is issued after all files specified have been copied. Key in Y if a second tape mark is to be written after the file, or N if it is not to be written. After processing this response, the function is completed.

An example of tape copy with block count might be as follows:

System message:

0i? CUUMMB - INPUT TAPE

Operator response:

0i 100△△B

System message:

0i? CUUMMB - OUTPUT TAPE

Operator response:

0i 101△△B

System message:

0i? #FILES 1-99

Operator response:

0i 3

System message:

0i? END OF VOLUME? Y/N

Operator response:

0i Y

7.4.3.2. Printing a Tape in Either Character or Hexadecimal Format (TH)

You use the TH function code to print a tape containing either standard or compressed mode (hexadecimal) characters. The tape error correction routine is enabled to allow you to bypass or ignore tape errors with this function. Tape positioning does not occur at either the beginning or the ending of this function. The block number printed on the printer is relative to the location where printing begins.

Procedure:

An option permits you to print the entire field or only a specified number of blocks:

0i? CUUMMB - INPUT TAPE B=BLK CNT

Respond by entering the tape unit ID, mode set, and block count characteristics of the input tape (refer to 7.4.1). If a block count is entered, data is considered to begin in position 3, relative to position 0, for a length of blocksize minus 3 bytes.

0i? #BLKS OR END

Respond END if the entire tape is to be printed, or respond with the number of blocks to be printed. If the end-of-file code (that is, two tape marks side by side) is detected before the specified number of blocks is printed, the function is terminated.

7.4.3.3. Printing a Tape in Either Character or Hexadecimal Deblocked Format (THR)

You use the THR function code to print records from a tape individually rather than in a contiguous string. Prints logical tape records in character and vertical hexadecimal formats. The tape error correction routine is enabled to permit you to correct, bypass, or ignore tape errors for this function.

Procedure:

An option allows you to print the entire file or only a specified number of blocks:

0i? CUUMMB - INPUT TAPE B=BLK CNT

Enter the tape unit ID, mode set, and block count characteristic of the input tape (refer to 7.4.1).

0i? #BLKS OR END

If the entire tape is to be printed, enter END or the number of blocks to be printed. If the end of file (two tape marks side by side) is detected before the specified number of blocks are read, the function is terminated.

0i? LOGICAL REC LNTH

Respond with the logical record length.

7.4.3.4. Printing a Tape Containing Only Standard Characters (TP)

You use the TP function code to print a tape in blocked format containing only standard characters. This function is identical to the TH function, with the exception that your output is only in character format rather than character and hexadecimal format.

7.4.3.5. Printing a Tape in Character and Deblocked Format (TPR)

You use the TPR function code to print records from a tape individually in character format only. This function is identical to the THR function, with the exception that your output is in character format rather than character and hexadecimal format.

7.4.3.6. Locating a Specific Record (TRS)

You use the TRS function code to search for one or more specific logical records in your file. The following information is needed for the symbiont to locate the record:

- logical record length;
- length scan argument (length of the data field);
- starting data position in your record;
- whether the data is in character or hexadecimal format; and
- actual data needed.

Procedure:

The scan field is identified by the operator with the length and location within the logical record. The logical record length is also required for deblocking purposes. The scan argument is entered through the console, in either hexadecimal or character format, and compared to the scan field of each logical tape record. The scan argument is printed on the printer for future reference. When an equal is located, the block having the record is printed in character and hexadecimal format. After each "hit", you are given the option of continuing the scan for more "hits" or terminating the function. Tape positioning does not occur at either the beginning or the end of this function. The tape error correction route is enabled, so you can ignore or bypass tape errors (refer to 7.4.2). The function is terminated either at the end-of-file (two tape marks side by side) or when a hit is made and you terminate the function. The scale is printed to improve readability. You also enter the scan argument for use in the search.

For the messages:

0i INVALID HEX CHAR

An error message is printed when the scan argument is entered in hexadecimal and an invalid hexadecimal character was detected. You reenter the sequence to obtain the scan argument.

0i NO MATCH FND

This message is printed when an end-of-file mark (two tape marks back-to-back) is encountered before any hits were made. The function is terminated.

0i? CONTINUE SCAN? Y/N

This message is printed each time a hit is made. You enter Y if you wish to continue the scan, or N to terminate the scan.

An example of a tape with block count that is to be searched for a tape record containing a key field of "4637275467" in the first 10 bytes of the record is as follows:

System message:

0i? CUUMMB - INPUT TAPE

Operator response:

0i 10000B

System message:

0i? LOGICAL REC LENGTH

Operator response:

0i 125

System message:

0i? LENGTH ARGUMENT (1-30)

Operator response:

0i 10

System message:

0i? STARTING DATA POSITION IN REC

Operator response:

0i 1

System message:

0i ENTER IN HEX-H, CHAR-C

Operator response:

0i C

System messages:

0i ENTER 10 BYTES, 1 CHAR PER BYTE
0i? 1...5...10

Operator response:

0i 4637275467

System message:

0i? CONTINUE SCAN? Y/N

Operator response:

0i N

7.4.3.7. Changing Existing Records (TRL)

You use the TRL function code to change an existing block in your tape file. You can change either character or hexadecimal data. In order for the symbiont to change your block, the following information is needed:

- relative block number;
- record number;
- data to be changed; and
- position where the change is to take place.

