SPERRY UNIVAC

PUBLICATIONS UPDATE

Operating System/3 (OS/3)

Operations Handbook

Operator Reference

(Series 90)

For System 80 see UP-8859

UP-8072 Rev. 7–B

This Library Memo announces the release and availability of Updating Package B to "SPERRY Operating System/3 (OS/3) Operations Handbook Operator Reference", UP-8072 Rev. 7.

This update for the 8.1 release documents the transient work area feature.

Copies of Updating Package B are now available for requisitioning. Either the updating package only or the complete manual with the updating package may be requisitioned by your local Sperry representative. To receive only the updating package, order UP-8072 Rev. 7–B. To receive the complete manual, order UP-8072 Rev. 7.

LIBRARY MEMO ONLY	LIBRARY MEMO AND ATTACHMENTS	THIS SHEET IS
Mailing Lists BZ, CZ and MZ	Mailing Lists A00, AA1, A13, 20, 20U, 21, 21U, 75, 75U, 76 and 76U (Package A to UP-8072 Rev. 7, 19 pages plus Memo)	Library Memo for UP-8072 Rev. 7–B RELEASE DATE: June, 1983

Horst.

SPERRY UNIVAC
COMPUTER SYSTEMS

PUBLICATIONS UPDATE

Operating System/3 (OS/3)

Operations Handbook
Operator Reference

UP–8072 Rev. 7–A

This Library Memo announces the release and availability of Updating Package A to "SPERRY UNIVAC Operating System/3 (OS/3) Operations Handbook Operator Reference", UP–8072 Rev. 7.

All changes in this update are corrections or clarifications applicable to features present in the operations handbook (90/30, 90/40) for 8.0 and prior releases.

Copies of Updating Package A are now available for requisitioning. Either the updating package only or the complete manual with the updating package may be requisitioned by your local Sperry Univac representative. To receive only the updating package, order UP–8072 Rev. 7–A. To receive the complete manual, order UP–8072 Rev. 7.

LIBRARY MEMO ONLY	LIBRARY MEMO AND ATTACHMENTS	THIS SHEET IS
Mailing Lists BZ,CZ and MZ	Mailing Lists A00,AA1,AA3,20,20U,21, 21U,75,75U,76 and 76U. (Package A to UP–8072 Rev. 7, 27 pages plus Memo)	Library Memo for UP–8072 Rev. 7–A RELEASE DATE: February, 1983

SPERRY UNIVAC
COMPUTER SYSTEMS

PUBLICATIONS REVISION

Operating System/3 (OS/3)

Operations Handbook

Operator Reference

UP-8072 Rev. 7

This Library Memo announces the release and availability of "SPERRY UNIVAC® Operating System/3 (OS/3) Operations Handbook Operator Reference", UP-8072 Rev. 7.

This revision documents the following new features for the 8.0 release:

- A new system command – SHUTDOWN – provides the operator with a method for orderly terminating system activity.
- The SWITCH command has been enhanced with an ALL parameter to allow the new priority level to be in effect for all subsequent job steps.
- The HOLD, BEGIN, DELETE, and DISPLAY console commands have two new features:

  (1) a HOST parameter, which specifies that only those queued jobs having the specified host-id will be affected; and

  (2) a DDP parameter, which specifies that only distributed data processing jobs will be affected.
- DDP activity can now be logged into the interactive services log file and be printed.
- The SI/SC commands have a new parameter allowing the user to run job control streams that have been saved in an alternate library file.

The following changes are also in effect for this release:

- All 90/30 B information has been removed from this document and placed in the 90/25 operations handbook, UP-8511.
- Support has been decommitted for split cylinder disk file allocation.

All other changes are corrections or expanded descriptions applicable to features present prior to the 8.0 release.

Destruction Notice: If you are going to OS/3 release 8.0, use this revision and destroy all previous copies. If you are not going to OS/3 release 8.0, retain the copy you are now using and store this revision for future use.

LIBRARY MEMO ONLY	LIBRARY MEMO AND ATTACHMENTS	THIS SHEET IS
Mailing Lists BZ, CZ and MZ	Mailing Lists A00, AA1, AA3, 20, 20U, 21, 21U, 75, 75U, 76 and 76U (Cover and 168 pages)	Library Memo for UP-8072 Rev. 7
		RELEASE DATE:
		October, 1982

Copies of UP-8072 Rev. 6 will be available for 6 months after the release of 8.0. Should you need additonal copies of this edition, you should order them within 90 days of the release of 8.0. When ordering the previous edition of a manual, be sure to identify the exact revision and update packages desired and indicate that they are needed to support an earlier release.

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

Operations Handbook

OS/3

Operator Reference

Environment: 90/30, 90/40 Systems

H

SPERRY UNIVAC

UP-8072 Rev. 7

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 consult the appropriate release documentation or 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 Corporation.

FASTRAND, SPERRY UNIVAC, UNISCOPE, UNISERVO, and UNIVAC are registered trademarks of the Sperry Corporation. ESCORT, MAPPER, PAGEWRITER, PIXIE, and UNIS are additional trademarks of the Sperry Corporation.

The statement below is included in this manual to comply with a Federal Communications Commission (FCC) Regulation. The FCC is an agency of the United States government; thus, the statement below applies to computing equipment installed in the United States of America. Sperry Univac is taking appropriate steps to be in compliance with FCC regulations and similar regulations of other countries.

WARNING: This equipment generates, uses, and can radiate radio frequency energy and if not installed and used in accordance with the instructions manual, may cause interference to radio communications. It has been tested and found to comply with the limits for a Class A computing device pursuant to Subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such interference when operated in a commercial environment. Operation of this equipment in a residential area is likely to cause interference, in which case the user, at his own expense, will be required to take whatever measures may be required to correct the interference.

Compliance is based upon a system configuration that includes Sperry Univac peripherals/subsystems so labeled, and cables furnished by Sperry Univac or built to Sperry Univac specifications and assembly procedures.

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: Update B – UP-8072 Rev. 7
RELEASE LEVEL: 8.1 Forward

Part/Section	Page Number	Update Level
Cover/Disclaimer		Orig.
PSS	1	B
Preface	1,2	Orig.
Contents	1	Orig.
	2,3	B
	4	Orig.
	5	A
	6	Orig.
1	1 thru 15	Orig.
2	1,2	Orig.
3	1 thru 4	Orig.
	5	A
	6 thru 9	Orig.
	10	B
4	1 thru 6	Orig.
	7,8	B
	8a	B*
	9 thru 28	Orig.
	28a	Orig.
	29 thru 51	Orig.
	52	B*
5	1 thru 6	Orig.
6	1,2	A
	3 thru 6	Orig.
7	1 thru 28	Orig.
	29,	A
	30,31	Orig.
	32	A
	33 thru 35	Orig.
	36	A
	37 thru 41	Orig.
	42	A
	42a	A
	43	Orig.
	44	A
	45	Orig.
Appendix A	1,2	Orig.
Appendix B	1 thru 7	Orig.
Appendix C	1 thru 3	Orig.
Index	1,2	Orig.
	3,4	B
	5	Orig.
Index (cont)	6	B
	7	Orig.
	8	A
	9	B
User Comment Sheet		

*New pages

All the technical changes are denoted by an arrow (⟹) in the margin. A downward pointing arrow (⇓) next to a line indicates that technical changes begin at this line and continue until an upward pointing arrow (⇑) is found. A horizontal arrow (⟹) pointing to a line indicates a technical change in only that line. A horizontal arrow located between two consecutive lines indicates technical changes in both lines or deletions.

Preface

This handbook is designed to instruct and guide the operator, at the system console, in the procedures required to operate the SPERRY UNIVAC 90/30 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.

Whereas the system operator uses the procedures and commands described in this handbook to control the entire system, many of these commands are also available to a workstation user under certain limitations. For a description and explanation of workstation operation procedures and available functions, see the workstation user guide, UP-8845 (current version).

One other document relating to the operation of the 90/30 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 or 90/40 data processing system and the appropriate responses, when necessary.

This operations handbook is organized as follows:

- Section 1. System Definition

  Briefly describes the minimum and maximum hardware configurations for the 90/30 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. Interactive Services

  Describes the operator commands and messages used to control the interactive system environment.

- Section 6. Integrated Communications Access Method (ICAM) Procedures

  Describes the ICAM communications tasks, including how to load the ICAM symbiont, change the ICAM name established during SYSGEN, run and terminate the global user service task, and use message instructions 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 or 90/40 systems.

- Appendix C. Supervisor Modification Procedure

  Describes the output messages and operator responses required to modify the supervisor during the IPL procedure.

Contents

FIGURES

TABLES

1. System Definition

1.1. GENERAL

The SPERRY UNIVAC 90/30 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 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).

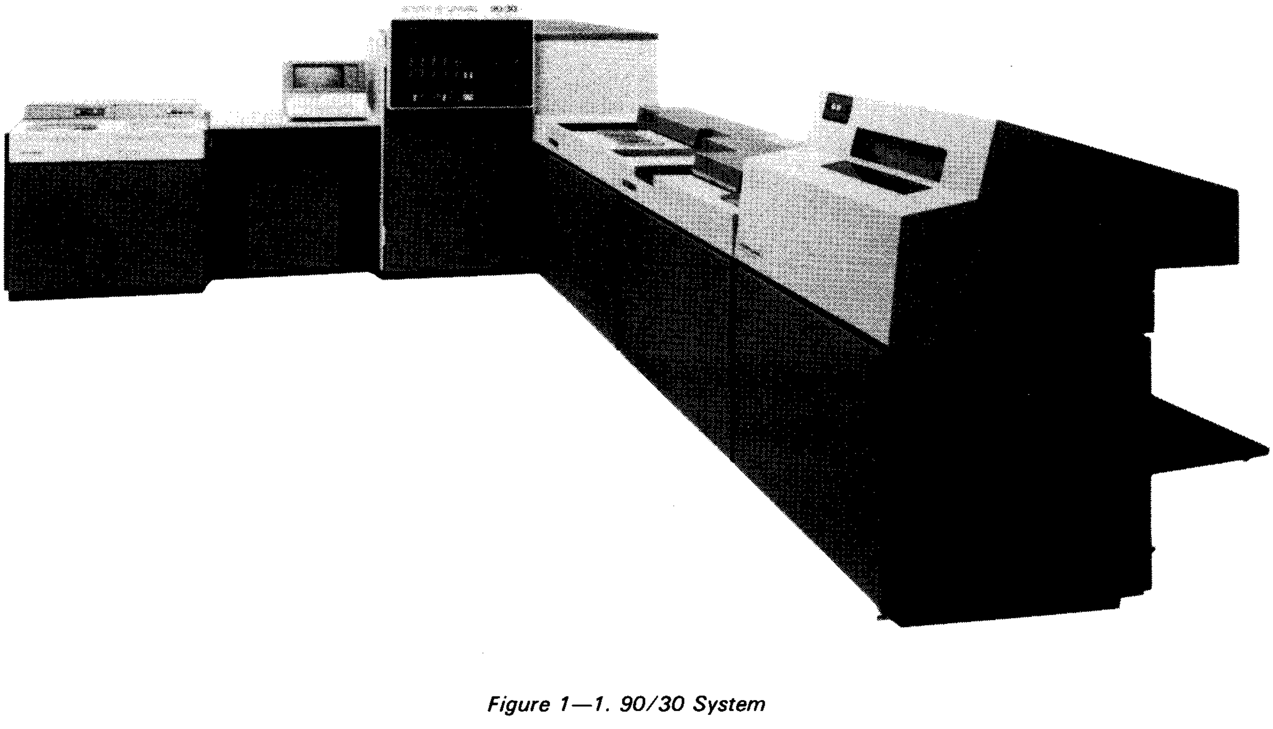


Figure 1—1. 90/30 System

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 90/40 system are presented in Figures 1-4 and 1-5, respectively.

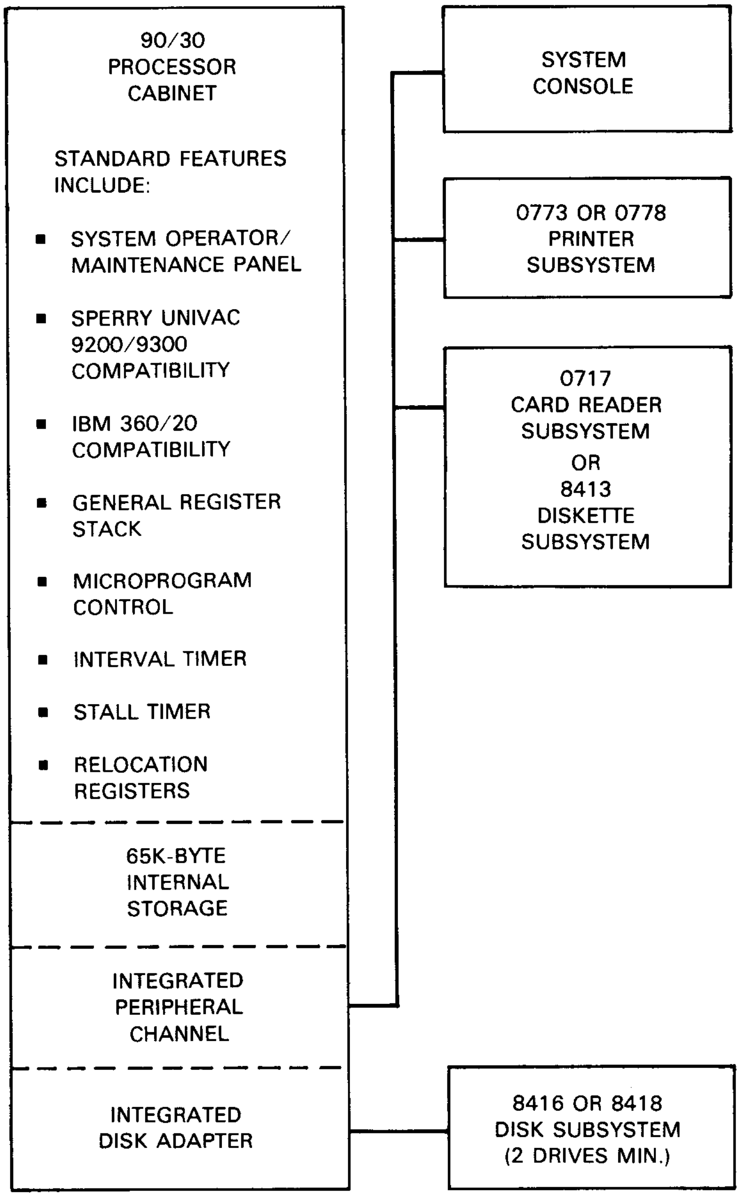
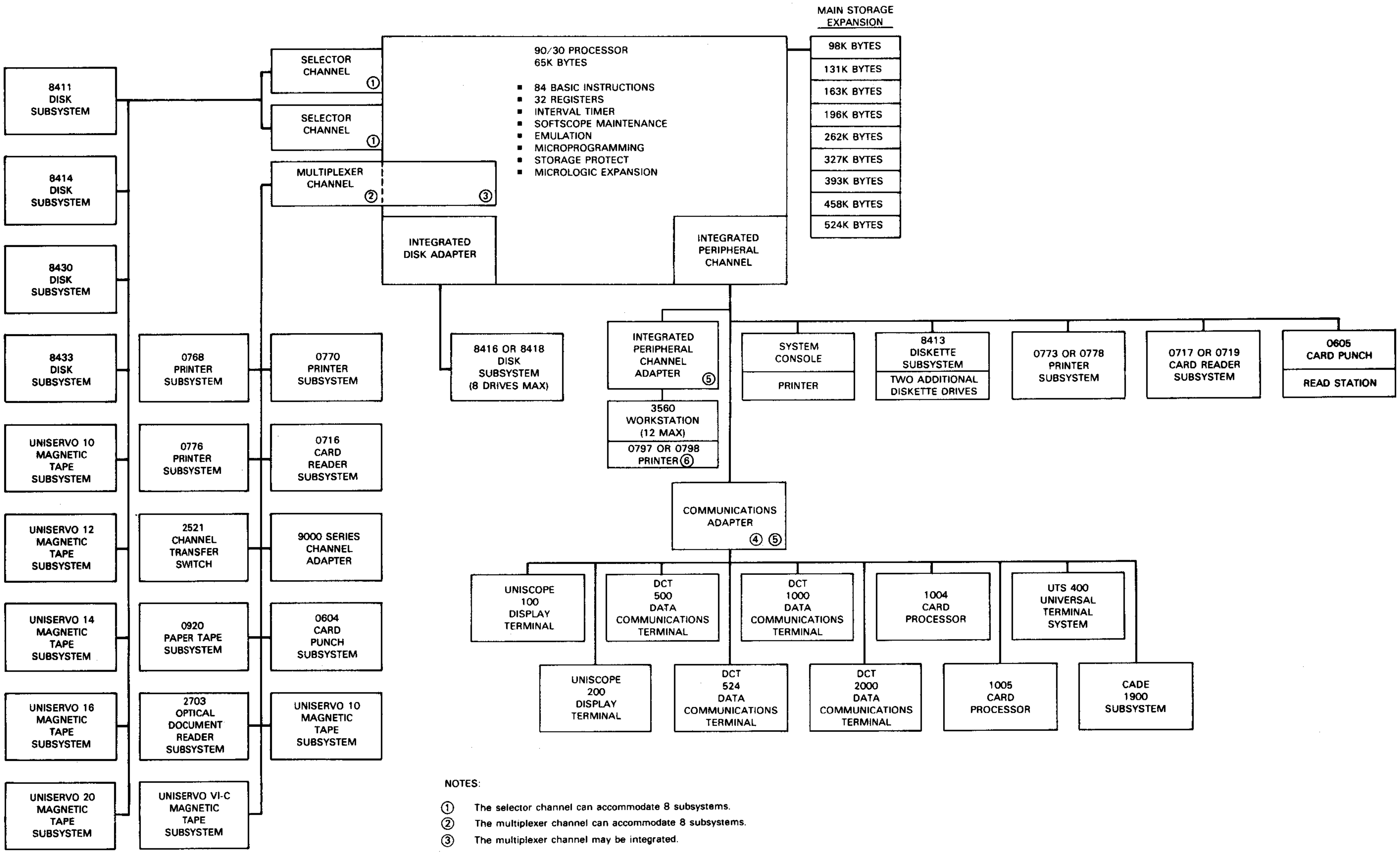

90/30
PROCESSOR
CABINET
STANDARD FEATURES
INCLUDE:
■ SYSTEM OPERATOR/ MAINTENANCE PANEL
■ SPERRY UNIVAC 9200/9300 COMPATIBILITY
■ IBM 360/20 COMPATIBILITY
■ GENERAL REGISTER STACK
■ MICROPROGRAM CONTROL
■ INTERVAL TIMER
■ STALL TIMER
■ RELOCATION REGISTERS
65K-BYTE INTERNAL STORAGE
INTEGRATED PERIPHERAL CHANNEL
INTEGRATED DISK ADAPTER
SYSTEM CONSOLE
0773 OR 0778 PRINTER SUBSYSTEM
0717 CARD READER SUBSYSTEM OR 8413 DISKETTE SUBSYSTEM
8416 OR 8418 DISK SUBSYSTEM (2 DRIVES MIN.)

Figure 1—2. Basic 90/30 System Configuration


MAIN STORAGE EXPANSION
98K BYTES
131K BYTES
163K BYTES
196K BYTES
262K BYTES
327K BYTES
393K BYTES
458K BYTES
524K BYTES
90/30 PROCESSOR
65K BYTES
■ 84 BASIC INSTRUCTIONS
■ 32 REGISTERS
■ INTERVAL TIMER
■ SOFTSCOPE MAINTENANCE
■ EMULATION
■ MICROPROGRAMMING
■ STORAGE PROTECT
■ MICROLOGIC EXPANSION
SELECTOR CHANNEL ①
SELECTOR CHANNEL ①
MULTIPLEXER CHANNEL ②
③
INTEGRATED DISK ADAPTER
INTEGRATED PERIPHERAL CHANNEL
8411 DISK SUBSYSTEM
8414 DISK SUBSYSTEM
8430 DISK SUBSYSTEM
8433 DISK SUBSYSTEM
UNISERVO 10 MAGNETIC TAPE SUBSYSTEM
UNISERVO 12 MAGNETIC TAPE SUBSYSTEM
UNISERVO 14 MAGNETIC TAPE SUBSYSTEM
UNISERVO 16 MAGNETIC TAPE SUBSYSTEM
UNISERVO 20 MAGNETIC TAPE SUBSYSTEM
0768 PRINTER SUBSYSTEM
0776 PRINTER SUBSYSTEM
2521 CHANNEL TRANSFER SWITCH
0920 PAPER TAPE SUBSYSTEM
2703 OPTICAL DOCUMENT READER SUBSYSTEM
UNISERVO VI-C MAGNETIC TAPE SUBSYSTEM
0770 PRINTER SUBSYSTEM
0716 CARD READER SUBSYSTEM
9000 SERIES CHANNEL ADAPTER
0604 CARD PUNCH SUBSYSTEM
UNISERVO 10 MAGNETIC TAPE SUBSYSTEM
8416 OR 8418 DISK SUBSYSTEM (8 DRIVES MAX)
INTEGRATED PERIPHERAL CHANNEL ADAPTER ⑤
3560 WORKSTATION (12 MAX)
0797 OR 0798 PRINTER ⑥
SYSTEM CONSOLE
PRINTER
8413 DISKETTE SUBSYSTEM
TWO ADDITIONAL DISKETTE DRIVES
0773 OR 0778 PRINTER SUBSYSTEM
0717 OR 0719 CARD READER SUBSYSTEM
0605 CARD PUNCH
READ STATION
COMMUNICATIONS ADAPTER ④ ⑤
UNISCOPE 100 DISPLAY TERMINAL
DCT 500 DATA COMMUNICATIONS TERMINAL
DCT 1000 DATA COMMUNICATIONS TERMINAL
1004 CARD PROCESSOR
UTS 400 UNIVERSAL TERMINAL SYSTEM
UNISCOPE 200 DISPLAY TERMINAL
DCT 524 DATA COMMUNICATIONS TERMINAL
DCT 2000 DATA COMMUNICATIONS TERMINAL
1005 CARD PROCESSOR
CADE 1900 SUBSYSTEM

NOTES:

① The selector channel can accommodate 8 subsystems.