Procedure:

This is a tape copy that allows you to alter the contents or length of error or nonerror tape blocks. Tape positioning does not occur at the beginning or end of this function. The tape error correction routine allows you to ignore or bypass tape errors. You locate the desired record by entering the direction (forward or backward) and number of blocks the tape must move to reach the record. As tape is positioned, the output tape is written from the input tape if direction is forward, and backspaced if direction is backward. When the record is located, it is printed, and you have the opportunity to confirm that this is the desired record. When all changes are complete, the tape is copied onto the output tape from its current position; therefore, blocks should be changed sequentially (e.g., if records 5, 6, 9 are changed, the changes should be on record 5 first; then 6 and 9). When locating a record, if a tape mark is encountered, you must determine to continue or terminate the function. If you continue, reenter the direction and number of blocks to the desired record.

For messages:

0i? CUUMMB - INPUT TAPE B=BLK CNT

Enter the tape unit ID, mode setting, and block count characteristics of the input tape (refer to 7.4.1).

0i CUUMMB - OUTPUT TAPE B=BLK CNT

Enter the tape unit ID, mode setting, and block count characteristic of the output tape (refer to 7.4.1).

0i? #BLKS OR END

If all records are changed, key in END. If more changes are required, enter the number of blocks (relative to current tape position) to the desired record. The maximum number of records is 9999.

0i? COPY FWD OR BACK? F OR B

Issued if the response to the last message is not END. Respond to this message with F if tape is to move forward to the desired record, or B if tape is to move backward.

0i? DESIRED REC

Indicates the desired record has been reached and printed. Respond Y if this is the desired record to initiate the change. Respond N if this is not the desired record, so as to reenter the record locating sequence.

0i TAPE MARK, CONTINUE? Y/N

Indicates a tape mark is encountered before the desired record is reached. Enter N to terminate the function, or Y to reenter the tape record locating sequence.

0i? CHANGES TO LENGTH? Y/N

Enter N if the block size is not to be changed, or Y if the block size is to change.

0i? DESIRED REC LNTH

Indicates a change is pending on the tape block length. Enter the new block length. If the new length is shorter than the old length, the block is truncated on the right. If the new block is longer, the block is extended on the right by spaces.

0i INPUT EXCEEDS ALLOCATED BUFFER

Indicates a change made to the record length caused the record length to exceed maximum.

0i? #BYTES TO BE CHANGED, (1-30)

Indicates a change is to be made to data. Enter the number of bytes (1–30) required for change during the sequence.

0i? STARTING DATA POSITION IN REC

Enter the first byte position (relative to 1) to be changed.

0i LENGTH EXCEEDS END OF RECORD

Indicates the number of bytes to change plus the starting data position minus 1 exceeds the length of the active record. The sequence to locate the change field is reentered.

0i? ALTER IN HEX-H, CHAR-C

Enter H if new data to be entered is in hexadecimal, or enter C if new data is in character format.

0i ENTER XX BYTES, Y CHAR PER BYTE

0i? 1...5...10...

where:

XX
Is number of bytes.

Y
Is 1 if alter is in character format; 2 if alter is in hexadecimal format.

Respond with new data. The scale is used to improve readability.

0i INVALID HEX CHAR

An invalid hexadecimal character is detected when the alter data is entered in hexadecimal. An error message is printed. The sequence to obtain new data is reentered.

0i CHANGES COMPLETE? Y/N

Enter Y if changes to the current block are complete and to initiate the new record locating sequence. Enter N to reenter the change sequence for this record if more changes are required.

Example 1:

Data record 5 is to be changed to hexadecimal 003C in positions 6–7. The tape has standard labels.

System message:

0i? CUUMMB - INPUT TAPE B=BLK CNT

Operator response:

0i 10000B

System message:

0i? CUUMMB - OUTPUT TAPE B=BLK CNT

Operator response:

0i 10100B

System message:

0i? #BLKS OR END

Operator response:

0i 3 (to copy standard label)

System message:

0i? COPY FWD OR BACK? F OR B

Operator response:

0i F

System message:

0i? TAPE MARK, CONTINUE? Y OR N

Operator response:

0i Y

System message:

0i? #BLKS OR END

Operator response:

0i 5

System message:

0i? COPY FWD OR BACK? F OR B

Operator response:

0i F

System message:

0i? DESIRED REC? Y OR N

Operator response:

0i Y

System message:

0i? CHANGES TO LENGTH? Y OR N

Operator response:

0i N

System message:

0i #BYTES TO BE CHANGED, (1-30)

Operator response:

0i 2

System message:

0i? STARTING DATA POSITION IN REC

Operator response:

0i 6

System message:

0i? ALTER IN HEX-H, CHAR-C

Operator response:

0i H

System messages:

0i ENTER 02 BYTES, 2 CHAR PER BYTE

0i? 1...

Operator response:

0i 003C

System message:

0i? CHANGES COMPLETE? Y OR N

Operator response:

0i Y

System message:

0i? #BLKS OR END

Operator response:

0i END

Example 2:

The header and trailer records on a tape with a block count must be changed so the label is:

{ BLOCK COUNT HDR1 LAB1 }
{ BLOCK COUNT EOF1 LAB1 }

The first byte of the label is 5 and the length of the new label is 4. The tape contains more than 9999 records.

System message:

0i? CUUMMB - INPUT TAPE

Operator response:

0i 10000B

System message:

0i? CUUMMB - OUTPUT TAPE

Operator response:

0i 10100B

System message:

0i? #BLKS OR END

Operator response:

0i 2

System message:

0i? COPY FWD OR BACK? F OR B

Operator response:

0i F

System message:

0i? DESIRED REC? Y OR N

Operator response:

0i Y

System message:

0i? CHANGES TO LENGTH? Y OR N

Operator response:

0i N

System message:

0i? #BYTES TO BE CHANGED, (1-30)

Operator response:

0i 4

System message:

0i? STARTING DATA POSITION IN REC

Operator response:

0i 5

System message:

0i? ALTER IN HEX-H, CHAR-C

Operator response:

0i C

System messages:

0i ENTER 4 BYTES, 1 CHAR PER BYTE

0i? 1...

Operator response:

0i LAB1

System message:

0i? CHANGES COMPLETE? Y OR N

Operator response:

0i Y

System message:

0i? #BLKS OR END

Operator response:

0i 9999

System message:

0i? COPY FWD OR BACK? F OR B

Operator response:

0i F

System message:

0i? DESIRED REC? Y OR N

Operator response:

0i N

System message:

0i #BLKS OR END

Operator response:

0i 9999

System message:

0i? COPY FWD OR BACK? F OR B

Operator response:

0i F

System message:

0i? TAPE MARK, CONTINUE? Y/N

Operator response:

0i Y

System message:

0i? #BLKS OR END

Operator response:

0i 1

System message:

0i? COPY FWD OR BACK? F OR B

Operator response:

0i F

System message:

0i? DESIRED REC? Y OR N

Operator response:

0i Y

System message:

0i? CHANGES TO LENGTH? Y OR N

Operator response:

0i N

System message:

0i #BYTES TO BE CHANGED, (1-30)

Operator response:

0i 4

System message:

0i? STARTING DATA POSITION IN REC

Operator response:

0i 8

System message:

0i? ALTER IN HEX-H, CHAR-C

Operator response:

0i C

System messages:

0i ENTER 4 BYTES, 1 CHAR PER BYTE

0i? 1...

Operator response:

0i LAB1

System message:

0i? CHANGES COMPLETE? Y OR N

Operator response:

0i Y

System message:

0i? #BLKS OR END

Operator response:

0i END

7.4.3.8. Punching Cards from a Tape (TC)

You use the TC function code to punch cards from either a blocked or unblocked tape. The data portion of the tape is transferred 80 bytes at a time to the card. If the data portion of the block is not a multiple of 80, then the remaining data is left-justified and space-filled. When a tape mark is detected, a card containing all asterisks (*) is punched. When two tape marks are detected back-to-back, the function ends.

Procedure:

When a tape mark is encountered, a card with all * is punched. The function is terminated when two tape marks back-to-back are encountered. Tape error processing is described in 7.4.2.

For the message:

0i? CUUMMB - INPUT TAPE B=BLK CNT

Respond with the device ID, mode setting, and block characteristics of the input tape (as described in 7.4.1).

7.4.3.9. Prepping a Tape (INT)

You use the INT function code to prep a tape by writing a standard label on a specified tape. Your volume-serial-number and file label are entered through the system console. The tape is positioned to the load point prior to the prep.

Procedure:

For messages:

0i? CUUMMB - OUTPUT TAPE B=BLK CNT

Respond with the tape unit ID, mode setting, and block count characteristic of the output tape (refer to 7.4.1).

0i ENTER NEW VOL#

0i? XXXXXX

Enter up to six digits for VOL1 record. If you enter fewer than six digits, the new volume serial number is left-justified and space is filled on the right.

0i ENTER NEW FILE LABEL

0i? XXXXXXXXXXXXXXXXX

Enter up to 17 characters for a new file identifier.

Example:

System message:

0i? CUUMMB - OUTPUT TAPE

Operator response:

0i 10100B

System messages:

0i ENTER NEW VOL#

0i? XXXXXX

Operator response:

0i 001036

System messages:

0i ENTER NEW FILE LABEL

0i? XXXXXXXXXXXXXXXXX

Operator response:

0i TAPE1

7.4.3.10. Forward Space to a Specific File (FSF)

You use the FSF function code to cause the tape to be advanced to the next tape mark.

Procedure:

Advances the specified tape to the next tape mark.

For the message:

0i? CUUMM - OUTPUT TAPE

Enter the tape unit ID and mode setting of the tape to be spaced forward (refer to 7.4.1).

7.4.3.11. Backward Space to a Specific File (BSF)

You use the BSF function code to cause the tape to be backspaced until a tape mark is encountered.

Procedure:

For the message:

0i? CUUMM - OUTPUT TAPE

Enter the tape unit ID and mode setting of the tape to be backspaced (refer to 7.4.1).

7.4.3.12. Forward Space to a Specific Record (FSR)

You use the FSR function code to forward space a specific number of blocks on your tape. You enter the number of blocks to be forward spaced through the system console.

Procedure:

For the message:

0i? CUUMM - OUTPUT TAPE

Enter the tape unit ID and mode setting of the tape to be forward spaced (refer to 7.4.1).

0i? #BLKS

Enter the number of blocks to be advanced (maximum number is 9999).

7.4.3.13. Backward Space to a Specific Record (BSR)

You use the BSR function code to backspace a specific number of blocks on your tape. You enter the number of blocks to be backspaced through the system console.

Procedure:

For the message:

0i? CUUMM - OUTPUT TAPE

Enter the tape unit ID and mode setting of the tape to be backspaced (refer to 7.4.1).

0i? #BLKS

Enter the number of blocks to be backspaced (maximum number of blocks is 9999).

7.4.3.14. Writing Tape Marks (WTM)

You use the WTM function code to write tapemarks on your file.

Procedure:

For the message:

0i? CUUMM - OUTPUT TAPE

Key in the tape unit ID and mode setting of the output tape (refer to 7.4.1).