② The multiplexer channel can accommodate 8 subsystems.

③ The multiplexer channel may be integrated.

④ Supports up to 12 full-duplex or 24 half-duplex lines

⑤ IPCA or second CA can be supported.

⑥ One printer per workstation

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

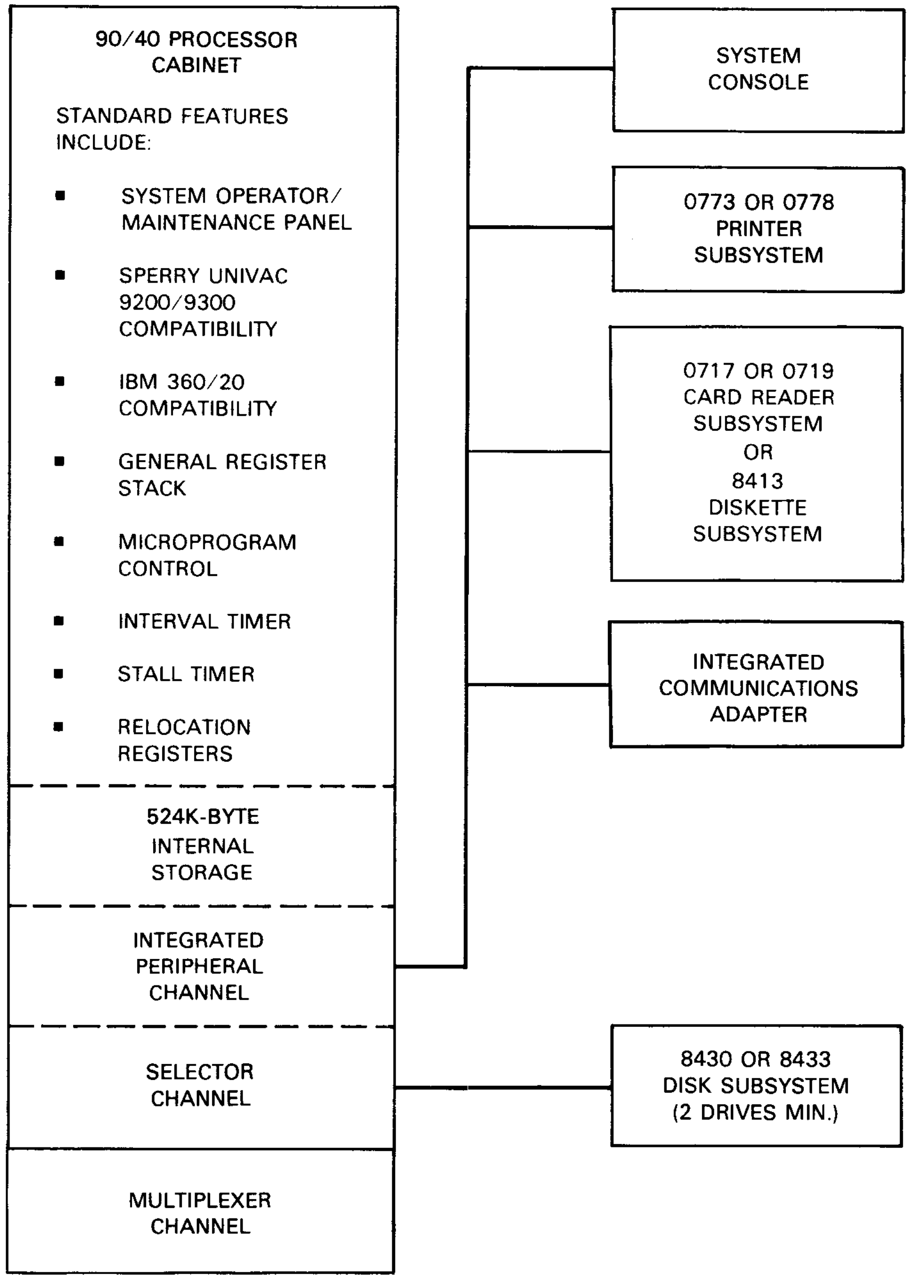
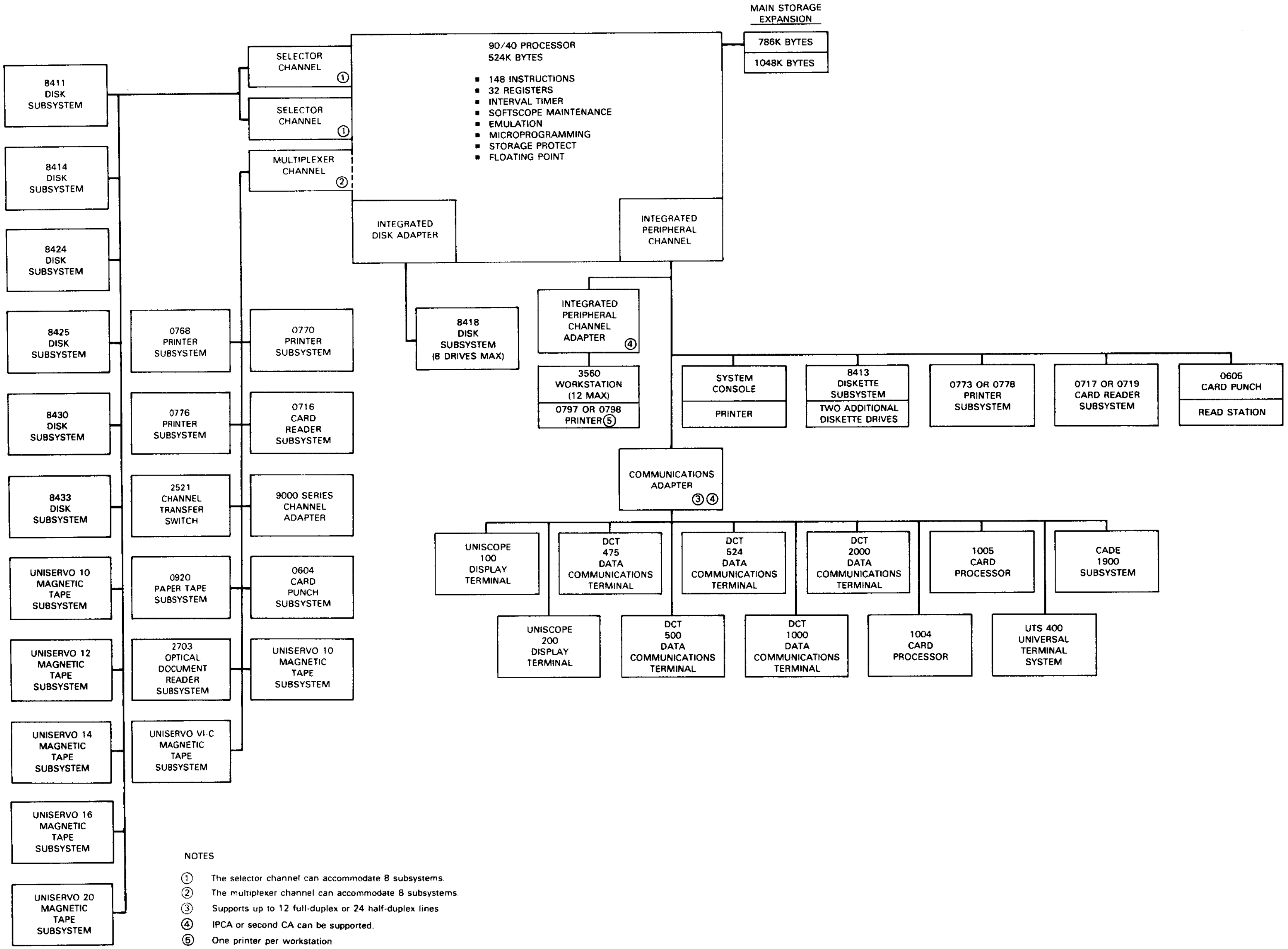

90/40 PROCESSOR CABINET
STANDARD FEATURES INCLUDE:
■ SYSTEM OPERATOR/ MAINTENANCE PANEL
■ SPERRY UNIVAC 9200/9300 COMPATIBILITY
■ IBM 360/20 COMPATIBILITY
■ GENERAL REGISTER STACK
■ MICROPROGRAM CONTROL
■ INTERVAL TIMER
■ STALL TIMER
■ RELOCATION REGISTERS
524K-BYTE INTERNAL STORAGE
INTEGRATED PERIPHERAL CHANNEL
SELECTOR CHANNEL
MULTIPLEXER CHANNEL
SYSTEM CONSOLE
0773 OR 0778 PRINTER SUBSYSTEM
0717 OR 0719 CARD READER SUBSYSTEM OR 8413 DISKETTE SUBSYSTEM
INTEGRATED COMMUNICATIONS ADAPTER
8430 OR 8433 DISK SUBSYSTEM (2 DRIVES MIN.)

Figure 1—4. Basic 90/40 System Configuration


MAIN STORAGE EXPANSION
786K BYTES
1048K BYTES
90/40 PROCESSOR
524K BYTES
148 INSTRUCTIONS
32 REGISTERS
INTERVAL TIMER
SOFTSCOPE MAINTENANCE
EMULATION
MICROPROGRAMMING
STORAGE PROTECT
FLOATING POINT
SELECTOR CHANNEL ①
SELECTOR CHANNEL ①
MULTIPLEXER CHANNEL ②
INTEGRATED DISK ADAPTER
INTEGRATED PERIPHERAL CHANNEL
8411 DISK SUBSYSTEM
8414 DISK SUBSYSTEM
8424 DISK SUBSYSTEM
8425 DISK SUBSYSTEM
8430 DISK SUBSYSTEM
8433 DISK SUBSYSTEM
UNISERVO 10 MAGNETIC TAPE SUBSYSTEM
UNISERVO 12 MAGNETIC TAPE SUBSYSTEM
UNISERVO 14 MAGNETIC TAPE SUBSYSTEM
UNISERVO 16 MAGNETIC TAPE SUBSYSTEM
UNISERVO 20 MAGNETIC TAPE SUBSYSTEM
0768 PRINTER SUBSYSTEM
0776 PRINTER SUBSYSTEM
2521 CHANNEL TRANSFER SWITCH
0920 PAPER TAPE SUBSYSTEM
2703 OPTICAL DOCUMENT READER SUBSYSTEM
UNISERVO VI-C MAGNETIC TAPE SUBSYSTEM
0770 PRINTER SUBSYSTEM
0716 CARD READER SUBSYSTEM
9000 SERIES CHANNEL ADAPTER
0604 CARD PUNCH SUBSYSTEM
UNISERVO 10 MAGNETIC TAPE SUBSYSTEM
8418 DISK SUBSYSTEM (8 DRIVES MAX)
INTEGRATED PERIPHERAL CHANNEL ADAPTER ④
3560 WORKSTATION (12 MAX)
0797 OR 0798 PRINTER⑤
SYSTEM CONSOLE
PRINTER
8413 DISKETTE SUBSYSTEM
TWO ADDITIONAL DISKETTE DRIVES
0773 OR 0778 PRINTER SUBSYSTEM
0717 OR 0719 CARD READER SUBSYSTEM
0605 CARD PUNCH
READ STATION
COMMUNICATIONS ADAPTER ③④
UNISCOPE 100 DISPLAY TERMINAL
DCT 475 DATA COMMUNICATIONS TERMINAL
DCT 524 DATA COMMUNICATIONS TERMINAL
DCT 2000 DATA COMMUNICATIONS TERMINAL
1005 CARD PROCESSOR
CADE 1900 SUBSYSTEM
UNISCOPE 200 DISPLAY TERMINAL
DCT 500 DATA COMMUNICATIONS TERMINAL
DCT 1000 DATA COMMUNICATIONS TERMINAL
1004 CARD PROCESSOR
UTS 400 UNIVERSAL TERMINAL SYSTEM

NOTES

① The selector channel can accommodate 8 subsystems.

② The multiplexer channel can accommodate 8 subsystems.

③ Supports up to 12 full-duplex or 24 half-duplex lines

④ IPCA or second CA can be supported.

⑤ One printer per workstation

Figure 1—5. 90/40 System Configuration with All Hardware Options Shown

1.3. CENTRAL HARDWARE

The central hardware of a basic 90/30 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 and 1-5. 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 subsections.

1.3.1. Processor

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

- Basic instruction set (90/30) 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 system, 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 system. 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. The system automatically monitors and protects the main storage contents to ensure data integrity. 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 independently 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, integrated peripheral channel adapter and workstation, and the 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 90/40 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 designed 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 system 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 system, 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 system 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 system provides I/O capability between the processor and up to eight subsystems with a throughput rate of 83K bytes per second. The 90/30 system uses 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 that 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 90/40 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 subsystems (no more than eight) 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 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 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 and 90/40 systems include a workstation subsystem, disk and diskette subsystems, magnetic and paper tape subsystems, high-speed printer subsystem, card reader and punch subsystems, an optical document reader subsystem, and a communications adapter with remote I/O subsystems.

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 (Part 1 of 2)

Document Number (Current Version)				
Processor/Device	General Description, UP-	Subsystem Programmer Reference UP-	Operator Reference UP-	Programmer/ Operator Reference UP-
Processors			7781	
9200/9300	8547	8460	8459	
90/30		8052	8097	
90/40			8522	
Integrated Peripheral Channel (IPC)				
90/30, 90/40		8041		
Card Punch				
0604		7772	7773	
0605	8192		8088	
Card Reader				
0716			7921	
0717	8196		8089	
0719	8493			
Printer				
0768			7931	7688
0770		8016	7938	
0773	8191		8086	
0776	8354	8441	8250	
0778	8524		8525	
8541 (COP)	7939			
0797	9159		9160	
0798	8871		8882	
UNISERVO Magnetic Tapes				
VI-C				7644
10/14	8206	8205	8207	
12/16				7661
U20			7956	
Disks				
8411	7605	7977	7802	
8414	7691	7977	7802	
8415			8511	
8416			8511	
8418		8362	8361	
8424			7983	
8425			8230	
8430		8344	8343	
8433		8344	8343	

Table 1—1. System Hardware Documentation (Part 2 of 2)

Processor/Device	Document Number (Current Version)			
	General Description, UP-	Subsystem Programmer Reference UP-	Operator Reference UP-	Programmer/ Operator Reference UP-
Diskette 8413	8463		8490	
Paper Tape 0920	7595	7998	7830	
Optical Document Reader (ODR) 2703	7710	7993	7994	
Channel Transfer Switch 2521			8489	
Workstation 3560		8742	8880	
Communications Adapter	8273	8247		
UNISCOPE 100/200	8155		7788	7807
DCT 475/500/524 Data Communications Terminal	7804		7832	7836
DCT 1000 Data Communications Terminal	7782		7827	7859
DCT 2000 Data Communications Terminal	7511		7545	7532
Card Processor 1004/1005			7839	
UTS 400 Universal Terminal System	8358		8357	
CADE 1900	8335			

1.6. OPERATING SYSTEM

Operating System/3 (OS/3) is used with the 90/30 and 90/40 systems. OS/3 (Figure 1-6) 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, application programs, and interactive services.

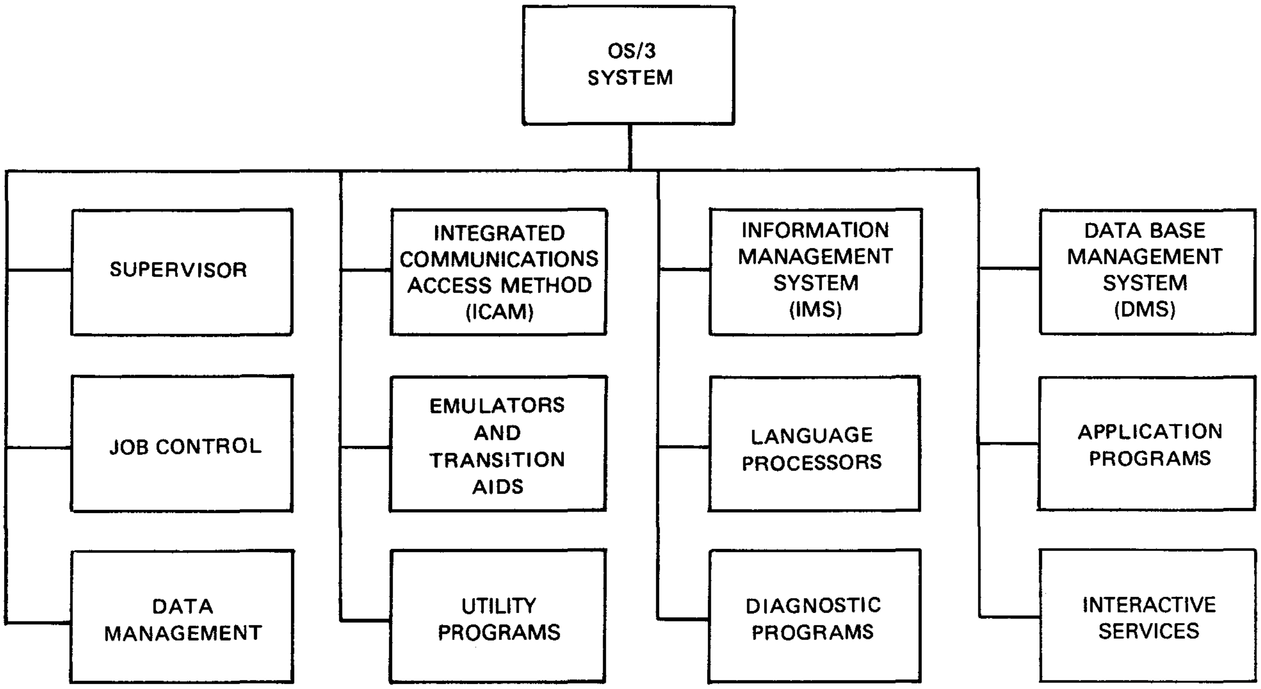

OS/3 SYSTEM
SUPERVISOR
JOB CONTROL
DATA MANAGEMENT
INTEGRATED COMMUNICATIONS ACCESS METHOD (ICAM)
EMULATORS AND TRANSITION AIDS
UTILITY PROGRAMS
INFORMATION MANAGEMENT SYSTEM (IMS)
LANGUAGE PROCESSORS
DIAGNOSTIC PROGRAMS
DATA BASE MANAGEMENT SYSTEM (DMS)
APPLICATION PROGRAMS
INTERACTIVE SERVICES

Figure 1—6. 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 ,:ogrammer 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 macroinstructions, and imperative macroinstructions.

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) that communicate with the logical control levels via declarative and imperative macroinstructions. 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;
- the 9200/9300 emulator; and
- the IBM System/3 transition program.

1.6.6. Utility Programs

The OS/3 utility 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 utility programs include:

- Data utilities
- Linkage editor
- System librarian
- Sort/merge
- Disk, diskette, and tape prep routines
- System utility

1.6.7. Language Processors

Six language processors are available with OS/3: assembler, COBOL, FORTRAN, report program generator II (RPG II), BASIC, and ESCORT. 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) used with OS/3 is common to the 9000 series and facilitates access to information stored in data files. IMS 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 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) 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 data bases are derived from the CODASYL data base specifications. A data base may be accessed by batch application programs and communications application programs.

1.6.12. Interactive Services

Interactive services provide an extensive interactive command set that is available to a user as part of the standard software package. These commands enable you to control the interactive system environment, including all jobs within the system, all workstation users (local locations), and all terminal users (local and remote locations).

Interactive services also perform a variety of functions, including program creation and file/data manipulation. The interactive facilities used to perform these functions are:

- Screen format services
- Interactive data utilities
- General editor
- Interactive job stream preparation

2. System Turn-on and Turn-off Procedures

2.1. SYSTEM TURN ON

To turn on your 90/30 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 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 turned off from its operator control panel because 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. This prevents the card punch from being allocated to a job while offline.

3. System Initialization Procedures

3.1. GENERAL

Initialization of the 90/30 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 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.

NOTE:

At this point, you specify the 3-digit number for the control storage load device identification (did). The did is a concatenation of the channel, subchannel, and selected device number. The did is represented by the device address you set in steps f, g, and h.

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 (IPL) 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 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 position.

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

   NOTE:

   At this point, you specify the 3-digit number for the program load device identification (did). The did is a concatenation of the channel, subchannel, and selected device number. The did is represented by the device address you set in steps f, g, and h.

   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 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 (not turned on) during IPL time to not available. 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).*

I. 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, press the TRANSMIT key again.

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 not 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.

The date entered is compared to the date keyed in at the last load from the same SYSRES. If the date entered is six days less than or six days greater than the date of the last load, the message DATE QUESTIONABLE appears. If the date entered is correct, press the TRANSMIT key and the date is accepted. If the date is incorrect, key in the correct date.

```
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=system-generation-option)___
```

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

Remember, the volume identified as the $Y$RUN 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)
```

To retain jobs previously filed in the job queue for processing, key in Y; to place them into hold status, key in H. To delete them 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. To clear all accumulated errors in the error log and start a new error log file, key in N. Otherwise, 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=system-generation-option)
```

This message appears only if the spooling option is configured in your system. It requests that you specify the level of recovery desired for the spool file in your system. To specify the same level of recovery that was specified during SYSGEN, press the RETURN key to advance to the next line. To specify a different level of recovery, key 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

Recover all spooled subfiles when the spool file is reinitialized because previously spooled input or output files in the spool file are to be processed. With this response, all spooled subfiles, whether complete or incomplete, are saved. This response (or H) must be specified to recover the console and workstation log file (if configured into the system). Console and workstation 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

Recover only completed subfiles when the spool file is initialized.

L

Recover only the user log directory 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 pressing the SYSTEM RESET switch, then the RUN switch. (Refer to the dump analysis manual, UP-8837 (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. (Workstation messages that accumulate in the main storage buffer are not recovered, however.) Use this method of recovery (sometimes referred to as the hot start) only if a system crash occurs. When 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.

$$\text{SPOOLING DVC ADDR? }\left(\text{DEFAULT=}\begin{Bmatrix}\text{blank}\\ \text{vsn}\\ \text{SYSRES}\end{Bmatrix}\right)\text{---}$$

This message appears only if the spooling option is configured in your system. Further, if multivolume spooling was configured, it appears 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 disk volumes that are to contain the spool file. To use the volume specified during SYSGEN, identified in the message as the default value, press the space bar three times. 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 RETURN key, as required, to position cursor to point of entry for next statement. Remember, all volumes identified as spool volumes at IPL time must be online for the system to be operational.

MODIFY SUPERVISOR? (N, Y DEFAULT=N)

If no modification to the supervisor is required (the SYSGEN selections for the supervisor are to be used), press the TRANSMIT key on the system console. This action causes the message presented in step m to be displayed.

If the supervisor is to be modified, key in Y. The screen clears and system output messages are displayed. Proceed with the appropriate operator action described in Appendix C.

NUMBER OF 32K BYTE BLOCKS FOR CACHE (Ø-4: Ø=NO CACHE, DEF=3) #

This message appears only if the disk cache feature is configured in your system. Reply with a number from 0 to 4:

Ø

Tells the system to run without the cache feature.

1, 2, 3, or 4

Tells the system to set up a cache of 32K, 65K, 98K, or 131K bytes, respectively.

If your reply does not fall between 0 and 4 or if you do not reply at all, the system defaults to 98K bytes. The cache operates at its present size until the next IPL.

CAUTION

Be sure to respond to all the previous 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 is 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:

If three jobs named A, B, and C are running in the system, the screen format is:

```
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 OS/3 software version loaded into the system.

supnam

Specifies the name of the supervisor loaded into the system.

n

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

yy/mm/dd
Signifies the year, month, and day used by the system.

hh:mm:ss
Signifies the time in hours, minutes, and seconds 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 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 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 (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 (4.4.7) 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,did,TY,2ØØ2
```

Jobs requiring more than the available devices are terminated with an R277 message. The operator may review device ready status with the MIX command (4.4.2) and change the status with the SET IO command.

NOTE:

When the IPL is completed, you should immediately initiate the transient work area feature if your system doesn't display the message "TRANSIENT WORK AREA IS INITIALIZED". To do this, press the MESSAGE WAITING key, key in the TW command, and then press the TRANSMIT key. For more information about the transient work area, see 4.4.14.

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 (or the input data required by a prefiled job to be processed) on the input device. 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 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, you must press the MESSAGE WAITING key on the system console. 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 is 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-digit device identification number that identifies the channel, subchannel, and selected device number to be used when carrying out the command. A did 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 is 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 did 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 did by a comma. The diskette record size must be 128 bytes or less, and the records must be unblocked.

`(RDR,label)`
: Specifies that the device address to be used is the input 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`
  Specifies the diskette with a device address 010 contains a file labeled DATANAM1 to be accessed by the command.

  `RUN (,DATANAM1) JOBNAM1`
  Specifies the diskette configured as SYSRDR contains a file labeled DATANAM1 to be accessed by the command.

  `RUN (RDR,DATANAM1) JOBNAM1`
  Specifies the input spool file contains a subfile labeled DATANAM1 to be accessed by the command.

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, this portion of the RUN command format

```
RUN ⎡⎧(did)          ⎫⎤ △ ⎡⎧jobname (new-name)⎫⎤ ⎡,⎧PRE ⎫⎤
    ⎢⎨([did],label)  ⎬⎥   ⎣⎩(new-name)        ⎭⎦ ⎢ ⎨HIGH⎬⎥
    ⎣⎩(RDR,label)    ⎭⎦                          ⎣ ⎩NOR ⎭⎦
```

indicates that two positional parameters are associated with the 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 requires 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:

```
▷ ji⎧?⎫message-text
    ⎨△⎬
    ⎩•⎭
```

where:

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

j
Is a 1-digit job number 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 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.

NOTE:

In some cases, an apparent system halt is caused by the operator's failure to answer output messages that require a response. Before deciding that the system is in a halt condition, you should ensure that all output messages have been answered.

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. Unsolicited messages may be entered from the console or from the workstation that initiated the job or symbiont. The format for all unsolicited input messages is:

```
▷ j0△[symbiont-name][(did)]△message-text
```

where:

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

`0`
: Is the message number used to identify the message as an unsolicited message.

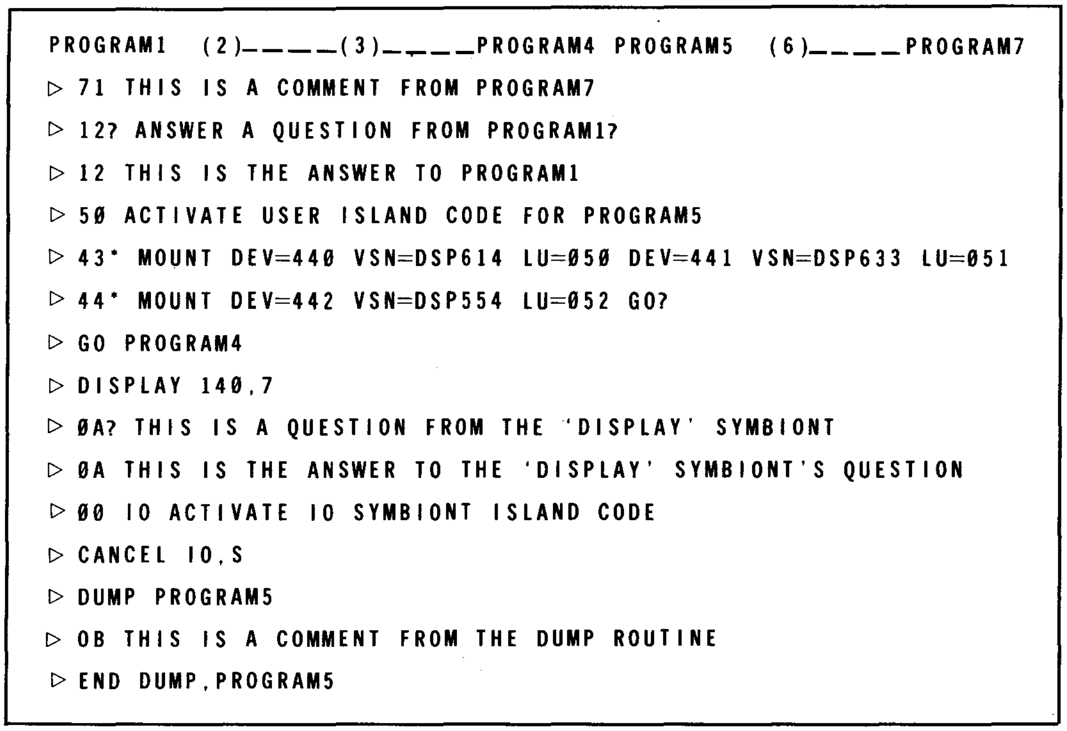
`symbiont-name`
: Is the 2-character alphanumeric name of the supervisor symbiont to receive the unsolicited message (the job and message numbers are 00). If a symbiont is not the recipient of the message, no symbiont name is required.

`did`
: Is the address of the device 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 did must be enclosed in parentheses.

`message-text`
: Is the actual text of the message.

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.



```
       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     ▷ 5Ø ACTIVATE USER ISLAND CODE FOR PROGRAM5
 5     ▷ 43* MOUNT DEV=44Ø VSN=DSP614 LU=Ø5Ø DEV=441 VSN=DSP633 LU=Ø51
 6     ▷ 44* MOUNT DEV=442 VSN=DSP554 LU=Ø52 GO?
 7     ▷ GO PROGRAM4
 8     ▷ DISPLAY 14Ø,7
 9     ▷ ØA? THIS IS A QUESTION FROM THE 'DISPLAY' SYMBIONT
10     ▷ ØA THIS IS THE ANSWER TO THE 'DISPLAY' SYMBIONT'S QUESTION
11     ▷ ØØ IO ACTIVATE IO SYMBIONT ISLAND CODE
12     ▷ CANCEL IO,S
13     ▷ DUMP PROGRAM5
14     ▷ OB THIS IS A COMMENT FROM THE DUMP ROUTINE
15     ▷ END DUMP,PROGRAM5
```

Figure 4—1. Typical System Console Messages

The following subsections 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 begin execution. Jobs to be run are placed in one of the three scheduling priority queues:

- PRE (preemptive) Queue

  Contains jobs to be executed first, i.e., before any jobs assigned HIGH or NOR scheduling priority, even if the resources for any HIGH or NOR priority jobs are available and the resources for any PRE jobs aren't. PRE priority jobs are always executed first unless they're placed on hold via a HOLD command. (See 4.3.2.1) If rollin/rollout is configured, a PRE job 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 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. HIGH priority jobs are always executed before NOR priority jobs (unless they're placed on hold), even if the resources for a HIGH priority job are not available and the resources for the NOR job are.

- NOR (normal) Queue

  Contains jobs to be executed only when there are no jobs left in the PRE or HIGH queues or when the queues are 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

Job control streams are read into the system by using one of the job initialization commands (FILE, RUN/RV, SI/SC, and OCL/OV). These commands enable you to file the job for future use or to process the job immediately.

When a job is filed for future use, it is placed in the job control stream library ($Y$JCS) file or in an alternate library file, as specified in the operator command.

A job may be initiated for processing from one of three places:

1. from an input device (card reader, diskette, or spool file);

2. from $Y$JCS or an alternate library where the job is filed; or

3. from the $Y$SAVE file where it has been saved in its expanded *run* state via a statement included in the job control stream (// OPTION SAVE or // OPTION NOSCHED statement).

When a job is initiated, it is placed in a scheduling priority queue to await execution. This scheduling priority queue can be specified by the programmer submitting the job in the job control stream itself. The operator can override this specification by entering another scheduling priority in the job initialization command. The default is to use the normal priority queue.

4.3.1.1. Filing Job Control Streams (FILE)

Function:

The FILE command files jobs and JPROCs, read from an input device, into the permanent JCS library file ($Y$JCS) or an alternate library file. (The alternate file may be a MIRAM or a SAT file, but keep in mind that the RUN processor cannot access data in a MIRAM file.) The input device can be a card reader, a diskette drive, or the input spool file.

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.)

Jobs filed from the card reader must terminate with a // FIN job control statement. When jobs are filed from a diskette or the spool file, the // FIN job control statement is not needed. Jobs input from diskette to the spool file must be single volume.

The FILE command cannot be issued from an enter stream.

Format:

```
FILE ⎡⎧(did)            ⎫⎤ △ ⎡⎧:alt-filename                                          ⎫⎤
     ⎢⎨([did],label)    ⎬⎥   ⎢⎪:/alt-filename,⎧RES⎫\                                  ⎪⎥
     ⎣⎩(RDR,label)      ⎭⎦   ⎢⎪ (             ⎨RUN⎬ )                                 ⎪⎥
                             ⎢⎨ \             ⎩vsn⎭/                                  ⎬⎥
                             ⎢⎪:/alt-filename,⎡⎧RES⎫⎤,write-password\                 ⎪⎥
                             ⎢⎪ (             ⎢⎨RUN⎬⎥                )                ⎪⎥
                             ⎣⎩ \             ⎣⎩vsn⎭⎦               /                 ⎭⎦
```

Positional Parameter 1:

Identifies an alternate library file where job control streams and/or JPROCs are to be filed. Omit this parameter when no alternate file is required: the job is filed into $Y$JCS.

:alt-filename
Specifies the name of the alternate library file, residing on SYSRES, to receive the job and/or JPROC. If the alternate file name is cataloged, the vsn of that file in the catalog is used. There must not be a write password for the alternate file in the catalog.

```
:/alt-filename,⎧RES⎫\
 (             ⎨RUN⎬ )
 \             ⎩vsn⎭/
```

Specifies the name of the alternate library file to receive the job and/or JPROC and identifies a volume serial number (RES, RUN, or vsn) for the file. You specify RES to identify SYSRES as the volume to contain the file, RUN to identify the system RUN pack as the volume to contain the file, or you may specify the volume serial number (vsn) of a disk pack to be used. If file with the same file name is in the catalog, the volume serial number you include in the command makes the distinction between the files and overrides the catalog vsn. There must not be a write password for the alternate file in the catalog. The colon and enclosing parentheses are optional and included only for consistency with the format of the RUN/RV command.

$$:\left(\texttt{alt-filename,}\left[\begin{Bmatrix}\texttt{RES}\\ \texttt{RUN}\\ \texttt{vsn}\end{Bmatrix}\right]\texttt{,write-password}\right)$$

Specifies the name of the alternate library file to receive the job and/or JPROC and includes the write password, identified in the catalog, required to write to that file. You specify RES to identify SYSRES as the volume to contain the file, RUN to identify the system RUN pack as the volume to contain the file, or the volume serial number (vsn) of a disk pack to be used. If you omit a volume serial number (RES, RUN, or vsn), your file is written to the volume associated with that file name in the catalog. You must specify RES, RUN, or a vsn if you want the file written to a different volume; the volume serial number you specify overrides the catalog vsn. The colon and enclosing parentheses are optional and included only for consistency with the format of the RUN/RV command.

Example:

Operator keyin:

```
FI :(ALTJCS,RUN)
```

Function requested:

The job control stream, residing on a card reader, is to be written to the alternate job control library file called ALTJCS on the system RUN pack.

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 $Y$JCS or alternate job control library file. The input device can be a card reader, a diskette drive, or the input spool file.

The commands cause the job control stream to be written to the $Y$RUN library file and expanded to its *run* state (JPROCs are expanded), then scheduled for execution. When an // OPTION SAVE job control statement is included in the job control stream, the job is scheduled to be run from $Y$RUN and a copy of the expanded job is saved in the $Y$SAVE file or an alternate save file. When an // OPTION NOSCHED job control statement is included in the job control stream, a copy of the expanded job is saved in $Y$SAVE or an alternate save file; however, the job is not scheduled to be run.

The expanded job to be saved in $Y$SAVE or an alternate save file can be run using the SI/SC commands. (SC/SI commands are described in 4.3.1.3.)

The RV command initiates 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 initiates the reading of a job control stream that requires the use of an input device (i.e., card reader, diskette, or spool file). This means if the job is initiated from an input device or if the job contains a // CR statement to read input, you must use RUN. 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. 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. If a duplicate RUN command is initiated for the same job, the supervisor queues the second command until the input device is available. Assuming the hopper is empty when the second RUN command executes, the system displays a hopper empty message.

Format:

```
⎧RUN⎡⎧(did)              ⎫⎤ △⎡⎧jobname[(new-name)]⎫⎤⎫
⎪   ⎢⎨([did],label)]    ⎬⎥  ⎣⎩(new-name)         ⎭⎦⎪
⎨   ⎣⎩(RDR,label)       ⎭⎦                          ⎬
⎪                                                   ⎪
⎩RV△jobname[(new-name)]                             ⎭

 ⎡⎧:alt-filename                                          ⎫⎤
 ⎢⎪:⎛alt-filename,⎧RES⎫⎞                                  ⎪⎥
 ⎢⎪ ⎜             ⎨RUN⎬⎟                                  ⎪⎥
 ⎢⎨ ⎝             ⎩vsn⎭⎠                                  ⎬⎥
 ⎢⎪:⎛alt-filename,⎡⎧RES⎫⎤,read-password⎞                  ⎪⎥
 ⎢⎪ ⎜             ⎢⎨RUN⎬⎥              ⎟                  ⎪⎥
 ⎣⎩ ⎝             ⎣⎩vsn⎭⎦              ⎠                  ⎭⎦

 ⎡,⎧PRE ⎫⎤[,key-1=val-1,...,key-n=val-n]
 ⎢ ⎨HIGH⎬⎥
 ⎣ ⎩NOR ⎭⎦
```

Command Code:

RUN

Initiates the running of a job control stream that requires an input device. You must specify a job name when the job control stream is prefiled. If it is prefiled and you don't specify an input device, the first available card reader is assigned to the job. If you omit the input device and job name, the first available card reader, as defined when the system was generated, is expected to contain the job control stream to be run.

You must specify a label if the job control stream is on a diskette. If it is in the input spool file, you must specify RDR and a label (4.2.1). For diskette and spool file input, the last // FIN job control statement is not needed because it is used only to terminate card reader operation. However, the // FIN statements that separate groups of card images read with // CR statements are still necessary. Jobs input from diskette to the spool file must be single volume.

RV

Initiates the running of a prefiled job control stream that does not require an input device, that is, does not contain a // CR (read card reader, diskette, or spool file) statement. You must specify a job name.

Positional Parameter 1:

jobname[(new-name)]

Identifies the name of the job to be read from $Y$JCS or an alternate job control library file and stored in a scheduling priority queue to await execution. The job name consists of one to eight alphanumeric characters and is required with RV.

You include new-name to assign a new 1- to 8-character alphanumeric name to a job already stored in $Y$JCS or an alternate job control library file. The job identified by the jobname parameter is read from $Y$JCS or an alternate file and stored in a scheduling priority queue under the name identified by the new-name parameter to await execution. The new name cannot contain blanks.

(new-name)

Used with the RUN command to assign 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. The new name cannot contain blanks.

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

Positional Parameter 2:

Used when the job control stream resides in an alternate job control library file, rather than in $Y$JCS. When the job resides in an alternate library file, this parameter identifies the library file to be read. If omitted, the job is read from $Y$JCS.

:alt-filename

Specifies the name of the alternate library file, residing on SYSRES, that contains the job. If the alternate file name is cataloged, the vsn of that file in the catalog is used. There must not be a read password for the alternate file in the catalog.

```
:(alt-filename,{RES})
               {RUN}
               {vsn}
```

Specifies the name of the alternate library file that contains the job or JPROC and identifies a volume serial number (RES, RUN, or vsn) for the file. You specify RES to identify SYSRES as the volume that contains the file, RUN to identify the system RUN pack as the volume that contains the file, or the volume serial number (vsn) of a disk pack to be read. If a file with the same file name is in the catalog, the volume serial number you include in the command makes the distinction between the files and overrides the catalog vsn. There must not be a read password for the alternate file in the catalog.

```
: alt-filename,[{RES}],read-password
               [{RUN}]
               [{vsn}]
```

Specifies the name of the alternate library file that contains the job stream and includes the read password, identified in the catalog, required to read from that file. You specify RES to identify SYSRES as the volume that contains the file, RUN to identify the system RUN pack as the volume that contains the file, or the volume serial number (vsn) of a disk pack to be read. If you omit a volume serial number (RES, RUN, or vsn), your file is read from the volume associated with that file name in the catalog. You must specify RES, RUN, or a vsn if you want the file read from a different volume; the volume serial number you specify overrides the catalog vsn.

Positional Parameter 3:

PRE

Places the job in the preemptive scheduling priority queue to await execution.

HIGH

Places the job in the high scheduling priority queue to await execution.

NOR

Places the job in the normal scheduling priority queue to await execution.

If positional parameter 3 is omitted, the scheduling priority specified in the job control stream is used. If not specified in the job stream, the normal priority is used.

Positional Parameters 4 through n:

key-1=val-1,...,key-n=val-n

Are the keywords and their values that may be used by the job being run. The keywords and their values must be supplied by the user 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.

Examples:

1. Operator keyin:

```
RU MYJOB:(ALTJCS,RUN)
```

Function requested:

The job named MYJOB, which is filed in the alternate job control library file ALTJCS on the system RUN pack, is to be run under the priority specified in the job control stream. The first available card reader is expected to contain some input for MYJOB that contains a // CR statement.

2. Operator keyin:

```
RV MYJOBA(NETPAY)
```

Function requested:

The job named MYJOBA, filed in YJCS, is to be run under the new name NETPAY according to the priority specified in the job control stream.

4.3.1.3. Running Saved Job Control Streams (SI/SC)

Function:

The SI/SC commands initiate the running of a job control stream from the YSAVE MIRAM library file or from an alternate library file, then schedule the job for execution. In either case, the control stream has been saved in its expanded *run* state. A control stream is expanded in the YRUN file when the RUN/RV or OCL/OV command is issued for the job. When an // OPTION SAVE or // OPTION NOSCHED job control statement is included in the job control stream, a copy of the expanded control stream is stored in YSAVE or an alternate file for subsequent runs using SI or SC.

The SC command is used only to initiate the reading of a job control stream that does not require an input device to replace embedded data. The SI command initiates the reading of a job control stream that requires an input device (i.e., card reader, diskette, or spool file) to replace embedded data. When the SI command is issued, it is accepted only if an input device is available. The SC command allows a job control stream to be initiated that does not require an input device.

Format:

```
⎧SI ⎡⎧(did)          ⎫⎤⎫ △ jobname[(new-name)]
⎨   ⎢⎨([did],label)  ⎬⎥⎬
⎪   ⎣⎩(RDR,label)    ⎭⎦⎪
⎩SC                    ⎭

      ⎡⎧:alt-filename                                 ⎫⎤ ⎡,⎧PRE ⎫⎤
      ⎢⎪:⎛alt-filename,⎧RES⎫⎞                         ⎪⎥ ⎢ ⎨HIGH⎬⎥
      ⎢⎪ ⎜             ⎨RUN⎬⎟                         ⎪⎥ ⎣ ⎩NOR ⎭⎦
      ⎢⎨ ⎝             ⎩vsn⎭⎠                         ⎬⎥
      ⎢⎪:⎛alt-filename,⎡⎧RES⎫⎤,read-password⎞         ⎪⎥
      ⎢⎪ ⎜             ⎢⎨RUN⎬⎥              ⎟         ⎪⎥
      ⎣⎩ ⎝             ⎣⎩vsn⎭⎦              ⎠         ⎭⎦
```

Command Code:

SI

Initiates the running of a job control stream that requires an input device. If no device and label are identified, the first available card reader, as defined when the system was generated, is expected to contain the replacement data required by the job. If the data 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.) For diskette and spool file input, the last // FIN job control statement is not necessary.