7.4.3.15. Rewind Tape (REW)

You use the REW function code to rewind your tape to load point.

Procedure:

For the message:

0i? CUUMM - OUTPUT TAPE

Key in the tape unit ID and mode setting of the tape to be rewound (refer to 7.4.1).

7.4.3.16. Rewind and Unload Tape (RUN)

You use the RUN function to rewind either UNISERVO 16 or UNISERVO 20 tapes to load point with interlock.

Procedure:

For the message:

0i? CUUMM - OUTPUT TAPE

Key in the tape unit ID and mode setting of the tape to be rewound (refer to 7.4.1).

7.4.3.17. Erasing Tape Record Gap (ERG)

You use the ERG function code to erase a portion of your tape. This function is useful to erase known defective areas on your tape. Erases the specific tape for approximately 3.5 inches.

Procedure:

For the message:

0i? CUUMM - OUTPUT TAPE

Key in the tape unit ID and mode setting of the tape to be erased (refer to 7.4.1).

7.5. DISK FUNCTIONS

All the disk functions that can be performed are described in the following paragraphs.

7.5.1. Operating Considerations

The following considerations should be remembered when operating with the system utility on disks:

- When performing a disk prep or disk copy, mount the SYSRES disk pack on a disk drive having an address number lower than the disk drive on which the pack for prepping or copying is mounted.
- Disk packs used on IDA disk subsystems have been subjected to extensive surface analysis. Defective tracks found as a result of this analysis are listed in a defective track table location on the bottom of the plastic disk pack storage cover on the label containing the pack serial number.
- An end-of-file record is a disk record on which data length is in binary zeros (not applicable on IDA disk subsystems).
- Split cylinder mode is a method of processing sequential cylinders in which heads are selected sequentially within restricted head bounds.

  For example:

  START: 00000 (CYL 000 HD 00)

  END: 00903 (CYL 009 HD 03)

  As cylinders 000 through 009 are selected, only heads 00 through 03 of each cylinder are selected.

7.5.2. Disk Operating Instructions

To request one of these functions, proceed as follows:

1. Place the subject disk volume on an available disk unit.
2. As described in 7.2:
   a. Enter the SU symbiont command.
   b. Enter the appropriate function code either as a command parameter or as a solicited message response to the ENTER REQUIRED FUNCTION message.
   c. Enter the spooling option; otherwise, default is Y (applicable if spooling is configured).

7.5.2.1. Printing a Disk in Unblocked Format (DD)

You use the DD function code to print your disk pack in character and hexadecimal format. The DD function does not deblock your logical records.

Procedure:

For the messages:

0i? ENTER DVC ADDRESS

Enter the unit ID of the disk drive containing the disk pack to be displayed.

0i? CCCHH - BEGIN OR FILE-ID

Enter in decimal the beginning cylinder (CCC) and head (HH) to be displayed, or the file identifier (up to 44 characters) as used on the // LBL job control statement when the file was created. If less than 44 characters are entered, the file ID is padded with blanks on the right. Printing begins at the start of the file when a file is entered.

0i? CCCHH - END

Enter the last cylinder (CCC) and head (HH) in decimal to be displayed.

0i? ENTER NO TRACKS TO BE PRINTED (1-9)

If you had entered the file ID, now enter the number of tracks to be printed. Note that printing begins with the low cylinder and head numbers of the first extend and continues for the requested number of tracks. Printing is not confined to the extents specified in the format label.

Example 1:

Cylinder 3 head 4 through cylinder 6 head 6 on device 300 are to be printed.

System message:

0i? ENTER DVC ADDRESS

Operator response:

0i 300

System message:

0i? CCHHH - BEGIN OR FILE-ID

Operator response:

0i 00304

System message:

0i? CCCHH END

Operator response:

0i 00606

Example 2:

The first three tracks from a file called SEQUENTIAL DISC on device 440 are to be printed.

System message:

0i? ENTER DVC ADDRESS

Operator response:

0i 440

System message:

0i? CCCHH - BEGIN OR FILE-ID

Operator response:

0i SEQUENTIAL DISC

System message:

0i? ENTER NO OF TRACKS TO BE PRINTED

Operator response:

0i 3

7.5.2.2. Printing a Disk in Split Cylinder Mode (SD)

You use the SD function code to print your disk pack in split cylinder mode. This is a method of processing sequential cylinders in which the heads are selected sequentially within restricted head bounds. In other words, you can select a cylinder without accessing all the heads on that cylinder. This function, as in the previous function, does not deblock your logical records.

Prints the specified limits of a disk in character and hexadecimal formats in split cylinder mode. The function is the same as the DD function except this function operates in split cylinder mode (refer to the consideration in split cylinders).

7.5.2.3. Printing a Disk in Deblocked Format (DDR)

You use the DDR function code to print your disk pack in deblock format in both character and hexadecimal formats. This function is similar to the DD function, with the only exception that your logical records are deblocked. Disk limits are keyed in at the system console.

Procedure:

For the messages:

0i? ENTER DVC ADDRESS

Enter the unit ID of the disk drive containing the disk pack to be displayed.

0i? CCCHH - BEGIN

Enter in decimal the first cylinder (CCC) and head (HH) to be printed.

0i? CCCHH - END

Enter in decimal the last cylinder (CCC) and head (HH) to be printed.

0i? RECORD SIZE

Enter in decimal the logical record size.

0i? BLOCK SIZE

Enter in decimal the logical block size. However, if the block size is not an exact multiple of the record size, reenter the record size/block size sequence.

7.5.2.4. Printing a Disk in Split Cylinder Deblocked Format (SDR)

You use the SDR function code to print your disk pack in both character and hexadecimal formats in split cylinder mode. This function is similar to the SD function, with the only exception that your logical records are deblocked. Disk limits are keyed in at the system console.

7.5.2.5. Printing the Volume Table of Contents (VTP)

You use the VTP function code to get a copy of your VTOC. You have the option of telling the operator that you need one of the following:

- a full VTOC listing, giving you all the device information plus extents and other information for all your allocated files;
- device information only, giving you the available space left on your volume and other information regarding your volume; or
- file information only, giving you the extent and other information regarding the file.

Edits and prints the volume table of contents (VTOC) information for the requested volume. VTP will not process other than OS/3-created VTOCs. The DD function should be used to print a VTOC from a non-OS/3-created disk.

Procedure:

Three list options are available, as follows:

- VSN, DI – Device Information Only

  This option lists the free extents that are available and other information on the requested volume.

- VSN, FILE ID. – File Information

  This option lists device information plus the extent and other information on the requested file.

- VSN, ALL – Full VTOC Listing

  This option lists device information plus the extent and other information for all files allocated on the requested file.

For the message:

0i? ENTER DVC/VSN, DI, ALL, FILE ID, OR END

Key in one of the following, where vsn is the volume serial number of the disk pack to be printed.

- For full VTOC listing:

  0i vsn, ALL

- For a listing of only the device information:

  0i vsn, DI

- For a listing of up to 44 characters as used on the // LBL card when the file was created:

  0i vsn, FILE ID

- To terminate the VTP function:

  0i vsn, END

NOTE:

The device address of the disk pack may replace the VSN in any of the preceding messages.

7.5.2.6. Changing the Disk Volume Serial Number (DID)

You use the DID function code to change the VSN of your disk. The DID function will only change the VSN if the device is not allocated to any other job in the system, including SYSRES, SYSRUN, and SYSPOOL.

Procedure:

For the messages:

0i? ENTER DEVICE ADDRESS

Enter the unit ID of the disk drive containing the disk pack VSN to be changed.

0i? ENTER OLD VSN OF DISK

Enter the old VSN to be changed (up to six characters). The old VSN entered must match the VSN in the disk VOL label; otherwise, the function terminates with an error message.

0i? ENTER NEW VSN OF DISK

Enter the new VSN (up to six characters).

7.6. DISKETTE FUNCTIONS

7.6.1. Diskette Operating Instructions

To request one of these functions, proceed as follows:

1. Place the subject diskette volume on an available diskette unit.
2. As described in 7.2:
   a. Enter the SU symbiont command.
   b. Enter the appropriate function code either as a command parameter or as a solicited message response to the ENTER REQUIRED FUNCTION message.
   c. Enter the spooling option; otherwise, default is Y (applicable if spooling is configured).

7.6.1.1. Printing a Diskette in Unblocked Format (DD)

You use the DD function code to print your diskette in character and hexadecimal format. The DD function does not deblock your logical records.

Procedure:

For the messages:

0i? ENTER DVC ADDRESS

Enter the unit ID of the diskette drive containing the diskette to be displayed.

0i? TTRR-BEGIN OR FILE-ID

Enter in decimal the beginning track (TT) and sector (RR) to be displayed, or the file identifier (up to 44 characters) as used on the // LBL job control statement when the file was created. If fewer than 44 characters are entered, the file ID is padded with blanks on the right. Printing begins at the start of the file when a file ID is entered.

0i? TTRR-END

Enter the last track (TT) and sector (RR) in decimal to be displayed.

7.6.1.2. Printing the Data Set Labels (VTP)

You use the VTP function code to get a copy of your data set labels. These labels are similar in function to a VTOC. You have the option of telling the operator that you need one of the following:

- a full data set label listing, giving you all the device information plus extents and other information for all your allocated files;
- device information only, giving you defective track information, owner-id, number of recording surfaces, and the physical sector length; or
- file information only, giving you the extent and other information regarding the file.

The VTP function code edits and prints the data set label information for the requested volume.

Procedure:

Three list options are available, as follows:

- VSN, DI – Device Information Only

  This option lists the free extents that are available and other information on the requested volume.

- VSN, FILE ID. – File Information

  This option lists device information plus the extent and other information on the requested file.

- VSN, ALL – Full Data Set Label Listing

  This option lists device information plus the extent and other information for all files allocated on the requested file.

For the message:

0i? ENTER DVC/VSN, DI, ALL, FILE-ID, OR END

Key in one of the following, where vsn is the volume serial number of the diskette to be printed:

- For a full data set label listing:

  0i vsn, ALL

- For a listing of only the device information:

  0i vsn, DI

- For a listing of up to 44 characters as used on the // LBL card when the file was created:

  0i vsn, FILE-ID

- To terminate the VTP function:

  0i vsn, END

NOTE:

The device address of the diskette may replace the VSN in any of the preceding messages.

7.6.1.3. Changing the Diskette Volume Serial Number (DID)

You use the DID function code to change the VSN of your diskette. The DID function will only change the VSN if the device is not allocated to any other job in the system, including SYSRES, SYSRUN, and SYSPOOL.

Procedure:

For the messages:

0i? ENTER DEVICE ADDRESS

Enter the unit ID of the diskette drive containing the diskette VSN to be changed.