SC

Initiates the running of a job control stream that does not require an input device to replace embedded data.

Positional Parameter 1:

jobname[(new-name)]

Identifies the name of the job to be read from $Y$SAVE or an alternate library file and stored in a scheduling priority queue to await execution. The job name consists of one to eight alphanumeric characters.

You include the new-name parameter to assign a new 1- to 8-character alphanumeric name to a job stored in $Y$SAVE or an alternate file. The job identified by the jobname parameter is read from $Y$SAVE or an alternate file and stored in a scheduling priority queue under the name identified by the new-name parameter to await execution. The new name cannot contain blanks.

Positional Parameter 2:

Used when the job control stream resides in an alternate job control library file on disk or format label diskette, rather than in $Y$SAVE. When the job resides in an alternate library file, this parameter identifies the library file to be read. If omitted, the job is read from $Y$SAVE.

:alt-filename

Specifies the name of the alternate library file residing on SYSRES that contains the job. If the alternate file name is cataloged, the vsn of that file in the catalog is used. There must not be a read password for the alternate file in the catalog.

$:\left(\text{alt-filename,}\begin{Bmatrix}\text{RES}\\ \text{RUN}\\ \text{vsn}\end{Bmatrix}\right)$

Specifies the name of the alternate library file that contains the job or JPROC and identifies a volume serial number (RES, RUN, or vsn) for the file. You specify RES to identify SYSRES as the volume that contains the file or RUN to identify the system RUN pack as the volume that contains the file; or you may specify the volume serial number (vsn) of a disk pack or format label diskette to be read. If a file with the same file name is in the catalog, the volume serial number you include in the command makes the distinction between the files and overrides the catalog vsn. There must not be a read password for the alternate file in the catalog.

$:\left(\text{alt-filename,}\left[\begin{Bmatrix}\text{RES}\\ \text{RUN}\\ \text{vsn}\end{Bmatrix}\right]\text{,read-password}\right)$

Specifies the name of the alternate library file that contains the job stream and includes the read password, identified in the catalog, required to read from that file. You specify RES to identify SYSRES as the volume that contains the file or RUN to identify the system RUN pack as the volume that contains the file; or you may specify the volume serial number (vsn) of a disk pack or format label diskette to be read. If you omit a volume serial number (RES, RUN, or vsn), your file is read from the volume associated with that file name in the catalog. You must specify RES, RUN, or a vsn if you want the file read from a different volume; the volume serial number you specify overrides the catalog vsn.

Positional Parameter 3:

PRE

Places the job in the preemptive scheduling priority queue to await execution.

HIGH

Places the job in the high scheduling priority queue to await execution.

NOR

Places the job 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.

Example:

Operator keyin:

```
SI MYJOB(AVGYTD),H
```

Function requested:

The job called MYJOB, expanded and filed in $Y$SAVE, is to be run under the new name AVGYTD. The job is to be placed in the high scheduling priority queue to await execution. Replacement embedded data for AVGYTD is expected to be found on the first available card reader.

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

Function:

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

The OV command initiates the reading of a prefiled control stream that does not contain a // CR statement indicating that there is input (cards, diskette, or spool file) to be read and inserted into the stream. The OCL command initiates the reading of a control stream that requires an input device (i.e., card reader, diskette, or spool file). When the OCL command is issued, it is accepted only if an input device is available, whether or not one is needed by the control stream being read. The OV command allows a control stream to be initiated that does not require an input device. You must include a job name when you enter an OV command.

Format:

```
⎧OCL ⎡⎧(did)          ⎫⎤ △[(new-name)]      ⎫⎡ ,⎧PRE ⎫⎤ [,key-1=val-1,...,key-n=val-n]
⎨    ⎢⎨([did],label)  ⎬⎥                    ⎬⎢  ⎨HIGH⎬⎥
⎪    ⎣⎩(RDR,label)    ⎭⎦                    ⎪⎣  ⎩NOR ⎭⎦
⎩OV△jobname-library-unit[(new-name)]        ⎭
```

Command Code:

OCL

Initiates the running of a control stream that requires an input device. If you omit the input device, the first available card reader, as defined when the system was generated, is expected to contain the control stream to be run.

You must specify a label if the control stream is on a diskette. If it is in the input spool file, you must specify RDR and a label (4.2.1). For diskette and spool file input, the last // FIN job control statement is not needed because it is used only to terminate card reader operation. However, the // FIN statements that separate groups of card images read with // CR statements are still necessary. Jobs input from diskette to the spool file must be single volume.

OV

Initiates the running of a prefiled control stream that does not require an input device; that is, does not contain a // CR (read card reader, diskette, or spool file) statement. You must specify a jobname–library-unit.

Positional Parameter 1:

(new-name)

Used with the OCL command to assign a new 1- to 8-character alphanumeric name to the job; otherwise, the job name is taken from the // JOB statement or is defined as OCLnnnn by default, where nnnn is a decimal number from 0001 to 9999. The new name cannot contain blanks.

jobname–library-unit[(new-name)]

Identifies the name of the job to be read from the library specified by the library-unit code. The job name must be appended with a dash (11-punch), then the library-unit code. Jobname – library-unit is required with OV.

You include *new-name* to assign a new 1- to 8-character alphanumeric name to a job that is stored in the library specified by the library-unit code. The job identified by the *jobname* parameter is read from the library-unit specified, and stored in a scheduling priority queue under the name identified by the *new-name* parameter to await execution. The job name and library-unit must be separated by a dash (11-punch). The new name cannot contain blanks.

If omitted from the OCL command, the job name is taken from the // JOB statement or is defined as OCLnnnn by default.

Positional Parameter 2:

PRE
Places the job in the preemptive scheduling priority queue to await execution.

HIGH
Places the job in the high scheduling priority queue to await execution.

NOR
Places the job in the normal scheduling priority queue to await execution.

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

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 user requesting the job.

Examples:

1. Operator keyin:

```
OC
```

Function requested:

The control stream is to be run from the first available card reader under the priority assigned in the control stream. The job name is taken from the // JOB statement or will be OCLnnnn by default.

2. Operator keyin:

```
OV MYJOB-F1(PAYROLL)
```

Function requested:

The job called MYJOB, filed in the library F1, is to be run under the new name PAYROLL according to the priority assigned in the control stream.

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) to execute that job. 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 by using job scheduling commands. The operator may also control jobs initiated by a specific workstation user. 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 jobs according to the command parameters specified. You can defer scheduling for all jobs in all queues or in a specific queue; for a specific job within a queue; for a specific workstation user's jobs in all queues or a specific queue; or for a specific host's jobs in all queues or a specific queue. Scheduling remains deferred until the jobs are reactivated via the BEGIN command.

Format:

```
HOLD△⎧JBQ⎡,⎧PRE ⎫⎤⎧⎡,⎧OLD⎫⎤⎡,⎧DDP  ⎫⎤              ⎫⎫
     ⎪   ⎢ ⎪HIGH⎪⎥⎪⎣ ⎩NEW⎭⎦⎢ ⎪LOCAL⎪⎥              ⎪⎪
     ⎨   ⎢ ⎨NOR ⎬⎥⎨         ⎢ ⎨RBP  ⎬⎥              ⎬⎬
     ⎪   ⎣ ⎩ALL ⎭⎦⎪         ⎣ ⎩WKSTN⎭⎦              ⎪⎪
     ⎪            ⎩[,UID=user-id][,HOST=host-id]    ⎭⎪
     ⎩jobname                                         ⎭
```

Positional Parameter 1:

```
JBQ⎡,⎧PRE ⎫⎤
   ⎢ ⎪HIGH⎪⎥
   ⎢ ⎨NOR ⎬⎥
   ⎣ ⎩ALL ⎭⎦
```

Specifies the command applies to the jobs in a job scheduling priority queue, as further defined by PRE, HIGH, NOR, or ALL, where:

PRE
: Defers the jobs in the preemptive scheduling priority queue.

HIGH
: Defers the jobs in the high scheduling priority queue.

NOR

Defers the jobs in the normal scheduling priority queue.

ALL

Defers the jobs in all scheduling priority queues.

jobname

Defers a particular job from being scheduled for execution. No further parameters are permitted.

Keyword Parameters:

OLD

Defers only jobs already in the scheduling priority queue defined in parameter 1. Jobs subsequently entered in this queue are not to be deferred.

NEW

Defers only jobs subsequently placed in the scheduling priority queue defined in parameter 1. All existing jobs are still available for execution.

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.

DDP

Defers only distributed data processing jobs initiated from a remote host.

LOCAL

Defers only locally entered jobs.

RBP

Defers only jobs entered remotely (i.e., from a remote batch terminal).

WKSTN

Defers only jobs initiated from a workstation.

NOTE:

All of the above keyword parameters may be interchanged.

UID=user-id

Defers all old jobs associated with a particular workstation user-id in the scheduling priority queue defined in parameter 1. Jobs subsequently entered with this user-id are not deferred. The user-id is a 1- to 6-character alphanumeric identification that does not start with Y.

HOST=host-id

Defers all old jobs associated with a particular host in the scheduling priority queue defined in parameter 1. Jobs subsequently entered with this host-id are not deferred. The host-id is one to four alphanumeric characters.

NOTES:

1. *The UID and HOST parameters may be interchanged. No further keyword parameters are permitted.*

2. *If a command is entered from the system console (local or remote), the HOST parameter should not be used; that site's host-id is used automatically.*

3. *The special user-id OPERATOR indicates that the command applies to console-initiated jobs.*

4. *Any command from the system console (local or remote) pertains only to jobs with a matching host-id, regardless of user-id.*

5. *DDP users submitting the HOLD command via a parameterized enter stream may use HOST=$HOST to specify that the host-id of the command's submitter be used.*

Examples:

1. Operator keyin:

```
HO JBQ,A,NE,WKSTN
```

Function requested:

All jobs subsequently initiated from workstations and placed in all scheduling priority queues are to be deferred.

2. Operator keyin:

```
HO JBQ,H,UID=WKSTA1
```

Function requested:

All jobs in the high scheduling priority queue associated with the workstation user WKSTA1 are to be deferred.

3. Operator keyin:

```
HO JBQ,N,HOST=B
```

Function requested:

All jobs in the normal priority queue and associated with host B are to be deferred. New jobs entered into this queue are not to be deferred.

4.3.2.2. Scheduling Deferred Jobs (BEGIN)

Function:

The BEGIN command reinstitutes the scheduling for execution of currently deferred jobs, according to the command parameters specified. You can reinstitute scheduling for all jobs in all queues or in a specific queue; for a specific job within a queue; for a specific workstation user's jobs in all queues or a specific queue; or for a specific host's jobs in all queues or a specific queue. Jobs remain deferred by a HOLD command until you enter a BEGIN command to permit their rescheduling for execution.

Format:

```
BEGIN△ ⎧JBQ ⎡,⎧PRE ⎫⎤⎧⎡,⎧OLD⎫⎤⎡,⎧DDP  ⎫⎤                  ⎫⎫
       ⎪    ⎢ ⎪HIGH⎪⎥⎪⎣ ⎩NEW⎭⎦⎢ ⎪LOCAL⎪⎥                  ⎪⎪
       ⎨    ⎢ ⎨NOR ⎬⎥⎨        ⎢ ⎨RBP  ⎬⎥                  ⎬⎬
       ⎪    ⎣ ⎩ALL ⎭⎦⎪        ⎣ ⎩WKSTN⎭⎦                  ⎪⎪
       ⎪             ⎩[,UID=user-id][,HOST=host-id]       ⎭⎪
       ⎩jobname                                            ⎭
```

Positional Parameter 1:

```
JBQ ⎡,⎧PRE ⎫⎤
    ⎢ ⎪HIGH⎪⎥
    ⎢ ⎨NOR ⎬⎥
    ⎣ ⎩ALL ⎭⎦
```

Specifies the command applies to the jobs in a job scheduling priority queue, as further defined by PRE, HIGH, NOR, or ALL, where:

PRE

Permits the jobs in the preemptive scheduling priority queue to be scheduled for execution.

HIGH

Permits the jobs in the high scheduling priority queue to be scheduled for execution.

NOR

Permits the jobs in the normal scheduling priority queue to be scheduled for execution.

ALL Permits the jobs in all scheduling priority queues to be scheduled for execution.

jobname

Permits the particular job to be scheduled for execution. No further parameters are permitted.

Keyword Parameters:

OLD

Permits only jobs already in the scheduling priority queue defined in parameter 1 to be scheduled for execution.

NEW

Permits newly entered jobs placed in the scheduling priority queue defined in parameter 1 to be scheduled for execution.

If omitted, both old and new jobs are permitted to be scheduled for execution. (See 4.3.2.1.)

DDP

Permits only distributed data processing jobs to be scheduled for execution.

LOCAL

Permits only locally entered jobs to be scheduled for execution.

RBP

Permits only jobs entered remotely (i.e., from a remote batch terminal) to be scheduled for execution.

WKSTN

Permits only jobs either initiated from a workstation or jobs using the // OPTION ORI= or // OPTION MAS= job control statement, as defined in parameter 1, to be scheduled for execution.

NOTE:

All of the above keyword parameters may be interchanged.

UID=user-id

Permits all jobs associated with a particular workstation user-id to be scheduled in the scheduling priority queue defined in parameter 1. The user-id is a 1- to 6-character alphanumeric identification that does not start with Y.

HOST=host-id

Permits all jobs associated with a particular host to be scheduled in the priority queue defined in parameter 1. The host-id is one to four alphanumeric characters.

NOTES:

1. *The UID and HOST parameters may be interchanged, but no further parameters are permitted.*
2. *If a command is entered from the system console (local or remote), the HOST parameter should not be used; that site's host-id is used automatically.*
3. *The special user-id OPERATOR indicates that the command applies to console-initiated jobs.*
4. *Any command from the system console (local or remote) pertains only to jobs with a matching host-id, regardless of user-id.*

Examples:

1. Operator keyin:

```
BE MYJOB
```

Function requested:

The currently deferred job named MYJOB is permitted to be scheduled for execution.

2. Operator keyin:

```
BE JBQ,UID=WKSTA2
```

Function requested:

All currently deferred jobs in all scheduling priority queues associated with the workstation user WKSTA2 are permitted to be scheduled for execution.

3. Operator keyin:

```
BE JBQ,H,HOST=ABC
```

Function requested:

All currently deferred jobs in the high priority queue associated with host ABC are permitted to be scheduled for execution.

4.3.2.3. Deleting Jobs from Scheduling Priority Queues (DELETE)

Function:

The DELETE command permits the operator to delete jobs according to the command parameters specified. You can delete all jobs in all queues or in a specific queue; a specific job within a queue; a specific workstation user's jobs in all queues or a specific queue; or a specific host's jobs in all queues or a specific queue. Only those jobs residing in a scheduling priority queue, and thus waiting to begin execution, can be deleted.

The DELETE command cannot be issued from an enter stream.

Format:

```
DELETE△ ⎧JBQ,⎧PRE ⎫⎧⎡,⎧DDP  ⎫⎤                                 ⎫⎫[,LOG]
        ⎪    ⎪HIGH⎪⎪⎢ ⎨LOCAL⎬⎥                                 ⎪⎪
        ⎨    ⎨NOR ⎬⎨⎢ ⎪RBP  ⎪⎥                                 ⎬⎬
        ⎪    ⎩ALL ⎭⎪⎣ ⎩WKSTN⎭⎦                                 ⎪⎪
        ⎪          ⎩[,UID=user-id][,HOST=host-id]              ⎭⎪
        ⎩jobname                                                ⎭
```

Positional Parameter 1:

```
JBQ,⎧PRE ⎫
    ⎪HIGH⎪
    ⎨NOR ⎬
    ⎩ALL ⎭
```

Specifies the command applies to the jobs in a job scheduling priority queue, as further defined by PRE, HIGH, NOR, or ALL, where:

PRE
: Deletes the jobs in the preemptive scheduling priority queue.

HIGH
: Deletes the jobs in the high scheduling priority queue.

NOR
: Deletes the jobs in the normal scheduling priority queue.

ALL
: Deletes the jobs in all scheduling priority queues.

jobname
: Deletes a particular job from being scheduled for execution.

Keyword Parameters:

DDP
: Deletes only distributed data processing jobs.

LOCAL
: Deletes only locally entered jobs.

RBP
: Deletes only jobs entered remotely (i.e., from a remote batch terminal).

WKSTN
: Deletes only jobs either initiated from a workstation or jobs using the // OPTION ORI= or // OPTION MAS= job control statement.

NOTE:

All of the above keyword parameters may be interchanged.

UID=user-id
: Deletes all old jobs associated with a particular workstation user-id in the scheduling priority queue defined in parameter 1. Jobs subsequently entered with this user-id are not deleted. The user-id is a 1- to 6-character alphanumeric identification that does not start with Y.

HOST=host-id
: Deletes all old jobs associated with a particular host in the scheduling priority queue defined in parameter 1. Jobs subsequently entered with this host-id are not deleted. The host-id is one to four alphanumeric characters.

NOTES:

1. *The UID and HOST parameters may be interchanged, but no further parameters are permitted.*

2. *If a command is entered from the system console (local or remote), the HOST parameter should not be used; that site's host-id is used automatically.*

3. *The special user-id OPERATOR indicates that the command applies to console-initiated jobs.*

4. *Any command from the system console (local or remote) pertains only to jobs with a matching host-id regardless of user-id.*

LOG
: Specifies the job log is printed for all jobs deleted. If omitted, the log is not printed.

Examples:

1. Operator keyin:

```
DE JBQ,H,HOST=CDE
```

Function requested:

All jobs originating from host CDE and residing in the high scheduling priority queue are to be deleted.

2. Operator keyin:

```
DE JBQ,A,UID=WKSTA1,LOG
```

Function requested:

All jobs in all scheduling priority queues associated with the workstation user WKSTA1 are to be deleted and their logs are to be printed.

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. You can display all jobs in all queues or in a specific queue; a specific job within a queue; a specific workstation user's jobs in all queues or a specific queue; or a specific host's jobs in all queues or a specific queue. All requested jobs within the specified queue are displayed. Jobs in a deferred status (HOLD command) are displayed with parentheses around the job name. When all queues are requested, PRE is displayed first, followed by HIGH, and then NOR. If no jobs are found in the queue you request, a system output message is displayed stating that condition.

A system output message is displayed prior to the list of jobs. The message specifies:

- whether the request was for LOCAL, RBP, DDP, WKSTN, or for all jobs (QUEUED);
- the user-id and host-id;
- the priority of the queue display to follow; and
- whether a hold local (HL), hold remote (HR), hold workstation (HW), or hold DDP (HD) status is in effect for that queue.

Jobs are displayed in three different formats, depending on the parameters you specify on the DISPLAY command. In all cases, jobs in hold status are displayed in parentheses.

- Two jobs per line

  If the DDP parameter is specified, the host-id is included in the following format:

  ```
  jobname: host-id/user-id          jobname: host-id/user-id
  ```

- Three jobs per line

  This format is the most common and is used when you specify the HOST parameter or the WKSTN parameter, or you specify no parameters and issue the command from the system console (local or remote). The format is:

  ```
  jobname: user-id          jobname: user-id          jobname: user-id
  ```

- Five jobs per line

  If there can be no user-id (LOCAL or RBP) or if there is only one user-id to be selected (UID= or DI JBQ from workstation), the jobs are displayed in the following format:

  ```
  jobname      jobname      jobname      jobname      jobname
  ```

NOTE:

The DISPLAY command cannot be issued from an enter stream.

Format:

```
DISPLAY△JBQ ⎡ ⎧PRE ⎫⎤ ⎧⎡ ⎧DDP  ⎫⎤                               ⎫
            ⎢,⎨HIGH⎬⎥ ⎪⎢,⎨LOCAL⎬⎥                               ⎪
            ⎢ ⎪NOR ⎪⎥ ⎨⎢ ⎪RBP  ⎪⎥                               ⎬
            ⎣ ⎩ALL ⎭⎦ ⎪⎣ ⎩WKSTN⎭⎦                               ⎪
                      ⎩[,UID=user-id][,HOST=host-id]            ⎭
```

Positional Parameter 1:

```
JBQ ⎡ ⎧PRE ⎫⎤
    ⎢,⎨HIGH⎬⎥
    ⎢ ⎪NOR ⎪⎥
    ⎣ ⎩ALL ⎭⎦
```

Specifies the command applies to the jobs in a job scheduling priority queue, as further defined by PRE, HIGH, NOR, or ALL, where:

PRE
: Displays the jobs in the preemptive scheduling priority queue.

HIGH
: Displays the jobs in the high scheduling priority queue.

NOR
: Displays the jobs in the normal scheduling priority queue.

ALL
: Displays the jobs in all scheduling priority queues.

Keyword Parameters:

DDP
: Displays only distributed data processing jobs.

LOCAL
: Displays only locally entered jobs.

RBP
: Displays only jobs entered remotely (i.e., from a remote batch terminal).

WKSTN
: Displays only jobs either initiated from a workstation or jobs using the // OPTION ORI= or // OPTION MAS= job control statement.

NOTE:

All of the above keyword parameters may be interchanged.

UID=user-id
Displays all old jobs associated with a particular workstation user-id in the scheduling priority queue defined in parameter 1. Jobs subsequently entered with this user-id are not displayed. The user-id is a 1- to 6-character alphanumeric identification that does not start with Y.

HOST=host-id
Displays all old jobs associated with a particular host and residing in the scheduling priority queue defined in parameter 1. The host-id is one to four alphanumeric characters.