0i? ENTER OLD VSN OF DISK

Enter the old VSN to be changed (up to six characters). The old VSN entered must match the VSN in the diskette VOL label; otherwise, the function terminates with an error message.

0i? ENTER NEW VSN OF DISK

Enter the new VSN (up to six characters).

Appendix A. Message and Command Format Conventions

The conventions used to illustrate the messages and commands presented in this manual are:

- Informaticn that must be keyed in exactly as shown in information displayed by OS/3 is presented in uppercase letters. For example, a message output to the system console by the supervisor is illustrated as follows:

  ▷ 01 EARLY WARNING OVERTEMP. CONDITION EXISTS

- Lowercase letters represent variable information that is either displayed or keyed in. For example, the following command format implies that the command DELETE must be followed by the name of the job to be deleted.

  ```
  DELETE jobname
  ```

- The first two letters of each command are underlined to remind you that they are the only letters required to be keyed in to initiate processing of the command and its associated symbiont. For example, only the letters DE need be keyed in to initiate processing of the DELETE command and subsequent running of the delete symbiont. Its format is thus presented as:

  ```
  DELETE jobname
  ```

- Braces { } illustrate alternate choices. For example, the format of the change command

  ```
  CHANGE jobname,{PRE }
                 {HIGH}
                 {NOR }
  ```

  indicates that PRE (P), HIGH (H), or NOR (N) may be keyed in after the jobname.

- Brackets [ ] denote optional entries.

  For example, the format of the FILE command

  ```
  FILE [{(did)          }]
       [{([did],label)}]
       [{(RDR,label)  }]
  ```

  indicates that the FILE command can be keyed in by itself, or with a parameter as specified in the format.

- Default parameters are shaded. For example, the DISPLAY command format

specifies that the JBQ parameter must be included. If parameters 2 and 3 are not specified, the NOR scheduling priority queue with both LOCAL and REMOTE jobs is displayed.

NOTE:

Not all optional parameters have a default specification.

- An ellipsis (three periods) indicates the omission of a variable number of entries.

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

Appendix B. Operating Procedures for the 9200/9300 Series Subsystem

B.1. GENERAL

This appendix outlines the procedures required for operating the SPERRY UNIVAC 9200/9300 Series Subsystem (Figure B–1) online with the SPERRY UNIVAC 90/30, 90/30 B, or 90/40 Systems under control of Operating System/3 (OS/3). For complete details about the 9200/9300 system, see the 9200/9300 series processor and storage operator reference, UP-7781 (current version).

The 9200/9300 subsystem is connected to the 90/30, 90/30 B, or 90/40 system by means of a 9000 channel adapter attached to multiplexer channels on both units. When operated online with the 90/30, 90/30 B, or 90/40 system, the 9200/9300 acts as an I/O controller through which the integrated printer, card reader, and card punch are used as 90/30, 90/30 B, or 90/40 I/O devices. Other 9200/9300 devices and the read/punch feature on the integrated card punch are not supported. To perform the function of an I/O controller, the 9200/9300 uses a highly modified MOS supervisor, which is provided as a deck of punched cards.

OS/3 interfaces with the 9200/9300 I/O devices through a device handler that is configured at system generation time if the keyword parameter TYPE=9200 or TYPE=9300 is specified for any printer, reader, or punch category. OS/3 supports each I/O device independently, regardless of the status or condition of the other 9200/9300 devices.

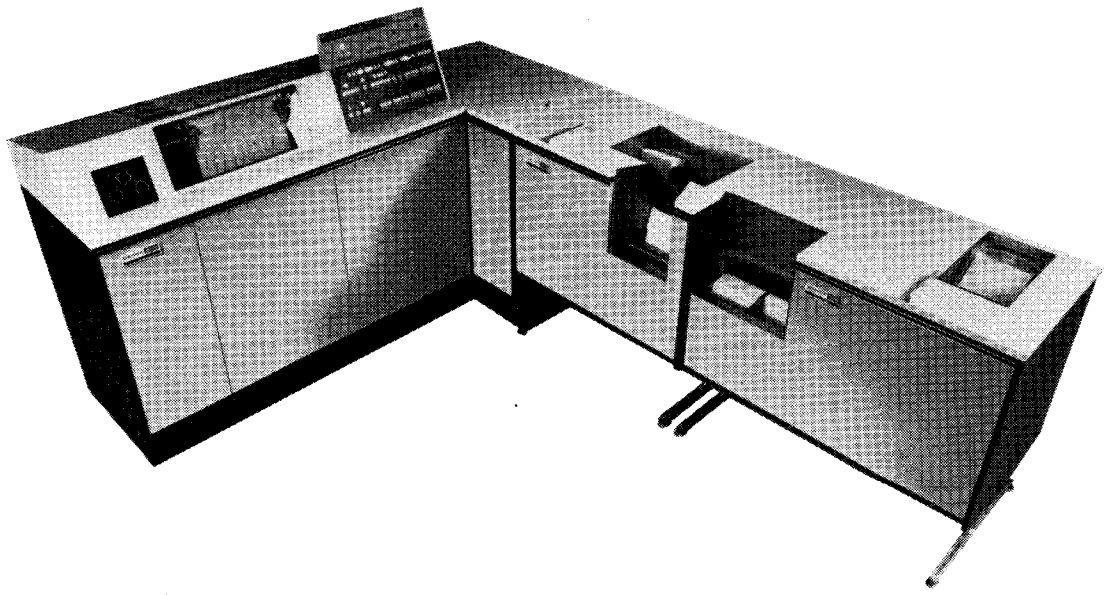


Figure B—1. 9200/9300 Series Subsystem

B.2. OPERATING CONTROLS AND INDICATORS

The control console (Figure B–2) contains the controls and indicators required to:

- perform an initial program load of the 9200/9300 subsystem;
- initialize and reset the printer, reader, and punch;
- produce a main storage dump; and
- receive and respond to error messages.

Controls on the control console are of two types: momentary and 2-position. Momentary controls may be pushbuttons or switches. Momentary switches are pressed on the upper portion; when released, they return to the normal position. Two-position switches are set to either of two positions by pressing the upper or lower portion.

Data is entered by setting the DATA ENTRY switches to represent hexadecimal values. Storage locations are indicated by setting the MEMORY ADDRESS switches. In both cases, the upper portion of each switch is depressed to designate a bit value of 1; the lower portion, to designate a bit value of 0.

Error messages are displayed in hexadecimal code on the NEXT INSTRUCTION/HALT DISPLAY indicators. A lighted indicator designates a bit value of 1; an unlighted indicator, a bit value of 0. Error messages relating to the 9200/9300 devices are also received on the 90/30, 90/30 B, or 90/40 system console.

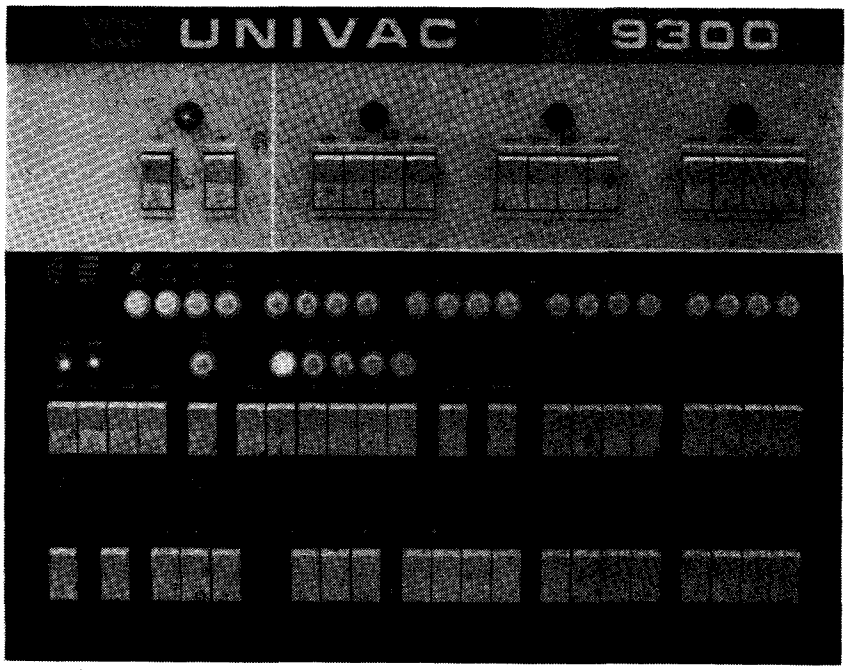


Figure B—2. Control Console, Controls and Indicators

B.3. INITIAL PROGRAM LOAD PROCEDURE

The initial program load for the 9200/9300 supervisor may be performed either before or after the IPL for the 90/30, 90/30 B, or 90/40 system (3.3). The procedure is as follows:

1. Place program deck in the reader input hopper.
2. Press READER CLEAR and READER FEED switches.
3. Set DATA ENTRY switches to X'01'.
4. Press CLEAR switch.
5. Press LOAD switch.
6. Press START switch.
7. Reset LOAD switch.
8. Press START switch.

B.4. INITIALIZATION AND RESET PROCEDURES

Initialization and reset procedures are performed by means of the OPERATOR REQUEST function and the low-order (right position) DATA ENTRY switches. To perform each procedure, set the indicated hexadecimal value in the DATA ENTRY switches and press the OP REQ pushbutton. The high-order DATA ENTRY switches have no significance.

B.4.1. Initializing the Printer

Before attempting to operate the printer, be certain that the printer form control tape meets the specifications of the vertical format buffer being used. Procedures for preparing the printer form control tape are detailed in the 9200/9300 series processor and storage manual, UP-7781 (current version).

To initialize the vertical format buffer, enter the value X'03' and press the OP REQ pushbutton. This sets the VFB to the home paper position but does *not* move the forms.

To initialize the VFB and move the forms to the home paper position at the same time, enter the value X'0B' and press the OP REQ pushbutton.

B.4.2. Initializing the Card Reader

To initialize the card reader, enter the value X'09' and press the OP REQ pushbutton. Two cards are read and any existing errors or images in the card reader buffers are discarded. Be sure to place at least two blank cards at the end of each job deck.

Although it is required only that a card be in the card reader wait station before attempting to read cards, it is advisable to reinitialize the card reader between jobs to ensure that residual information is not inadvertently carried over from job to job.

When this procedure is performed, the channel adapter is also initialized (B.4.3).

B.4.3. Initializing the Channel Adapter

To initialize the channel adapter, enter the value X'0C' and press the OP REQ pushbutton. An "ack" message is transmitted to OS/3, containing the current status of all the 9200/9300 I/O devices. This can be helpful when misoperation has stalled the system.

B.4.4. Obtaining a Storage Dump and Reinitializing the System

When the 9200/9300 subsystem has failed as a result of stalling, power failure, or other reason, the following procedure may be performed to obtain a storage dump and/or restart the input/output operation.