NOTES:

1. *The UID and HOST parameters may be interchanged but no further parameters are permitted.*
2. *If a command is entered from the system console (local or remote), the HOST parameter should not be used; that site's host-id is used automatically.*
3. *The special user-id OPERATOR indicates that the command applies to console-initiated jobs.*
4. *Any command from the system console (local or remote) pertains only to jobs with a matching host-id, regardless of user-id.*

Examples:

1. Operator keyin:

```
DI JBQ,N,LO,RBP
```

Function requested:

All locally and remotely entered jobs in the normal scheduling priority queue are to be displayed.

2. Operator keyin:

```
DI JBQ,UID=WKSTA2
```

Function requested:

All jobs in all scheduling priority queues that are associated with the workstation user WKSTA2 are to be displayed.

4.3.2.5. Changing a Job Scheduling Priority (CHANGE)

Function:

The CHANGE command changes the scheduling priority of a specific job. If you place a deferred job into a new (changed) scheduling priority queue, the job retains its deferred status. Likewise, if you place a job 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; it is the last examined for scheduling for execution in that queue.

Format:

```
CHANGE△jobname, ⎧PRE ⎫
                ⎨HIGH⎬
                ⎩NOR ⎭
```

Positional Parameter 1:

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

Positional Parameter 2:

`PRE`
Moves the job defined in positional parameter 1 into the PRE scheduling priority queue.

`HIGH`
Moves the job defined in positional parameter 1 into the HIGH scheduling priority queue.

`NOR`
Moves the job defined in positional parameter 1 into the NOR scheduling priority queue.

Example:

Operator keyin:

```
CH JOBABC,H
```

Function requested:

This moves JOBABC from a previously assigned (via the job control stream, command entry for running the job, or NOR by default) scheduling priority queue into the HIGH queue. The following message appears on your screen:

```
CH OF JOB JOBABC
```

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. You may enter the command at any time and job processing suspends immediately. If the job is between job steps, PAUSE takes 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 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:

jobname
Specifies the job to be reactivated after execution has 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. The switching priority level can be changed for the currently executing job step or for all subsequent job steps. If the priority is changed for the current job step only, any subsequent job step executes under the priority established for it (via // EXEC job control statement or default to the lowest level established at SYSGEN) unless changed by another SWITCH command. 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 than 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 (maximum 60 levels).

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 you change any job's switching priority, all tasks of that job retain the same switching priority relative to each other; therefore, if a job's task exceeds the upper or lower switching priority limit, all the job's tasks 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-priority-levels}[,ALL]
               {-number-of-priority-levels}
```

Positional Parameter 1:

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

Positional Parameter 2:

`+number-of-priority-levels`
Specifies the number of switching priority levels a job is raised.

`-number-of-priority-levels`
Specifies the number of switching priority levels a job is lowered.

Positional Parameter 3:

`ALL`
Indicates that the priority to which the job is switched will be in effect for all subsequent job steps. If omitted, the priority is in effect only for the current job step.

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. Canceling a Job in Progress (CANCEL)

Function:

The CANCEL command immediately halts all processing of a job or symbiont. You enter the CANCEL command at any time during job processing to immediately terminate the job step currently being executed, plus any subsequent job steps scheduled for the job. The job run library file for the job also is deleted.

Format:

$$\underline{\texttt{CA}}\texttt{NCEL}\triangle\left\{\begin{matrix}\texttt{jobname}\left[,\left\{\begin{matrix}\texttt{D}\\ \texttt{N}\end{matrix}\right\}\right]\\ \texttt{symbiont,S[,N]}\end{matrix}\right\}$$

Positional Parameter 1:

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

`symbiont`
Specifies the 2-character console command that called the symbiont to be terminated.

Positional Parameter 2:

`D`

Specifies that a dump is taken when the job terminates, regardless of the dump option specified in its job control stream.

N

Specifies that no dump is taken when the job terminates, regardless of the dump option specified in its job control stream.

S

Specifies that the name in positional parameter 1 is the name of a symbiont.

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

Positional Parameter 3:

N

Specifies that no dump is taken when the symbiont terminates.

If omitted, a symbiont dump is taken.

4.3.4.2. Stopping Execution of a Dump (END)

Function:

The END command terminates execution of a cancel or end-of-job dump for a particular job.

Format:

```
END△DUMP,jobname
```

Positional Parameter 1:

DUMP

Specifies that the execution of a dump is stopped.

Positional Parameter 2:

jobname

Specifies the name of the job whose cancel or end-of-job dump you want stopped.

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 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 an area of main storage;
- display information on active jobs and symbionts, 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 or date;
- set on I/O device down or up as required for normal maintenance or device malfunction;
- control software-detected hardware error logging;
- read the volume serial number of a mounted disk or tape volume;
- display the status of jobs in the system;
- dump contents of main storage;
- set the YDUMP file to an unlocked condition; and
- terminate system activity.

The commands enabling you to perform these functions are described in this subsection. Commands required to control interactive 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 displays selected areas of main storage at the system console. You usually enter the command when your system administrator needs a job address displayed for program debugging purposes.

Format:

```
DISPLAY△addr ⎡,⎧jobnumber⎫⎤
--           ⎣ ⎩Ø        ⎭⎦
```

Positional Parameter 1:

addr

Is a hexadecimal number used for a specific (absolute) main storage address or a job-relative main storage address. A job-relative address is identified by the job number for the job using it; otherwise, an absolute address is displayed.

Positional Parameter 2:

jobnumber
: Specifies the job number (1 through 7) of the job in main storage whose relative address is displayed.

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

After you enter DISPLAY to load the display symbiont (DI), the following output message appears on the system console:

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

where:

i
: Is a 1-digit hexadecimal message number (1 through F) consecutively assigned to output messages generated by the supervisor (in this case, in behalf of the display symbiont).

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 the job's 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 to display the next sequential main storage location.

N
: Is a message response to terminate the display symbiont.

NXT
: Is a reminder that you can display another nonconsecutive main storage location without recalling the display symbiont by responding to this output message with the solicited input message:

```
▷ Øi△addr ┌ , ┌ jobnumber ┐ ┐
          └   └ Ø         ┘ ┘
```

where:

i
: Is the 1-digit hexadecimal message number (1 through F) of the display symbiont output message you are responding to.

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

jobnumber
: Identifies the job number (1 through 7) of the next job whose relative address is to be displayed.

Ø

Specifies the address entered in the input message is an absolute address.

When there are no more addresses to display, terminate the display symbiont by responding to the last display message with the solicited input message:

```
▷ Øi△N
```

4.4.2. Displaying System Information (MIX)

Function:

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

Format:

```
MIX△ ⎧DA ⎡,⎧jobname      ⎫⎤⎫
     ⎪   ⎣ ⎩symbiont-name⎭⎦⎪
     ⎪VI                   ⎪
     ⎪SQ                   ⎪
     ⎪SI                   ⎪
     ⎨DS[,did]             ⎬
     ⎪SC                   ⎪
     ⎪FR                   ⎪
     ⎪MM                   ⎪
     ⎩EN                   ⎭
```

Positional Parameter 1:

DA

Displays the following information on the system console screen for the job or symbiont you specify (positional parameter 2) or for all active jobs and symbionts when you omit positional parameter 2:

- Job name or symbiont name
- Job slot number; 0 for symbiont
- Priority
- Allocated device numbers
- Allocated device types
- Volume serial number or user identification

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, system reader (RDR) device address, spool (SPL) device address, and total available main storage.

DS

Displays the following information on the system console screen for the device you specify in positional parameter 2, or for all devices in the system if you omit positional parameter 2. Use MI DS to determine whether a diskette drive is available when it requires maintenance; if the drive is in use, an I/O error occurs if its door is opened.

- The device address of each device
- Whether the device is up or not (Y or N)
- 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

SC

Displays the name, address, and decimal size of each shared code module currently residing in main storage.

FR

Displays the address and decimal size of all regions currently unused in main storage.

MM

Displays the type, name, address, and decimal size of every main storage region. The following types are displayed:

■	FREE	Free region
■	JOB	Job region
■	JOB SCHD	Job scheduler
■	SYMBIONT	Symbiont (e.g., RUN processor, ICAM)
■	RESERVED	Region reserved for a job just scheduled
■	BUF POOL	Dynamic buffer pool. These memory pools are internally subdivided to provide dynamically allocated main storage for system software.
■	SHR CODE	Shared code. When shared code occupies its own main storage region, you display it by using either the SC or the MM parameters. When shared code is part of a dynamic buffer pool, you display it by using the SC parameter only.
■	DISABLED	Not available. The region is either temporarily or permanently disabled. A region is temporarily disabled when it is in the process of being changed from one type to another (e.g., from SYM to FREE). This temporary condition exists for no more than a few seconds. A region is permanently disabled when a system error, such as a parity error, prevents the termination and clearing out of a region.

- DOWN   Down. The region is set down as the result of either an operator command or too many recoverable main storage errors occurring in that region.

EN

Terminates the MIX function currently in progress.

Positional Parameter 2:

jobname

Used with the DA parameter to identify a job name for which information is desired. If you omit this parameter, information for all active jobs and symbionts is displayed.

symbiont-name

Used with the DA parameter to identify a symbiont name for which information is desired. If omitted, information for all active jobs and symbionts is displayed.

did

Used with the DS parameter to obtain status information for a device or group of devices. One, two, or three characters may be used. Key in three characters to specify the exact device address of the device to be displayed. If you key in two characters, the group of devices whose device addresses start with those two characters is displayed. If you key in one character, information is displayed about all devices whose addresses begin with that character. If you omit this parameter, 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,4Ø	Displays any configured devices from 400 to 409
MIX DS,4ØØ	Displays device 400 only

4.4.3. 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.

4.4.4. Setting Simulated Day Clock (SET CLOCK)

Function:

The SET CLOCK command resets the time of day in the system-simulated day clock: for example, to change from 24:00:00 to 00:00:00 at midnight.

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 setting the simulated day clock to the time specified in positional parameter 2.

Positional Parameter 2:

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

hh
Specifies hours (00 through 99).

mm
Specifies minutes (00 through 59).

ss
Specifies seconds (00 through 59).

4.4.5. Setting Date Field (SET DATE)

Function:

The SET DATE command resets the calendar date in the system information block date field and resets the job date for every job currently in main storage (except those jobs containing a // SET DATE job control statement).

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 changing the calendar date in the system information block date field and in the preambles for current jobs to the date specified in positional parameter 2.

Positional Parameter 2:

yy/mm/dd
Specifies the date to be used for the calendar date in the system information block date field and job preambles, 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.

4.4.6. 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 $Y$ELOG 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, with the pub-id specified.

Format:

$$\underline{\text{SE}}\text{T}\triangle\underline{\text{EL}}\text{OG},\left\{\begin{array}{l}\left\{\begin{array}{l}\text{ON}\\ \underline{\text{O}}\text{FF}\end{array}\right\}\left[,\left\{\begin{array}{l}\underline{\text{AL}}\text{L}\\ \underline{\text{CO}}\text{MM}\\ \underline{\text{MC}}\text{HK}\\ \text{IO}\\ \underline{\text{TE}}\text{RM}\\ \text{did}\end{array}\right\}\right]\\ \\ \underline{\text{RE}}\text{SET},\left\{\begin{array}{l}\underline{\text{AL}}\text{L}\\ \text{did}\end{array}\right\}\\ \text{mask,did}\end{array}\right\}$$

Positional Parameter 1:

ELOG
Sets the PIOCS error logging processor to some condition.

Positional Parameter 2:

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

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

RESET
Resets error sense bytes to their original IPL specification. The did 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 whether an I/O error can be logged. Mask can only be used with the did 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 did 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 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 all device sense bytes are to be reset to their original specification.

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

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

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

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

did

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

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

Operator Considerations:

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, you should run the system-supplied ONUERL job by 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 $Y$RUN 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 $Y$ESUM 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.

```
//△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 $Y$ELOG; 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.)

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

```
//△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 $Y$ESUM.

The console messages that may be displayed during execution of the ONUERL program follow. An explanation and operator action, where required, are also provided.

```
ONUERL - OS/3 ERROR-LOG EDITOR. VER nn/nn
```

This message appears at start of ONUERL program execution, where nn/nn is the version number loaded.

```
ONUERL - ERROR READING PARAM. RØ=error-code. TERMINATE? (Y,N)
```

An error has been detected in reading // PARAM cards from the job control stream, where error-code specifies the error encountered.

Respond Y to terminate. Respond N to ignore the parameter.

```
INVALID ENTRY card-contents IGNORE? (Y, N)
```

A // PARAM card has the wrong format, where card-contents is a display of the // PARAM card in error.

Respond Y if the default condition is to be used; otherwise, respond N if ONUERL is to be retried with a corrected // PARAM card.

```
ONUERL - $Y$ELOG IS EMPTY.
```

$Y$ELOG contains no new data since its last reading. ONUERL is terminated.

```
ONUERL - ACCESS ERROR ON $Y$ELOG. RØ=error-code
```

An error has been detected in attempting to access $Y$ELOG, where error-code specifies the error encountered. ONUERL is terminated.

```
FIRST RECORD NOT HISTORICAL RECORD. JOB TERMINATED.
```

The beginning of $Y$ELOG cannot be found. ONUERL is terminated.

```
ONUERL - INVALID DATA IN $Y$ELOG. JOB CANCELLED.
```

The data in $Y$ELOG is meaningless. ONUERL is terminated.

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.

If the ELOG file becomes full, ELOG asks whether you want 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 by using the solicited message format:

$$\emptyset\ i\triangle\begin{Bmatrix}0\\W\end{Bmatrix}$$

where:

i
: Identifies the console message being answered.

0
: 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 $Y$ELOG 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 EDIT   VERSION nn/nn_________________________                  yy/mm/dd hh.mm  PAGE    1

CUSTOMER ID: xxxxxxxxx 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

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

CHDV

The channel, subchannel, and device number representing the did for the identified device.

FEATURE BYTES

A hexadecimal identification of the features installed on the device.

PHYSICAL ADDR

The starting address in main storage for the user job running at time of error report.

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 at time of error.

VSN

The volume serial number of the volume mounted on the device at time of error.

RT

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

R/U

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.

OPR

Operator response to error message displayed on system console screen, if any. Possible response is U (unrecovered), I (ignore), C (cancel), or R (retry).

I/O's

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

BCW and CCW fields

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 ADDR

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.

DV/SC STATS

The device and subchannel status recognized when the error recovery sequence was completed.

EMSK

The error log mask in effect at time of error.

SNS 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, XPCT 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-Ø

Contents of supervisor.

SUP R-1

Registers 0 and 1.

4.4.7. Setting Up Physical Unit Block (SET IO)

Function:

The SET IO command allows you to set specific bits in the physical unit blocks (PUB) that define operational characteristics and assignments of I/O devices. There is one physical unit block comprising a 3- or 4-character did (device address) for each physical device in your system. You must set all devices or subsystems *DOWN* before attempting operation on the device, such as forms loading or changing ribbon; or in case of malfunction, before turning the device off. This is required if the processor is to continue operation with other peripheral devices while the subject device undergoes isolated operations. Before performing a procedure or turning power off for the device, key in SET IO,did,DOWN. After the offline procedure completes or after turning on power independently of the processor, key in SET IO,did,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:

```
SET△IO,did,⎧AV            ⎫
            ⎪DOWN          ⎪
            ⎪EON           ⎪
            ⎪EOF           ⎪
            ⎪FEA,type-code ⎪
            ⎪HOME          ⎪
            ⎨NA            ⎬
            ⎪NOSHARE       ⎪
            ⎪RDR           ⎪
            ⎪SHARE         ⎪
            ⎪TYPE,type-code⎪
            ⎩UP            ⎭
```

Positional Parameter 1:

IO
: Specifies the change is made in the device address specified by positional parameter 2.

Positional Parameter 2:

did
: Specifies a 1- to 4-character numeric field identifying the device address to be changed (device addresses are usually attached at a visible location on the device cabinet) as follows:

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 the device identified by positional parameter 2 is recognized by the system and available for assignment to user jobs.

DOWN

Specifies the device identified in positional parameter 2 is not to be considered available for assignment to user jobs, although the device is recognized by the system.

EON

Turns on error logging for the specified device.

EOF

Turns off error logging for the specified device.

FEA

Modifies 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 the device identified by positional parameter 2 is not recognized by the system and is not available for assignment to user jobs.

NOSHARE

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

RDR

Assigns the new did 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

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

UP

Specifies the device identified in positional parameter 2 is considered available for assignment to user jobs. The device remains recognized by the system.

Positional Parameter 4:

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

4.4.8. Controlling the Seek Separation Feature (SET SEEKSEP)

The SET SEEKSEP command activates the seek separation feature. This feature allows you to access more than one disk device during input/output operations by separating seek time from read/write time.

Format:

```
SET△SEEKSEP△{OFF} [,{ALL}]
            {ON }  [ {did}]
```

Positional Parameter 1:

`OFF`
Disables previously enabled seek separation feature.

`ON`
Enables the seek separation feature.

Positional Parameter 2:

`ALL`
Sets the seek separation feature for all disks.

`did`
Sets the seek separation feature for this device only.

If this parameter is omitted, the feature is set for all devices.

4.4.9. Reading a Mounted Volume Serial Number (AVR)

Function:

The AVR command reads the volume serial number of a premounted prepped disk pack or magnetic tape volume and stores 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△did[,did][,did]
```

Positional Parameters 1 through 3:

`did`
Specifies the device addresses for the volumes to be recognized.

4.4.10. Displaying Job Status (DISPLAY JS)

Function:

The DISPLAY JS command displays the status of jobs in the system at the system console. You can display the status of a specific job or all jobs in main storage, a specific job in a scheduling priority queue, or a job being processed by the RUN or OCL processor. The display includes the job name along with the CPU time used when the job is under execution, or the reason why the job is not executing (such as waiting for IO, waiting for mount message, under a pause), or the scheduling priority queue in which the job resides.

Format:

```
DISPLAY△JS[,jobname]
```

Positional Parameter 1:

```
JS
```

Displays job status at the system console.

Positional Parameter 2:

```
jobname
```

Displays the status of the job name specified. If omitted, the status of all jobs in main storage is displayed.

Example:

Operator keyin:

```
DI JS,MYJOB
```

Function requested:

Produce a display of status information about a job named MYJOB.

Typical informational messages:

The following are examples of the job status informational messages produced by the DISPLAY JS command. The examples show what information DI JS might display if it were entered against MYJOB as the job proceeds through the various steps in job processing.

For the message:

```
MYJOB IN STEP Ø1(LNKEDTØØ)-PRI=1Ø CPU-TIME=ØØ:Ø1:43.874
```

MYJOB is active in its first step, performing linkage editing. The CPU TIME portion of the display indicates the linkage editor had control of the CPU for 1 minute, 43 seconds, and 874 milliseconds. If the job is proceeding, you can reenter DI JS for MYJOB and see an increase in the CPU TIME figure.

For the message:

```
MYJOB IN STEP Ø2(LIBSØØØØ)-WAITING FOR I/O #ØØØØ5736
```

MYJOB is in its second step, executing the librarian. Currently, the 5736th I/O operation of this step is being performed. If you reenter DI JS, you may see the I/O number increase. If MYJOB remains at #00005736, it might be stuck, requiring your intervention.

For the message:

```
MYJOB IN STEP Ø3 - IN STEP PROCESSOR
```

MYJOB is between job steps. Step 03 either has just completed or is about to start.

For the message:

```
MYJOB NOT YET SCHEDULED-INSUFFICIENT MAIN STORAGE
```

MYJOB is not executing; it is placed on a job queue but isn't scheduled for execution because not enough main storage is available.

4.4.11. Dumping the Contents of Main Storage (SYSDUMP)

Function:

The SYSDUMP command is used to dump the entire contents of main storage to the $Y$DUMP file on SYSRES. Use the SYSDUMP command whenever a system dump is required without supervisor reloading (no re-IPL is required). After the contents of main storage are dumped, the job SYSDMPnn (where nn is a unique number assigned by the system) is automatically initiated to print the $Y$DUMP file. The $Y$DUMP file locks until the SYSDMPnn job completes to prevent other system functions (caused, for example, by an //OPTION SYSDUMP statement or by system errors) from also using the file. (For further details, see the current version of the dump analysis manual, UP-8837.)

If you don't want the dump, enter NONE when the SYSDMPnn job asks what type of dump should be printed. If you delete the SYSDMPnn job from the job queue or cancel it before it sends you the SD01 output message, you must enter the SET SY command (4.4.11) to unlock the $Y$DUMP file.

You can't use the system console keyboard until the main storage contents are completely written (only a few seconds). After the keyboard unlocks, you can resume system activity without impairing the integrity of your dump.

Format:

```
SYSDUMP
```

There are no positional parameters.

NOTE:

When a re-IPL is required, the dump should be obtained by pressing SYSTEM RESET and RUN on the maintenance panel to preserve low-order main storage.

4.4.12. Setting the $Y$DUMP File to Unlocked Condition (SET SY)

Function:

The SET SY command unlocks the $Y$DUMP file after a SY command (4.4.10) or system error has locked it. You use the SET SY command to unlock $Y$DUMP if you previously entered the SYSDUMP command and deleted the SYSDMPnn job from the job queue or canceled it before it displayed the SD01 output message.

Format:

```
SET△SY,LOFF
```

Positional Parameter 1:

SY
Specifies the $Y$DUMP file condition is to be set.

Positional Parameter 2:

LOFF
Specifies the $Y$DUMP file is to be unlocked.

4.4.13. Terminating System Activity (SHUTDOWN)

Function:

The SHUTDOWN command terminates system activity in an orderly manner. The spooler and job scheduler will not start any new files or jobs. Interactive services will not start any new functions and will terminate when its current activity ceases.

Format:

```
SHUTDOWN△[DDP]
```

Positional Parameter 1:

DDP
Specifies that only distributed data processing will be terminated when its activity ceases.

NOTE:

DDP is shut down by either using the SHUTDOWN DDP command or automatically after 10 minutes of no DDP activity. During DDP shutdown processing, all DDP activity is logged into the interactive services log file. The activity information remains in the log file and will be printed only when interactive services is shut down by using the IS SHUTDOWN message (5.3.4).

4.4.14. Initiating Transient Work Area Feature (TW)

Function:

The TW command initiates the transient work area feature. This feature can improve system performance. The most recently used transients are stored in the transient work area in main storage. Therefore, an I/O delay can be avoided by satisfying the transient load request directly from the work area rather than from SYSRES.

Format:

```
TWΔ[SZ=nn]
```

Positional Parameter 1:

```
SZ=nn
```

Specifies the number (nn) of 1024-byte blocks for the work area. nn is a 2-digit value in the range of 01 to 99.

If omitted, 65K is allocated for the work area by default.

You can initiate the transient work area whenever sufficient main storage is available to satisfy the work area space requirements. However, we recommend that you initiate TW immediately after the IPL procedure in order to avoid possible main storage fragmentation.

Once the transient work area is activated, the message "TRANSIENT WORK AREA IS INITIALIZED" appears. If you receive this message after the IPL procedure, it indicates that the work area with the default size of 65K has already been allocated during system initialization.

To change the work area size, you must terminate the transient work area and then key in the TW command with a new value. To terminate the transient work area, key in:

```
ØØΔTWΔEOJ
```

The message "TRANSIENT WORK AREA TERMINATED" then appears.

5. Interactive Services

5.1. GENERAL

The system operator uses interactive services with an extended set of commands and messages to control the interactive system environment. (For a description of all interactive services commands, see the current version of the interactive services commands and facilities user guide/programmer reference, UP-8845.) These commands and messages enable you to exercise control over the interactive OS/3 operating system, all jobs within the system, all workstation users (local locations), and all terminal users (local and remote locations).

5.2. INTERACTIVE SERVICES COMMANDS

When interactivity is included in your system at SYSGEN time, the interactive services components are loaded automatically whenever required for the system operator and workstation users. Provided ICAM and the global user service task (GUST) are ready, a workstation user's interactive entry or a command from the operator loads interactive services for terminal users. Interactive services commands are used to:

- send messages to workstation and terminal users;
- ask questions of workstation and terminal users;
- display the volumes in use and the status of active workstations, terminals, jobs, and functions;
- display status of system resources in use and available; and
- load interactive services.

5.2.1. Sending Messages to Users (TELL)

Function:

The TELL command sends a message not requiring a response to a specific workstation or terminal user or to all users.

Format:

```
TELL△{ALL    },'text'
     {user-id}
```

Positional Parameter 1:

ALL
Displays the message on all active user screens.

user-id
Specifies the 1- to 6-character alphanumeric identification of the user to receive the message display.

Positional Parameter 2:

text
Is the text of the message to be sent. The text may be a maximum of 48 characters long. Text must be preceded and followed by apostrophes.

5.2.2. Asking Questions of Users (ASK)

Function:

The ASK command sends a message that requires a response to a specific workstation or terminal user. The command displays your question to the specified user, accepts the reply, and returns the reply to the console screen.

Format:

```
ASK△user-id,'text'
```

Positional Parameter 1:

user-id
Specifies the 1- to 6-character alphanumeric identification of the local or remote user to receive the message display.

Positional Parameter 2:

text
Is the text of the message to be sent. The text may be a maximum of 48 characters long. Text must be preceded and followed by apostrophes.

5.2.3. Displaying System Status (STATUS)

Function:

The STATUS command displays the volumes currently in use; the status of active workstations, terminals, jobs, and functions; and the status of system resources in use and available.

Format:

```
STATUS△ ⎡⎧ JOBS      ⎫⎤
        ⎢⎪ FUNCTIONS ⎪⎥
        ⎢⎨ RESOURCES ⎬⎥
        ⎢⎪ TERMINALS ⎪⎥
        ⎣⎩ VOLUMES   ⎭⎦
```

Positional Parameter 1:

JOBS
: Displays a listing of the jobs and symbionts currently active in the system. The list includes the job or symbiont name and the amount of storage and CPU time used by each, information on what program each job is executing and its job step number, the job slot number for the job, and the master user-id for the job or symbiont. A summary line displays the amount of free main storage in the system and the largest contiguous region.

FUNCTIONS
: Displays a listing of all active commands and tasks initiated by users.

RESOURCES
: Displays a listing of the amount of storage being used by all users; the number of interactive, batch, and batch with interactive commands (ENTER) tasks; the number of active jobs and configured job slots; and the amount of total system storage.

TERMINALS
: Displays a listing of all currently active (logged on) workstations and terminals and the associated user-id.

VOLUMES
: Displays a listing of the tape, disk, and diskette volumes currently mounted on the system.

5.2.4. Starting Interactive Services (IS)

Function:

Normally, the interactive services facility is started automatically for the system operator and all workstation users, whenever needed. However, there are occasions when you must manually load interactive services via the IS command. Enter the IS command with the REMOTE START parameter to manually start interactive services when ICAM and GUST are ready and interactive services are needed for terminals but no workstation entry has automatically loaded them. When ICAM and GUST are ready and the interactive services facility is active (already loaded) before being needed for terminals, use the IS REMOTE unsolicited message (5.3.5) to start it. If, however, ICAM and GUST are not ready, terminal users cannot initiate interactive sessions. (See Section 6 for the operator procedures on loading ICAM and running the global user service task.)

You also use the IS command to manually start interactive services after a completed shutdown (termination). When you terminate interactive services via the SHUTDOWN message (5.3.4), the interactive services facility is placed in a closed condition and is unable to start automatically when needed. After a completed shutdown, you enter the IS command to remove the closed condition and restart interactive services.

Format:

```
IS△[REMOTE△START]
```

Positional Parameter:

REMOTE START
: Loads interactive services for terminals when no workstation entry has automatically loaded them.

5.3. INTERACTIVE SERVICES MESSAGES

You use unsolicited messages provided by interactive services for additional control over the interactive environment. These messages are used to:

- terminate workstation and terminal user sessions;
- restrict and release new workstation and terminal user sessions;
- terminate interactive services; and
- control interactive services for terminals.

5.3.1. Terminating User Sessions (REMOVE Message)

Function:

The REMOVE unsolicited message terminates a specific workstation or terminal user session, all user sessions, or a single command for a specific task. Following the termination, a cancellation message is displayed on the terminated user's screen.

Message Format:

```
ØØ△IS△REMOVE△⎧user-id⎫
             ⎨ALL    ⎬
             ⎩task-id⎭
```

where:

user-id
Specifies the particular user whose session is to terminate. User-ids can be determined by using the STATUS TERMINAL command. All interactive functions for the user-id specified are terminated and the user is logged off. If the user is running an interactive session as a batch job (via ENTER), the user is logged off; however, the session is not affected. A message is displayed on the console screen to indicate when the user cannot be logged off.

ALL
Specifies all user sessions and batch runs are to terminate. All interactive functions for all users are terminated and the users are logged off. Users running interactive sessions as batch jobs (via ENTER) are logged off. A message indicating which users cannot be logged off is displayed on the console screen.

task-id
Specifies the particular task under which a command is to terminate. Task-ids can be determined by using the STATUS FUNCTION command.

NOTE:

When distributed data processing (DDP) is present in the system, the IS REMOVE ALL command will have no effect on the task with a user-id of DDP△MR and a command identification of DDP△MAIL. Also, this task cannot be removed by the user-id or the task-id. This is a DDP task that has several responsibilities including DDP termination processing. This task is removed when DDP is shut down (using the SHUTDOWN DDP command or automatically after 10 minutes of no DDP activity). During DDP shutdown processing, all DDP activity is logged into the interactive services log file. The activity information remains in the log file and will be printed only when interactive services is shut down (using the IS SHUTDOWN message).

5.3.2. Restricting New User Sessions (CLOSE Message)

Function:

The CLOSE unsolicited message restricts any new workstation or terminal user sessions from starting. Currently active sessions are not affected.

Message Format:

```
ØØ△IS△CLOSE
```

5.3.3. Releasing New User Session Restrictions (OPEN Message)

Function:

The OPEN unsolicited message removes a previously entered CLOSE or reverses an incomplete SHUTDOWN to permit new workstation and terminal user sessions to start.

Message Format:

```
ØØ△IS△OPEN
```

5.3.4. Terminating Interactive Services (SHUTDOWN Message)

Function:

The SHUTDOWN unsolicited message terminates interactive services after all sessions have completed. If no sessions are active when you enter the command, it takes effect immediately. Otherwise, interactive services do not terminate until all active sessions complete. An OPEN message can be used to stop a shutdown in progress (incomplete). After the interactive services facility is completely shut down, you can restart it only via the IS command (5.2.4); it is not automatically loaded for workstation users via an interactive services command entry.

Format:

```
ØØ△IS△SHUTDOWN
```

5.3.5. Controlling Interactive Services for Terminals (REMOTE Message)

Function:

The REMOTE unsolicited message provides additional control over interactive services for terminals. You must load ICAM and run the global user service task (GUST) before you start interactive services for terminals. (See Section 6 for a description of the ICAM and GUST operator procedures.)

Format:

```
ØØ△IS△REMOTE△⎧START   ⎫
             ⎨SHUTDOWN⎬
             ⎩CANCEL  ⎭
```

Parameters:

START

Starts interactive services for terminals after ICAM and GUST are ready. You use this parameter when interactive services are already active in the system (via a prior IS command or an automatic start by a workstation user). If you enter the REMOTE START message with ICAM and GUST not ready, an error message is displayed on the console screen.

SHUTDOWN

Terminates the interactive services facility for terminals as soon as the last terminal session completes. (Workstation users are not affected.)

CANCEL

Immediately terminates the interactive services facility for terminals. Terminal users are not logged off.

6. Integrated Communications Access Method (ICAM) Procedures

6.1. GENERAL

The integrated communications access method (ICAM) is an extension of the supervisor (a symbiont) that handles data communications tasks. At system generation time, the ICAM symbionts are tailored to each user's requirements. One or more ICAM symbionts can be configured during SYSGEN to satisfy specific communications network requirements; or a single ICAM symbiont can be configured to satisfy all communications requirements. Each symbiont may contain multiple network definitions (CCAs), and each CCA can handle one or more communications lines. The operator must load the appropriate ICAM symbiont before the programs requiring it can execute or before interactive services can start for terminals. In addition, when interactive services or global networks are required, the operator must initiate the running of the global user service task (GUST) after loading ICAM. The ICAM symbiont remains in main storage until the last program or interactive services terminal session is completed and GUST is shut down. Then ICAM shuts itself down unless the system operator loaded ICAM with a KEEP command, in which case ICAM must be terminated with a CANCEL command.

The following subsections describe how to load ICAM, change the ICAM name established during SYSGEN, run and terminate the global user service task, and use messages to control the active communications environment.

6.2. ICAM OPERATOR COMMANDS

6.2.1. Loading the ICAM Symbiont (Cn/Mn)

Function:

The Cn/Mn command loads the ICAM symbionts. The symbionts are named C1-C9 or M1–M9 and are normally assigned during SYSGEN.

Format:

```
{Cn}△[KEEP]
{Mn}
```

where:

```
{Cn}
{Mn}
```

Specifies the name of the required ICAM symbiont, where n is 1 to 9. It is the name specified on the MCPNAME parameter in the COMMCT phase of system generation.

KEEP

Keeps the ICAM symbiont loaded until cancelled by the system operator or if ICAM suffers an unrecoverable error.

When ICAM is successfully loaded, the output message

```
ICAM READY
```

is displayed on the console screen.

6.2.2. Changing the ICAM Name (SET IC)

Function:

The SET IC command changes the ICAM symbiont (C1-C9, M1-M9) that is loaded if remote batch output has output ready and ICAM is not loaded. By specifying C?, you are asked to supply the symbiont name to be used the next time output is ready and ICAM is not loaded.

Format:

```
SET△IC,{Cn}
       {Mn}
       {C?}
```

Positional Parameter 1:

IC

Specifies the SPOOLICAM SYSGEN parameter is to change.

Positional Parameter 2:

```
{Cn}
{Mn}
{C?}
```

Specifies the ICAM name to use (C1-C9 or M1-M9) or specifies that you be asked for a valid ICAM name if ICAM is not currently loaded to send remote batch output (C?).

6.3. INITIALIZING AND TERMINATING THE GLOBAL USER SERVICE TASK

You must initialize the global user service task before starting interactive services for terminals or before executing user programs requiring global networks. You initiate the running of the job that executes the global user service task program ML$$GI through a system console command entry. See your system administrator for the name of the GUST job to initiate. When global network processing is no longer required for interactive services at terminals or for user programs, you enter an unsolicited message to shut GUST down.

6.3.1. Running the Global User Service Task Job

Function:

The RUN command is entered with the global user service task job name to initiate ML$$GI program execution.

Format:

```
RUN△jobname
```

where:

jobname
Is the name of the prefiled job control stream to run for executing the global user service task ML$$GI program.

6.3.2. ML$$GI Program Operator Messages

When you execute ML$$GI, it sends the following messages to the system console to obtain the information it requires to initialize the global network. Respond to each message as indicated. (Only those messages pertaining to GUST initialization are presented here. See the current version of the system messages programmer/operator reference, UP-8076, for all global user service task messages.)

For the message:

```
MC#420 ENTER NETREQ: CCANAME, PASSWORD, LINE REQUEST
```

Enter the name, password, and line information associated with the global network to be initialized. When multinode global networks are used, ML$$GI automatically requests all virtual lines.

where:

CCANAME
Is the name of the global network to be initialized. This name must be the same as the label of the CCA macroinstruction that begins the network definition for the global network.

PASSWORD
Is a 1- to 8-character password.

LINE REQUEST
Specifies the lines for which line requests are to be issued. Response may be one of the following:

ALL
Issue line requests for all lines defined in the global network.

line-1,...,line-n
Specifies the lines for which line requests are to be issued. Lines specified must be identical to those specified in label fields of related LINE macroinstructions.

*
Issue line requests for the lines specified in forthcoming message MC#421.

For the message:

```
MC#421 ENTER LNEREQS: LINE-1, LINE-2,...* OR BLANK
```

Enter the lines for which line requests are to be issued. This message is displayed only when you respond to the MC#420 message with an asterisk for the line request.

where:

`LINE-1, LINE-2,...line-n`
Specifies the lines for which line requests are to be issued. Lines specified must be identical to those specified in label fields of related LINE macroinstructions.

`* OR BLANK`
Redisplays this message to permit additional line requests to be entered.

For the message:

```
MC#433 RESTART DESIRED FOR NETWORK ____? ANS Y OR N
```

Enter Y or N to set the RESTART flag for use when a NETREQ macroinstruction is issued.

where:

`Y`
Sets a restart flag during global network initialization to recover previous network messages from existing disk files.

`N`
Specifies previous network messages are not recovered during global network initialization.

For the message:

```
MC#430 GUST ACTIVE FOR CCA network-name
```

Specifies the global user service task (GUST) initialization is complete for the network named. No response is required.

6.3.3. Terminating the Global User Service Task

Function:

The following unsolicited message ends global network processing by cancelling the global user service task job.

Format:

```
ØØ△⎧Cn⎫△GU△S,network-name,jj
   ⎩Mn⎭
```

where:

```
{Cn}
{Mn}
```

Specifies the currently loaded ICAM symbiont name (C1-C9 or M1-M9).

`GU`

Specifies this message is for GUST.

`S`

Specifies a shutdown is required.

`network-name`

Is the 4-character name of the active global network. This name must be the same as the label of the CCA macroinstruction in the global network definition. If the name is less than four characters, pad it with blanks on the right.

`jj`

Is the GUST job number (01-07).

If the shutdown request is accepted, the following message is displayed when the global user service task job is cancelled:

```
MC#4Ø1 GUST SHUTDOWN COMPLETE
```

NOTE:

To resume communications processing with global networks or to start interactive services for terminals, the operator procedures for loading ICAM and initializing the global user service task must be repeated. You should never cancel the GUST job by means of the CANCEL operator command.

6.4. ICAM OPERATOR MESSAGES

On occasion, you enter unsolicited messages to ICAM to facilitate processing. These messages have the following format:

```
ØØ△{Cn}△cc△f,{xxxx},jj
   {Mn}        {xx  }
```

where:

```
{Cn}
{Mn}
```

Specifies the name of the required ICAM symbiont.

`cc`

Is a 2-character command code.

`f`

Is a 1-character facility type (L=line, P=port, T=terminal).

xxxx

Is a 1- to 4-character name of a line or terminal as defined in the label field of a LINE or TERM macroinstruction. If the name is less than four characters, pad it with blanks on the right.

xx

Is either a 2-character port number on the communications adapter or the SLCA number.

jj

Is a 2-digit job number (01-07).

Table 6-1 lists the selected-occasion unsolicited messages ICAM provides to the operator.

A typical ICAM response to an unsolicited message is:

```
MC#9Ø LINE xxxx MARKED UP, USER jj
```

Table 6—1. ICAM Unsolicited Messages

Messages	Description
ØØ △$\begin{Bmatrix}\text{Cn}\\\text{Mn}\end{Bmatrix}$△UP△L,xxxx,jj	Marks line specified as available (up)
ØØ △$\begin{Bmatrix}\text{Cn}\\\text{Mn}\end{Bmatrix}$△DO△L,xxxx,jj	Marks line specified as unavailable (down)
ØØ △$\begin{Bmatrix}\text{Cn}\\\text{Mn}\end{Bmatrix}$△UP△T,xxxx,jj	Marks terminal specified as available (up)
ØØ △$\begin{Bmatrix}\text{Cn}\\\text{Mn}\end{Bmatrix}$△DO△T,xxxx,jj	Marks terminal specified as unavailable (down)
ØØ △$\begin{Bmatrix}\text{Cn}\\\text{Mn}\end{Bmatrix}$△UP△P,xx,jj	Marks port specified as available (up)
ØØ △$\begin{Bmatrix}\text{Cn}\\\text{Mn}\end{Bmatrix}$△DO△P,xx,jj	Marks port specified as unavailable (down)
ØØ △$\begin{Bmatrix}\text{Cn}\\\text{Mn}\end{Bmatrix}$△CN△L,xxxx,jj	Notifies ICAM that dialing is completed on switched line specified
ØØ △$\begin{Bmatrix}\text{Cn}\\\text{Mn}\end{Bmatrix}$△CN△L,ALL,jj	Notifies ICAM that dialing is completed on all switched lines

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
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

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

Function Code	Function Performed
Tape Functions (cont)	
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
DDR	Printing a disk in reblocked format
VTP	Printing the volume table of contents of a disk
SVT	Printing short format VTOC file
AVX	Displaying available disk extents on console screen
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 SU/TU command loads the system utility symbiont. SU and TU can be used interchangeably for all functions. However, we recommend you use the TU symbiont for tape operations since TU increases the buffer size for all selector channel tapes from 8189 to 32,767 bytes.

You can include the required function 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. (For a description of all spooling commands, see the current version of spooling and job accounting concepts and facilities, UP-8869.) When you enter the command alone to load the symbiont, you enter the function as a solicited message. After the symbiont is loaded, you control it with solicited messages. You use unsolicited messages only to terminate the symbiont or current symbiont functions on certain occasions.

NOTES:

1. *The TU symbiont should be run only when no other job will be starting up or performing multiple steps that will allocate the same volumes. This could cause the system utility to terminate abnormally.*
2. *The SU/TU command cannot be issued from an enter stream.*

Format:

```
{SU} △ [function-code [,{Y}]]
{TU}                    {N}
                        {R}
                        {H}
```

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.

You display all possible function codes recognized by either symbiont 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`
Spools the system utility output.

`N`
Specifies the system utility output is not spooled.

`R`
Retains system utility output in HOLD condition in the spool file after it is printed or punched. The retained output is unavailable for additional processing until released via the BEGIN SPL command.

`H`
Places system utility output in HOLD condition in the spool file before it is printed or punched. The output must be released via the BEGIN SPL command to permit processing.

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.

Operator Considerations:

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 (in this case, in behalf of the system utility symbiont.)

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 you omit the function code, 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,R,H] DEFAULT=Y
```

- Without spooling:

```
0i? ENTER REQUIRED FUNCTION
```

Initiate the required function with the keyin:

- With spooling:

```
0i△function-code, ⎧Y⎫
                  ⎪N⎪
                  ⎨R⎬
                  ⎩H⎭
```

- 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, N, R, or H is entered), the following message is displayed:

```
0i? IS ⎧PRINTED⎫OUTPUT TO BE SPOOLED FROM SU Y,N,R,H
       ⎩PUNCHED⎭
```

Enter the required spooling option with the keyin:

$$0i\triangle\begin{Bmatrix}Y\\N\\R\\H\end{Bmatrix}$$

To terminate the symbiont, you reply to the ENTER REQUIRED FUNCTION with the end-of-job keyin:

```
0i△EOJ
```

The symbiont terminates and displays the message:

```
0i SYSTEM UTILITY SYMBIONT ENDED
```

You can also end the symbiont by pressing the MESSAGE WAITING key on the system console and keying in one of the following unsolicited messages:

```
00△SU△EOJ
00△TU△EOJ
```

This permits you to terminate the symbiont before it completes a function.

To terminate only the current function of the symbiont, key in one of the following unsolicited messages:

```
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 message (above) to terminate the current function or to terminate the symbiont.

7.3. CARD FUNCTIONS

All the card functions you can perform are described in this subsection. All input card files must be terminated by a card with the words END OF DATA punched in columns 1 through 11.

Proceed as follows 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 device address of the card reader assigned to read the input file.

If no card readers are available, the function aborts 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 aborts and the following message is displayed on the system console:

```
0i NO ⎧PUNCH  ⎫AVAILABLE
      ⎨TAPE   ⎬
      ⎩PRINTER⎭
```

If the required devices are available, the operation continues for each function code, as described in the following subsections.

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 terminated 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 II 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 II can be up to six characters in length. If you supply a name having fewer characters than the number 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 II 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 II.

SU displays the messages:

```
0i DECK NAME n CHARACTERS
0i? XXX
```

Key in the correct characters used as the program (deck) name for output cards. If you provide less than the specified number of characters, the data given is left-justified and space-filled to the right.