1. Do either of the following:

   a. Enter the value X'0F' and press the OP REQ pushbutton; or

   b. Depress the CLEAR and RUN controls.

   The NEXT INSTRUCTION/HALT DISPLAY indicators display the value X'1FFF'.

2. Set the low-order MEMORY ADDRESS switches to location 4 (X'4').

3. Key in a value in the DATA ENTRY switches and depress the RUN button:

   a. If 0 is entered, the system will proceed from where it was prior to the interruption.

   b. If 1 is entered, a storage dump is printed. At the conclusion of the dump, the HALT DISPLAY of '1FFF' is repeated, and one of the three keyins must be made again.

   c. If any value other than 0 or 1 is entered, the system is reinitialized. It goes to an idle loop and is in the same condition as after a successful IPL.

B.5. RECOVERY PROCEDURES

Error conditions are communicated to the operator through messages on the 90/30, 90/30 B, or 90/40 console or through halt displays on the 9200/9300 control panel.

B.5.1. 90/30, 90/30 B, and 90/40 Console Messages

All console messages pertaining to the 9200/9300 devices are in the standard device, status, sense format. There are three basic messages:

```
ji? DEVICE=14x STATUS=00FF SENSE=0000 9300-NAK RU*C

ji? DEVICE=14x STATUS=00FF SENSE=0000 9300-CLEAR RU*C

ji? DEVICE=14x STATUS=0200 SENSE=ss00 9300-6xss RU*C
```

where:

j
: Is the job number.

i
: Is the message number.

x

Is the device number:

1 = card reader

2 = card punch

3 = printer

ss

Is the device status.

RU*C

Indicates the allowable operator responses:

R = retry

U = unrecoverable – return control to issuing program

C = cancel user program

The first message (9300-NAK) indicates that an I/O order was accepted by the channel adapter but has not been executed within 20 seconds thereafter. Probably the 9200/9300 subsystem is stopped. If so, press RUN on the 9200/9300 and respond R on the console.

The second message (9300-CLEAR) indicates that a retry on message 1 has also timed out. If the 9200/9300 is running, but idle, try reinitializing (B.4.4) and respond R to this message.

The third message (9300-6xss) indicates that one of the 9200/9300 devices has developed an unrecoverable error. Correct the condition as described in Table B–1 and respond R, U, or C to the message.

Table B—1. 90/30, 90/30 B, and 90/40 Console Messages for the 9200/9300 Series Subsystem (Part 1 of 2)

Error Code	Module	Condition	Operator Action
6108	Reader	Multistrobe check error	Place the last card in the output stacker and the card in the wait station on the bottom of the input deck. Feed one card; then press READER CLEAR.
6140	Reader	Hopper is empty or stacker is full.	Correct the condition and press READER CLEAR.
6140	Reader	Misfeed	If there is a card in the wait station, place it on the bottom of the input deck; feed a card and press READER CLEAR.
6180	Reader	Card jam or photocell check	See operator action for 6108 error code.
6202	Punch	Hopper is empty or stacker is full.	Correct the condition and press PUNCH CLEAR.
6220	Punch	Punch check error	Press PUNCH CLEAR.
6280	Punch	Interlock check, misfeed, stacker jam, punch entry, or exit check	Correct the condition and press PUNCH CLEAR.
6301	Printer	Low paper supply	Correct the condition.

Table B—1. 90/30, 90/30 B, and 90/40 Console Messages for the 9200/9300 Series Subsystem (Part 2 of 2)

Error Code	Module	Condition	Operator Action
6308	Printer	Wrong print bar setting	Insert the correct bar or reset the bar switch.
6320	Printer	Storage overload	No action required.
6340	Printer	Skip code cannot be found on the paper loop.	Install the correct paper loop and press PRINTER CLEAR.
6340	Printer	Skip code cannot be found in the VFB.	No recovery is possible. Either the VFB has been destroyed or an incorrect skip has been issued.
6380	Printer	Abnormal condition on the printer	Correct the condition and press PRINTER CLEAR. An extra line may print or a print line may be missing.

B.5.2. 9200/9300 Halt Displays

Error conditions may be indicated by hexadecimal displays on the NEXT INSTRUCTION/HALT DISPLAY indicators on the 9200/9300 control panel. Halt displays, their causes, and recovery procedures are listed in Table B–2.

Table B—2. 9200/9300 Control Panel Halt Displays

Hexadecimal Display	Module	Cause	Operator Action
03ss	Printer	Error during VFB initialization and "home paper" (OP REQ X'B')	Follow procedure in Table B—1 for comparable 63ss message.
12FF	Channel adapter	Illogical command sequence between 90/30 and 9200/9300	Press CLEAR and RUN to dump storage.
12ss	Channel adapter	Error on I/O command to the channel adapter	Press RUN to retry. Press CLEAR and RUN to reinitialize or dump storage.
1FFF	Storage dump	See B.4.4.	See B.4.4.
4300	Loader	Card count discrepancy	Repeat IPL procedure.
6100	Loader	Hole count check	Repeat IPL procedure.
6100	Reader	Illogical sequence	Press CLEAR and RUN to dump storage.
61ss	Loader	Card reader error	Follow procedure in Table B—1 and press RUN, or repeat IPL procedure.
6200	Punch	Illogical sequence	Press CLEAR and RUN to dump storage.
6300	Printer	Illogical sequence	Press CLEAR and RUN to dump storage.
63ss	Storage dump	Printer error during storage dump	Follow procedure in Table B—1.
7676	Switcher	Software switch list is full.	Press CLEAR and RUN to dump storage.

Index