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:

- eight characters for a COBOL source deck;
- six characters for an RPG II source deck; and
- three characters for a BAL source deck.

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 tape unit device address, and 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 tape unit device address, mode setting, and block count characteristics of the output tape (as described in 7.4.1). If the output tape reaches EOV before all the card input is processed, SU writes two tapemarks, rewinds, and unloads the tape.

For the message:

```
0i? MOUNT NEXT OUTPUT TAPE. REPLY R OR END
```

Key in:

```
0i△END
```

To terminate the CT function.

Otherwise, dismount the unloaded tape, mount the next output tape; then

Key in:

```
0i△R
```

For the message:

```
0i? DOES USER WISH TO PREP THIS TAPE? REPLY YES OR NO
```

Key in:

```
0i△N
```

If no tape prep is required. The CT function resumes, using the new output tape.

```
0i△Y
```

To prep the tape. The SU INT tape function is activated (7.4.3.9). The CT function resumes when INT function is completed.

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 tape unit device address, and 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:

```
Oi? CUUMMB OUTPUT TAPE B=BLK CNT
```

You respond with the tape unit device address, mode setting, and block count characteristics of the output tape (as described in 7.4.1). If the output tape reaches EOV before all the card input is processed, SU writes two tapemarks, rewinds, and unloads the tape, and provides the option of mounting another output tape to continue. You respond to continue with another tape or terminate the function (as described in 7.3.6).

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 for SU and 1 to 400 for TU.

Procedure:

Same as the CT function, except that tape blocks written are 80 x f long, where f is the blocking factor. 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:

```
Oi? BLK FACTOR
```

You respond with blocking factor 1 to 100 (SU) or 1 to 400 (TU).

For the message:

```
Oi? CUUMM OUTPUT TAPE B=BLK CNT
```

You respond with the tape unit device address, and the mode setting of the output tape (7.4.1).

For the message:

```
Oi? OUTPUT EXCEEDS ALLOCATED BUFFER
```

You must reenter the function and blocking factor because the factor supplied was greater than the maximum allowed and the function was terminated. Multivolume output tape is described in 7.3.6.

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)

The JCP function 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 60 or more 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 or column 61 is required, key in a right parenthesis instead of a blank in that column. The symbiont replaces a right parenthesis in column 1 or 61 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 you can perform are described in the following subsections. We recommend you use the TU symbiont because of the increase in buffer size for selector channel tapes.

7.4.1. Tape Addressing

You identify the tape used for a TU function by its tape unit device address and the mode setting and block count characteristics for the tape, in the following format:

```
cuummb
```

where:

`cuu`
Is the device address.

`mm`
Is the tape mode setting. If mm is blank, 00, or not entered (e.g., response is 102), the SYSGEN mode settings are assumed. (Refer to the current version of the job control user guide, UP-8065.)

`b`
Is the tape block count characteristics. If blank or omitted, the tape is assumed not to 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 needed 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. *The tape block number (TBLK) printed for the TRS function is a display of the 3-byte block number prefixed to block-numbered tapes.*

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 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.

For the messages:

```
0i TAPE ERROR ON INPUT TAPE
0i? B-BYPASS, I-IGNORE, OR E-END FUNCTION
```

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

Proceed as follows to perform a tape function.

1. As described in 7.2:

   a. Enter the TU symbiont command (or enter SU if the tape block size does not exceed 8192 bytes).

   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).

2. Place the input tape volume on an available tape unit, and identify the tape unit to the TU (or SU) symbiont by keying in its unit address as described in 7.4.1.

If the required devices are available, the operation continues for each function code, as described in the following subsections.

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 device address, and the mode setting and block count characteristics of the input tape (7.4.1).

```
0i? CUUMMB-OUTPUT TAPE B=BLK CNT
```

Respond with the tape unit device address, and the 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 the previous description of TT the for the number of files on a standard labeled tape.

If you copy a high density tape to a lower density, or if you copy a longer tape to a shorter tape, all of the input tape data may not fit on the output tape. SU/TU provides the option of mounting another output tape to continue. When the output tape reaches EOV before all the tape input is processed, SU/TU:

- writes two tape marks;
- writes EOV labels, when appropriate, to correspond to a previously copied HDR1 HDR2 label set; and
- rewinds and unloads the tape.

For the message:

```
0i? MOUNT NEXT OUTPUT TAPE. REPLY R OR END
```

Key in:

```
0i△END
```

To terminate the TT function.

Otherwise, dismount the unloaded tape, mount the next output tape; then key in:

```
0i△R
```

If the input tape and the newly mounted output tape are prepped, the TT function resumes. If the input tape is prepped but the new output tape is not, SU/TU queries the operator.

For the messages:

```
0i SUPPLY VOLUME SERIAL NUMBER FOR NEW OUTPUT TAPE
0i? XXXXXX
```

Key in:

```
0i△vsn
```

where:

vsn

Is the 6-character volume serial number for the new output tape. If the vsn is less than six characters, it is left-justified by SU/TU. The TT function resumes, using the new output tape.

For the message:

```
USER LABEL RESTRICTIONS APPLY. SEE UP 8072
```

This message cautions you that SU/TU does not generate user trailer or header labels for multivolume output in addition to those actually encountered on the input tape.

For the message:

```
0i? END OF VOLUME= Y OR N
```

This message is issued after all files specified are copied. Key in Y to write a second tape mark after the file; otherwise N if it is not to be written. After processing this response, the function is completed.

Example:

This example shows tape copy with block count.

For the message:

```
0i? CUUMMB - INPUT TAPE
```

Key in:

```
0i 100△△B
```

For the message:

```
0i? CUUMMB - OUTPUT TAPE
```

Key in:

```
0i 101△△B
```

For the message:

```
0i? #FILES 1-99
```

Key in:

```
0i 3
```

For the message:

```
0i? END OF VOLUME= Y/N
```

Key in:

```
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 end of this function. The block number printed is relative to the location where printing begins.

Procedure:

An option allows you to print the entire field or only a specified number of blocks.

For the messages:

```
0i? CUUMMB - INPUT TAPE B=BLK CNT
```

Respond by entering the tape unit device address and the mode setting and block count characteristics of the input tape (7.4.1). If you enter a block count, 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 to print the entire tape, 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 terminates.

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 allow 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.

For the messages:

```
0i? CUUMMB - INPUT TAPE B=BLK CNT
```

Enter the tape unit device address, and the mode setting and block count characteristic of the input tape (7.4.1).

```
0i? #BLKS OR END
```

Respond END to print the entire tape or respond with 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 is printed, the function terminates.

```
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, except 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, except 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:

You identify the scan field by its length and location within the logical record. The logical record length is also required for deblocking purposes. When you enter the scan argument in either hexadecimal or character format, it is compared to the scan field of each logical tape record and 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 find, you are given the option of continuing the scan for more finds 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 (7.4.2). The function is terminated either at the end-of-file (two tape marks side by side) or when a find 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 invalid hexadecimal character was detected when you entered the scan argument in hexadecimal. You reenter the sequence to obtain the scan argument.

```
0i NO MATCH FND
```

An end-of-file mark (two tape marks back-to-back) was encountered before any finds were made. The function terminates.

```
0i? CONTINUE SCAN? Y/N
```

Message displayed each time a find is made. You enter Y to continue the scan or N to terminate the scan.

The displayed find record is preceded by a header that includes:

REC
Specifies the sequential number of the record, which is relative to the beginning of the search.

DATA
Specifies the logical record length.

TBLK
Specifies the block number, which is maintained by the system in a 3-byte prefix to each tape block.

Example:

This example shows a tape block count search for a record containing a key field of "4637275467" in its first 10 bytes.

For the message:

```
0i? CUUMMB - INPUT TAPE B=BLK CNT
```

Key in:

```
0i 10000B
```

For the message:

```
0i? LOGICAL REC LENGTH
```

Key in:

```
0i 125
```

For the message:

```
0i? LENGTH ARGUMENT (1-30)
```

Key in:

```
0i 10
```

For the message:

```
0i? STARTING DATA POSITION IN REC
```

Key in:

```
0i 1
```

For the message:

```
0i ENTER IN HEX-H, CHAR-C
```

Key in:

```
0i C
```

For the messages:

```
0i ENTER 10 BYTES, 1 CHAR PER BYTE
0i? 1...5...10...
```

Key in:

```
0i 4637275467
```

For the message:

```
0i? CONTINUE SCAN? Y/N
```

Key in:

```
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). If a tape mark is encountered when locating a record, you must decide to continue or terminate the function. If you continue, reenter the direction and number of blocks to the desired record.

If the output tape reaches EOV before the input tape is fully copied, SU/TU provides the option of mounting another output tape to continue (as described in 7.4.3.1). Once the TRL function resumes copying to the new tape, you can't specify a backward direction movement of the tapes (requiring the remounting of the previous output tape).

For the messages:

```
0i? CUUMMB - INPUT TAPE B=BLK CNT
```

Enter the tape unit device address, and the 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 device address, and the mode setting and block count characteristics 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 with F to move tape forward to the desired record, or B to move tape 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. Respond Y to reenter the tape record locating sequence or N to terminate the function.

```
0i? CHANGES TO LENGTH? Y/N
```

Respond Y to change the block size or N if the block size is not to be changed.

```
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 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 to enter new data in hexadecimal or C to enter new data in character format.

```
0i ENTER XX BYTES, Y CHAR PER BYTE
0i? 1...5...10...
```

Respond with new data in first message, where:

XX
Is number of bytes.

Y
Is the format type: 1 indicates character format, and 2 indicates hexadecimal format.

When response is made, the second message appears. This message serves as a guide for entering new data in the format selected.

```
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
```

If changes to the current block are complete, enter Y to initiate the new record locating sequence. If more changes are required, enter N to reenter the change sequence for this record.

Example 1:

Data record 5 is to be changed to hexadecimal 003C in positions 6-7. The tape has standard labels.

For the message:

```
0i? CUUMMB - INPUT TAPE B=BLK CNT
```

Key in:

```
0i 10000B
```

For the message:

```
0i? CUUMMB - OUTPUT TAPE B=BLK CNT
```

Key in:

```
0i 10100B
```

For the message:

```
0i? #BLKS OR END
```

Key in:

```
0i 3 (to copy standard label)
```

For the message:

```
0i? COPY FWD OR BACK? F OR B
```

Key in:

```
0i F
```

For the message:

```
0i? TAPE MARK, CONTINUE? Y OR N
```

Key in:

```
0i Y
```

For the message:

```
0i? #BLKS OR END
```

Key in:

```
0i 5
```

For the message:

```
0i? COPY FWD OR BACK? F OR B
```

Key in:

```
0i F
```

For the message:

```
0i? DESIRED REC? Y OR N
```

Key in:

```
0i Y
```

For the message:

```
0i? CHANGES TO LENGTH? Y OR N
```

Key in:

```
0i N
```

For the message:

```
0i #BYTES TO BE CHANGED, (1-30)
```

Key in:

```
0i 2
```

For the message:

```
0i? STARTING DATA POSITION IN REC
```

Key in:

```
0i 6
```

For the message:

```
0i? ALTER IN HEX-H, CHAR-C
```

Key in:

```
0i H
```

For the messages:

```
0i ENTER 02 BYTES, 2 CHAR PER BYTE
0i? 1...
```

Key in:

```
0i 003C
```

For the message:

```
0i? CHANGES COMPLETE? Y OR N
```

Key in:

```
0i Y
```

For the message:

```
0i? #BLKS OR END
```

Key in:

```
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.

For the message:

```
0i? CUUMMB - INPUT TAPE
```

Key in:

```
0i 10000B
```

For the message:

```
0i? CUUMMB - OUTPUT TAPE
```

Key in:

```
0i 10100B
```

For the message:

```
0i? #BLKS OR END
```

Key in:

```
0i 2
```

For the message:

```
0i? COPY FWD OR BACK? F OR B
```

Key in:

```
0i F
```

For the message:

```
0i? DESIRED REC? Y OR N
```

Key in:

```
0i Y
```

For the message:

```
0i? CHANGES TO LENGTH? Y OR N
```

Key in:

```
0i N
```

For the message:

```
0i? #BYTES TO BE CHANGED, (1-30)
```

Key in:

```
0i 4
```

For the message:

```
0i? STARTING DATA POSITION IN REC
```

Key in:

```
0i 5
```

For the message:

```
0i? ALTER IN HEX-H, CHAR-C
```

Key in:

```
0i C
```

For the messages:

```
0i ENTER 4 BYTES, 1 CHAR PER BYTE
0i? 1...
```

Key in:

```
0i LAB1
```

For the message:

```
0i? CHANGES COMPLETE? Y OR N
```

Key in:

```
0i Y
```

For the message:

```
0i? #BLKS OR END
```

Key in:

```
0i 9999
```

For the message:

```
0i? COPY FWD OR BACK? F OR B
```

Key in:

```
0i F
```

For the message:

```
0i? DESIRED REC? Y OR N
```

Key in:

```
0i N
```

For the message:

```
0i? #BLKS OR END
```

Key in:

```
0i 9999
```

For the message:

```
0i? COPY FWD OR BACK? F OR B
```

Key in:

```
0i F
```

For the message:

```
0i? TAPE MARK, CONTINUE? Y/N
```

Key in:

```
0i Y
```

For the message:

```
0i? #BLKS OR END
```

Key in:

```
0i 1
```

For the message:

```
0i? COPY FWD OR BACK? F OR B
```

Key in:

```
0i F
```

For the message:

```
0i? DESIRED REC? Y OR N
```

Key in:

```
0i Y
```

For the message:

```
0i? CHANGES TO LENGTH? Y OR N
```

Key in:

```
0i N
```

For the message:

```
0i? #BYTES TO BE CHANGED, (1-30)
```

Key in:

```
0i 4
```

For the message:

```
0i? STARTING DATA POSITION IN REC
```

Key in:

```
0i 8
```

For the message:

```
0i? ALTER IN HEX-H, CHAR-C
```

Key in:

```
0i C
```

For the messages:

```
0i ENTER 4 BYTES, 1 CHAR PER BYTE
0i? 1...
```

Key in:

```
0i LAB1
```

For the message:

```
0i? CHANGES COMPLETE? Y OR N
```

Key in:

```
0i Y
```

For the message:

```
0i? #BLKS OR END
```

Key in:

```
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, 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:

A card with all * is punched when a tape mark is encountered, The function terminates 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 tape unit device address, and the 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. When you enter the volume serial number and file label, the tape is positioned to the load point prior to the prep and the VOL1, HDR1, and HDR2 labels are written, along with two tape marks. When the prep is completed, the tape is positioned following the first tape mark. No AVR is performed; the volume serial number is not entered in the pub. It is, therefore, possible to prep multiple tapes with the same vsn.

Procedure:

For the messages:

```
0i? CUUMMB - OUTPUT TAPE B=BLK CNT
```

Respond with the tape unit device address and the 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.

NOTE:

System utility permits embedded blanks in file names. However, you must not place apostrophes or quotes around file names as delimiters, or they will be taken by system utility as part of the file names.

Example:

For the message:

```
0i? CUUMMB - OUTPUT TAPE
```

Key in:

```
0i 10100B
```

For the messages:

```
0i ENTER NEW VOL#
0i? XXXXXX
```

Key in:

```
0i 001036
```

For the messages:

```
0i ENTER NEW FILE LABEL
0i? XXXXXXXXXXXXXXXXX
```

Key in:

```
0i TAPE1
```

7.4.3.10. Forward Space to a Specific File (FSF)

You use the FSF function code to advance the tape 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 device address and the 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 backspace the tape until a tape mark is encountered.

Procedure:

For the message:

```
0i? CUUMM - OUTPUT TAPE
```

Enter the tape unit device address and the 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.

Procedure:

For the messages:

```
0i? CUUMM - OUTPUT TAPE
```

Enter the tape unit device address and the 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.

Procedure:

For the messages:

```
0i? CUUMM - OUTPUT TAPE
```

Enter the tape unit device address and the 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
```

Enter the tape unit device address and the 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
```

Enter the tape unit device address and the 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
```

Enter the tape unit device address and the 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
```

Enter the tape unit device address and the 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 subsections.

7.5.1. Disk Operating Instructions

When using the system utility for disk operations, the end-of-file record is a disk record on which data length is binary zero (not applicable on IDA disk subsystems).

Proceed as follows to perform a disk function.

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).

7.5.1.1. 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 disk unit device address 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 you enter less than 44 characters, the file ID is padded with blanks on the right. Printing begins at the start of the file when a file is entered.

NOTE:

System utility permits embedded blanks in file names. However, you must not place apostrophes or quotes around file names as delimiters, or they will be taken by system utility as part of the file names.

```
0i? CCCHH - END
```

Enter the last cylinder (CCC) and head (HH) in decimal to be displayed.

```
0i? NO OF TRACKS TO PRINT UP TO 9
```

If you 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 extent. If the file is not laced, printing continues for the requested number of tracks, or until end of file (EOF) is encountered, whichever comes first. If the file is laced, printing continues for the requested number of tracks. Printing is not confined to the extents specified in the format label. (Laced files are discussed in the system service programs user guide, UP-8062 (current version) in the section describing diskette prep.)

Example 1:

Cylinder 3 head 4 through cylinder 6 head 6 on device 300 are to be printed.

For the message:

```
0i? ENTER DVC ADDRESS
```

Key in:

```
0i 300
```

For the message:

```
0i? CCCHH - BEGIN OR FILE-ID
```

Key in:

```
0i 00304
```

For the message:

```
0i? CCCHH END
```

Key in:

```
0i 00606
```

Example 2:

The first three tracks from a file called SEQUENTIAL DISC on device 440 are to be printed.

For the message:

```
0i? ENTER DVC ADDRESS
```

Key in:

```
0i 440
```

For the message:

```
0i? CCCHH - BEGIN OR FILE-ID
```

Key in:

```
0i SEQUENTIAL DISC
```

For the message:

```
0i? NO OF TRACKS TO PRINT UP TO 9
```

Key in:

```
0i 3
```

7.5.1.2. 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.

Procedure:

For the messages:

```
0i? ENTER DVC ADDRESS
```

Enter the disk unit device address 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.

7.5.1.3. Printing the Disk Volume Table of Contents (VTP)

You use the VTP function code to get a copy of your VTOC. You can print:

- 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) for the requested volume. VTP will not process other than OS/3-created VTOCs. Use the DD function to print a VTOC from a non-OS/3-created disk.

Procedure:

Three list options are available, as follows:

- VSN, DI – Device information only

  Lists the available free extents and other information on the requested volume.

- VSN, FILE-ID – File information

  Lists device information plus the extent and other information on the requested file.

- VSN, ALL – Full VTOC listing

  Lists device information plus the extent and other information for all files allocated on the requested file.

For the message:

```
Oi? ENTER DVC/VSN, DI, ALL, FILE-ID, END OR EOJ
```

Key in one of the following, where vsn is the volume serial number of the disk pack to be printed.

For full VTOC listing:

```
Oi△vsn,ALL
```

For a listing of only the device information:

```
Oi△vsn,DI
```

For a listing of up to 44 characters as used on the // LBL card when the file was created:

```
Oi△vsn,FILE-ID
```

To terminate the VTP function:

```
Oi△END
```

To terminate SU:

```
0i△EOJ
```

NOTES:

1. *The device address of the disk pack may replace the vsn in the receding messages.*

2. *When a VTP request terminates, SU reproduces the message:*

   ```
   0i?ENTER DVC/VSN, DI, ALL, FILE-ID, END OR EOJ
   ```

   You may specify another volume for VTP display, END to end VTP, or EOJ to terminate SU.

3. *System utility permits embedded blanks in file names. However, you must not place apostrophes or quotes around file names as delimiters, or they will be taken by system utility as part of the file names.*

7.5.1.3.1. Disk VTP Listing Summary

The various information listed by VTP is either taken directly from the disk labels or calculated from data contained in the labels. A summary of the VTOC information listed by the VTP function is provided in Table 7-5. (Refer to Appendix D in the current version of the data management user guide, UP-8068, for details.)

Table 7—5. Summary of Disk VTP Information (Part 1 of 4)

Field Heading	Field Label
Volume Information	
VOLUME SERIAL NUMBER	DL$VSN
VTOC ADDRESS (CCC HHRR)	DL$VTC
VOLUME SECURITY	DL$VSB
OWNER NAME/ADDR CODE	DL$ONR
Device Information	
ADDRESS LAST FORMAT 1	DL$LF4
ADDRESS HIGHEST ALT TRACK	DL$HA4
NO CYL/TRK THIS DISK	DL$DS4
TOLERANCE	DL$TO4
NO UNUSED VTOC RECORDS	DL$AF4
NO OF ALT TRACKS	DL$AT4
NO BYTES PER TRACK	DL$TL4
NO VTOC LABELS PER TRACK	DL$LT4

Table 7—5. Summary of Disk VTP Information (Part 2 of 4)

Field Heading	Field Label
Available Extent Data	
START ADDRESS	DL$XT5
END ADDRESS	DL$XC5 (calculated)
ENTENT SIZE	(calculated)
File Information	
FILE NAME	DL$ID1
FILE SERIAL NUMBER	DL$FS1
VOLUME SEQ NO	DL$VS1
CREATION DATE	DL$CD1
EXPIRATION DATE	DL$ED1
EXTENT COUNT	DL$XC1
OPTION CODES	DL$OC1
PCA COUNT	DL$PC1
TRACKS/CYLINDER	DL$TPC2
LOW HEAD	DL$FLH2
FILE TYPE	DL$FT1
File Partition (PCA) Information	
PCA NUMBER	(1 through 7)
BLOCK SIZE	DL$BL1
RECORD SIZE*	DL$RL1
RECORD FORMAT*	DL$RF1
BLOCK FACTOR*	(calculated: records per block for fixed-length blocked record format.)
BLOCKS PER TRACK	DL$SLA2
NO. OF RECS IN LAST BLOCK	DL$SPC2
KEY LENGTH OR LACE FACTOR	DL$SLF2
END OF DATA BLOCK NO.	DL$SEP2
TOTAL # RECORDS	(calculated: totals number of records for sequential FIXBLK and FIXUNB files, and for VARUNB files.)

*Not applicable to IRAM/MIRAM files on PCA basis.

Table 7—5. Summary of Disk VTP Information (Part 3 of 4)

Field Heading	Field Label
Logical Extent Information	
PCA NO	(0 through 7)
START CCC' HH	DL$SXAR2 (calculated from relative track address)
END ADDR CCC HH	DL$SXAR2 (calculated from number of tracks in extent)
Allocation Summary	
EXT SIZE CCC HH	DL$SXAR2 (calculated: END ADDR minus START.)
CUM/PCA CCC HH	(calculated: cumulative CCC HH for this PCA.)
CUM/FIL CCC HH	(calculated: cumulative CCC HH for this file.)
Extent Utilization Summary	
OCCUPIED/PCA CCC HH BLOCK	(calculated: allocated space used for this PCA.)
AVAILABLE/PCA CCC HH BLOCK	(calculated: unused space for this PCA.)
RECORD CAPAC REMAINING	(calculated: approximate number of logical records that fit in available space; not calculated for variable-length blocked record format.)
END OF DATA CCC HH RRR	DL$SEP2 (calculated)
Physical Extent Information	
EXTENT TYPE	DL$XT1
EXTENT SEQ NO.	DL$SX1
EXTENT START CCC HH	DL$XL1
EXTENT END CCC HH	DL$XU1
CUM CCC HH	(calculated)
AUTO EXTENT INCREMENT	DL$SA1
ISAM File Information	
KEY LENGTH	DL$SLF2
KEY LOCATION	DL$KL1
LAST PRIME DATA REC ID	DL$PID2
FULL OVFL CYL COUNT	DL$NMA2
INDEP OVFL CYL ADDRESS	DL$1OF2
PRIME DATA LOAD COUNT	DL$PDLC2
OVERFLOW REC COUNT	DL$NMO2
LAST INDEX REC ADDRESS	DL$BID2
PRIME DATA RECORD COUNT	DL$NMP2
BYTES REQD FOR MAIN STOR	DL$NMS2

Table 7—5. Summary of Disk VTP Information (Part 4 of 4)

Field Heading	Field Label
ISAM File Information (cont)	
DELETED REC COUNT	DL$NMT2
BLOCKS PER CYL COUNT	DL$NMT2+2(2)
PERCENT OVERFLOW	(calculated)
SAT File Information	
DIRECTORY PCA LACE FACTOR	DL$DIRL2
DIRECTORY PCA LACE ADJUST	DL$DIRF2
TEXT PCA LACE FACTOR	DL$TXTL2
TEXT PCA LACE ADJ	DL$TXTF2
(M)IRAM File Information	
KEY LOCATION	DL$XILOC
KEY LENGTH	DL$XILOC+2
DUPLICATES ALLOWED	DL$XILOC+3, X'80'
CHANGES ALLOWED	DL$XILOC+3, X'40'
DATA RECORD COUNT	DL$COUTR
RECORD SIZE	DL$DREC
INDEX BUFF SIZE	DL$CSIZ

7.5.1.4. Printing the Short Format Volume Table of Contents (SVT)

You use the SVT function to obtain an abbreviated VTOC listing, consisting of a single print line for each physical extent for each file.

Procedure:

For the message:

```
0i? ENTER DVC OR VSN OR END
```

Enter the disk unit device address or the volume serial number (up to six characters) of the VTOC to be printed; either entry prints the abbreviated VTOC. Enter END to terminate the SVT function.

NOTE:

If SVT is attempted on a data set label diskette, an error message is displayed.

7.5.1.4.1. Disk SVT Listing Summary

The various information listed by SVT is either taken directly from the disk labels or calculated from data contained in the labels. A summary of the short format VTOC information listed by the SVT function is provided in Table 7-6. (Refer to Appendix D in the current version of the data management user guide, UP-8068, for details.)

Table 7—6. Summary of Disk SVT Information

Field Heading	Field Label
VSN	DL$VSN
SECUR	DL$VSB
OWNER	DL$ONR
FILE-NAME	DL$ID1
FILE TYPE	DL$FT1
FILE SER. NO	DL$FS1
SEQ NO	DL$VS1
CREATION DATE	DL$CD1
EXPIRATN DATE	DL$ED1
OPT COD	DL$OC1
PCA CT	DL$PC1
START CCC HH	DL$XL1
END CCC HH	DL$XU1
CUM CCC HH	(calculated: cumulative)
AI	DL$SA1

7.5.1.5. Displaying the Available Disk Extents (AVX)

You use the AVX function to display a list of available disk extents on the system console screen. The display is similar to the listing printed by the VTP function with the DI option; however, output is displayed at the system console and printed at the console output printer rather than at the customary line printer.

Procedure:

For the message:

```
0i? ENTER DEVICE ADDRESS
```

Enter the disk unit device address disk whose available extents are to be displayed.

NOTE:

→ *If AVX is attempted on a data set label diskette, an error message is displayed.*

Table 7—7. Summary of Disk AVX Information

Field Heading	Field Label
VSN	DL$VSN
SECUR	DL$VSB
VTOC ADRS	DL$VTC
OWNR	DL$ONR
START CCC HH	DL$XT5
END CCC HH	DL$XC5 (calculated)
SIZE CCC HH	(calculated)

7.5.1.5.1. AVX Listing Summary

The various information listed by AVX is either taken directly from the disk labels or calculated from data contained in the labels. A summary of the available disk extent information listed by the AVX function is provided in Table 7-7. (Refer to Appendix D in the current version of the data management user guide, UP-8068, for details.)

7.5.1.6. 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 disk unit device address of 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

Proceed as follows to request a diskette function.

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 diskette unit device address of 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 17 characters) as used on the // LBL job control statement when the file was created. If you enter less than 17 characters, the file ID is padded with blanks on the right. Printing begins at the start of the file when a file ID is entered.

NOTE:

System utility permits embedded blanks in file names. However, you must not place apostrophes or quotes around file names as delimiters, or they will be taken by system utility as part of the file names.

```
0i? TTRR-END
```

Enter the last track (TT) and sector (RR) in decimal to be displayed.

7.6.1.2. Printing the Diskette Volume Table of Contents (VTP)

You use the VTP function code to get a copy of your VTOC. You can print:

- 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 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 volume table of contents (VTOC) for the requested volume.

Procedure:

Three list options are available, as follows:

- VSN, DI – Device information only

  Lists the available free extents and other information on the requested volume.

- VSN, FILE-ID – File information

  Lists device information plus the extent and other information on the requested file.

- VSN, ALL – Full VTOC listing

  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, END OR EOJ
```

Key in one of the following, where vsn is the volume serial number of the diskette to be printed:

For a full VTOC listing:

```
0i△vsn,ALL
```

For a listing of only the device information:

```
0i△vsn,DI
```

For a listing of up to 17 characters as used on the // LBL card when the file was created:

```
0i△vsn,FILE-ID
```

To terminate the VTP function:

```
0i△END
```

To terminate SU:

```
0i△EOJ
```

NOTES:

1. *The device address of the diskette may replace the vsn in the preceding messages.*

2. *When a VTP request is completed, SU reproduces the message:*

```
0i? ENTER DVC/VSN, DI, ALL, FILE-ID, END OR EOJ
```

You may specify another volume for VTP display, END to end VTP, or EOJ to terminate SU.

3. *System utility permits embedded blanks in file names. However, you must not place apostrophes or quotes around file names as delimiters, or they will be taken by system utility as part of the file names.*

7.6.1.2.1. Diskette VTP Listing Summary

The information listed by VTP is either taken directly from the diskette volume and data set labels or calculated from data contained in the labels. A summary of the VTOC information listed by the VTP function is provided in Table 7–8. (Refer to Appendix D in the current version of the data management user guide, UP-8068, for details.)

Table 7—8. Summary of Diskette VTP Information

Field Heading	Byte Position
VOLUME INFORMATION	VOL1 LABEL
VOLUME SERIAL NUMBER	4
OWNER ID	37
NO. RECORDING SURFACES	71
EXTENT ARRANGEMENT CONSTRAINTS	73 C'P' = YES, else NO.
PHYSICAL SECTOR LENGTH	75 C'1' = 256, else 128.
DEFECTIVE TRACKS	6, 10 of ERMAP record.
FILE INFORMATION	DATA SET LABEL
FILE IDENT	5
BEGIN (OF EXTENT)	28, 30
END (OF EXTENT)	34, 36
END OF DATA	74, 76
BLOCK LENGTH	22
RECORD LENGTH	53
BYPASS INDICATOR	40
FILE SECURITY	41
WRITE PROTECT	42
EXCHANGE TYPE IND	43
VOL SEQ(UENCE)	45
(FILE) CREATION DATE	47-52
(FILE) EXPIR DATE	66-71

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 changes the VSN only 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 diskette unit device address of 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:

- Commands, parameters, and input messages in uppercase letters must be keyed in literally. For example:

  ```
  SYSDUMP
  ```

  Output messages are shown in uppercase letters. For example:

  ```
  ▷ 01 EARLY WARNING OVERTEMP. CONDITION EXISTS
  ```

- The delta symbol (△) indicates a space:

  ```
  SHUTDOWN△DDP
  ```

- 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
  ```

- Underscoring a portion of a command and its modifiers and parameters indicates the shortest abbreviations that must 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:

  $$\underline{\texttt{DE}}\texttt{LETE}\triangle\texttt{jobname}$$

- Braces { } illustrate alternate choices. For example, the format of the change command

  $$\underline{\texttt{CH}}\texttt{ANGE}\triangle\texttt{jobname,}\begin{Bmatrix}\underline{\texttt{P}}\texttt{RE}\\ \underline{\texttt{H}}\texttt{IGH}\\ \underline{\texttt{N}}\texttt{OR}\end{Bmatrix}$$

  indicates that PRE (P), HIGH (H), or NOR (N) may be keyed in after the jobname.

- Brackets [ ] denote optional entries. For example, this portion of the FILE command format

```
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, this portion of the DISPLAY command format

```
DISPLAY△JBQ ⎡,⎧PRE ⎫⎤
            ⎢ ⎪HIGH⎪⎥
            ⎢ ⎨NOR ⎬⎥
            ⎣ ⎩ALL ⎭⎦
```

specifies that the JBQ parameter must be included. If parameter 2 is not specified, jobs in all scheduling priority queues are displayed.

NOTE:

Not all optional parameters have a default specification.

- An ellipsis (three periods) indicates 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 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 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 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 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.

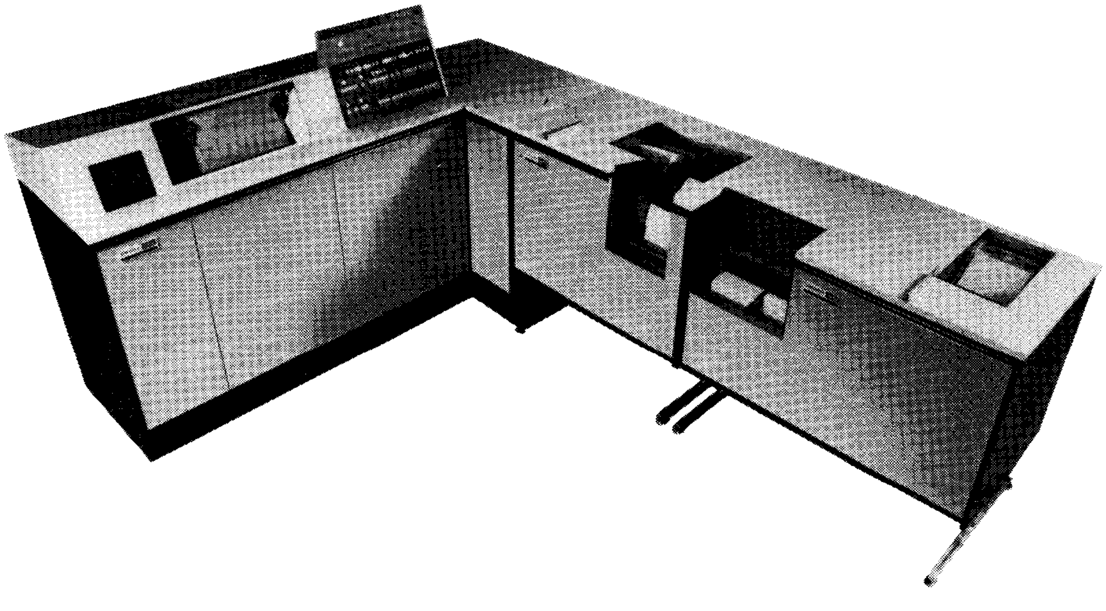


Figure B—1. 9200/9300 Series Subsystem

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 pressed 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 or 90/40 system console.

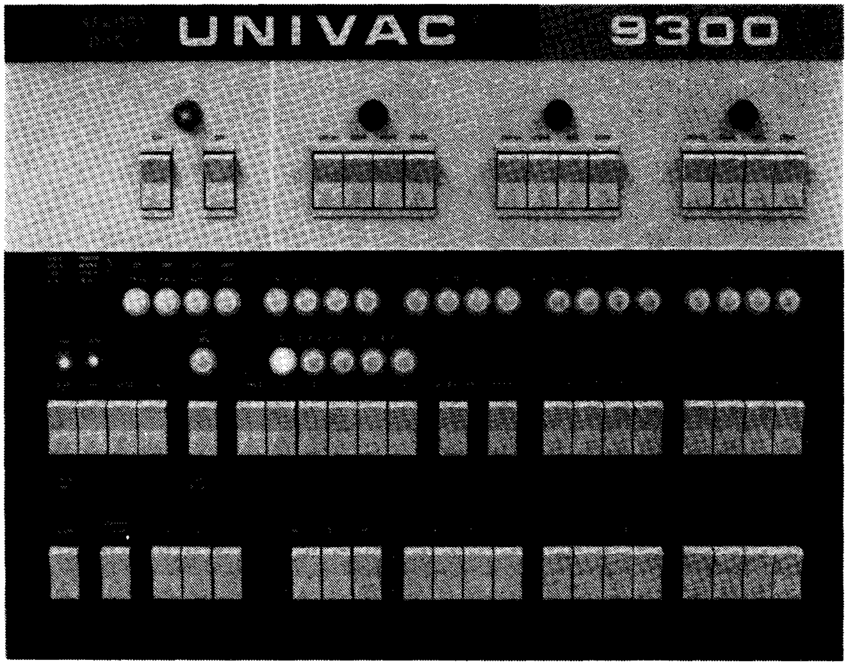


Figure B—2. Control Console, Controls and Indicators

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 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 acknowledge message containing the current status of all the 9200/9300 I/O devices is transmitted to OS/3. 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. Press 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 press 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 or 90/40 console or through halt displays on the 9200/9300 control panel.

B.5.1. 90/30 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 and 90/40 Console Messages for the 9200/9300 Series Subsystem

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.
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.

Appendix C. Supervisor Modification Procedure

For special processing requirements, your system administrator may tell you to modify the selections made for the supervisor during SYSGEN. Respond with Y to the MODIFY SUPERVISOR? IPL statement; then proceed with the appropriate operator action described in Tables C-1 and C-2.

NOTE:

For a description of system output messages and how to respond to them (via solicited input messages), see 4.2.2.1 and 4.2.2.2.

Table C—1. How to Modify List of Resident Shared Code Modules (Part 1 of 2)

Output Message	Operator Response
0i? IS THE SHARED CODE DIRECTORY INDEX TO BE BUILT?(Y,N)	0i Y Specifies an index is to be built for the shared code library. The system will make a binary search on the index rather than a serial directory search. This allows faster loading of shared code modules into main storage and faster processing between two or more shared code modules. The directory itself uses approximately 4000 bytes of main storage. 0i N No shared code directory index is to be built.
0i? ANY RESIDENT SHARED MODULES TO ADD OR DELETE? (Y,N)	0i N Terminates output messages for modifying shared code module list. Specifies no modification is required to SYSGEN list of shared code modules to be made resident. Proceed with operator action described in Table C-2. 0i Y Specifies SYSGEN list of shared code modules requires modification. Output message requesting a function is displayed.

Table C—1. How to Modify List of Resident Shared Code Modules (Part 2 of 2)

Output Message	Operator Response
0i? FUNCTION? (ADD,DEL,LIST,END,HELP)	0i END Terminates output messages for modifying shared code module list. Specifies no further supervisor modification is required. The SYSGEN list of shared code modules is updated by the changes previously specified with the ADD and DEL functions. Modules in updated list are loaded and made resident until next IPL. Proceed with operator action described in Table C-2. 0i HELP Displays brief description of the ADD, DEL, LIST, and END functions, followed by an output message requesting another function. 0i LIST Displays a list of the individual shared code modules currently specified to be made resident, followed by an output message requesting another function. 0i $\left\{\begin{matrix}\text{ADD}\\ \text{DEL}\end{matrix}\right\}$ Specifies an individual module or group of modules is to be added to or deleted (DEL) from the current list of shared code modules. Displays the following list of shared code module groups, followed by an output message requesting a group or module name. POSSIBLE SYSTEM GROUPS ARE: SF - SCREEN FORMAT SERVICES ISB - BASIC INTERACTIVE SERVICES ISF - FULL INTERACTIVE SERVICES EDT - EDITOR DP - DIALOGUE PROCESSOR RPG - RPG EDITOR DDPR - REMOTE DISTRIBUTED DATA PROCESSING DDPL - LOCAL DISTRIBUTED DATA PROCESSING ESC - ESCORT ESCF - ESCORT FULL BAS - BASIC
0i? WHICH GROUP TO $\left\{\begin{matrix}\text{ADD?}\\ \text{DELETE?}\end{matrix}\right\}$	0i $\left\{\begin{matrix}\text{group-name}\\ \text{module-name}\end{matrix}\right\}$[,L] Specifies the group of shared code modules or the individual shared code module to be added or deleted. Provides option to display (L) individual module names as they are added or deleted from the current list. After processing the addition or deletion, an output message requests another function. NOTE: Run the SCLIST job to get a list of all system shared-code modules filed in YSCLOD.

Table C—2. How to Modify Buffer Sizes Used by Dynamic Buffer Management

IPL Output Message	Operator Response
0i? ANY CHANGES TO DYNAMIC BUFFER MANAGEMENT PARAMETERS? (Y,N)	0i N Terminates output messages for modifying buffer sizes. Specifies no changes are required to the expansion region and resident buffer sizes specified at SYSGEN. Proceed with step 2m of the interactive IPL procedure (3.3). 0i Y Specifies SYSGEN-specified expansion region and resident buffer sizes require modification. Output message requesting new expansion region size is displayed.
0i? ENTER NEW EXPANSION REGION SIZE? (CURRENTLY decimal-byte-size)	0i new-size Specifies new size*, in decimal bytes, of expansion region and terminates output messages for modifying buffer sizes. Expansion region size is changed until next IPL. Proceed with step 2m of the interactive IPL procedure (3.3). 0i 0 Specifies no expansion regions are to be allocated until the next IPL. Dynamic buffer management must use resident buffer pool. Output message requesting new resident buffer size is displayed.
0i? ENTER NEW RESIDENT BUFFER SIZE? (CURRENTLY decimal-byte-size)	0i new-size Specifies new size*, in decimal bytes, of resident buffer pool and terminates output messages for modifying buffer sizes. Resident buffer size is changed until next IPL. Proceed with step 2m of the interactive IPL procedure (3.3).

*The new decimal size can be specified with or without a comma or as a multiple of K. For example:

200000 is 200,000 decimal bytes.

200,000 is 200,000 decimal bytes.

200K is 200 x 1024 = 204,800 decimal bytes.

Index

SPERRY UNIVAC

USER COMMENT SHEET

Your comments concerning this document will be welcomed by Sperry Univac for use in improving subsequent editions.

Please note: This form is not intended to be used as an order blank.

____
(Document Title)

______________ ______________ ______________
(Document No.) *(Revision No.)* *(Update No.)*

Comments:

From:

____
(Name of User)

____
(Business Address)

Fold on dotted lines, and mail. (No postage stamp is necessary if mailed in the U.S.A.)
Thank you for your cooperation

Cut along line.

FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 21 BLUE BELL, PA.

POSTAGE WILL BE PAID BY ADDRESSEE

SPERRY UNIVAC

ATTN.: SYSTEMS PUBLICATIONS

P.O. BOX 500
BLUE BELL, PENNSYLVANIA 19424

CUT

FOLD

SPERRY UNIVAC

Cut along line.

USER COMMENT SHEET

Your comments concerning this document will be welcomed by Sperry Univac for use in improving subsequent editions.

Please note: This form is not intended to be used as an order blank.

(Document Title)

(Document No.) *(Revision No.)* *(Update No.)*

Comments:

From:

(Name of User)

(Business Address)

Fold on dotted lines, and mail. (No postage stamp is necessary if mailed in the U.S.A.)
Thank you for your cooperation

FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 21 BLUE BELL, PA.

POSTAGE WILL BE PAID BY ADDRESSEE

SPERRY UNIVAC

ATTN.: SYSTEMS PUBLICATIONS

P.O. BOX 500
BLUE BELL, PENNSYLVANIA 19424

CUT

FOLD

SPERRY UNIVAC

USER COMMENT SHEET

Your comments concerning this document will be welcomed by Sperry Univac for use in improving subsequent editions.

Please note: This form is not intended to be used as an order blank.

____

(Document Title)

____________________ ____________________ ____________________

(Document No.) *(Revision No.)* *(Update No.)*

Comments:

From:

____

(Name of User)

____

(Business Address)

Fold on dotted lines, and mail. (No postage stamp is necessary if mailed in the U.S.A.)
Thank you for your cooperation

Cut along line.

FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 21 BLUE BELL, PA.

POSTAGE WILL BE PAID BY ADDRESSEE

SPERRY UNIVAC

ATTN.: SYSTEMS PUBLICATIONS

P.O. BOX 500
BLUE BELL, PENNSYLVANIA 19424

FOLD

CUT