UNISYS

System 80 Models 8-20 OS/3

Operations Guide

November 1991

Printed in U S America
7004 5208-000

Priced Item

UNISYS

System 80 Models 8-20 OS/3

Operations Guide

OS/3 Release 14

November 1991

Printed in U S America

Priced Item

7004 5208-000

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

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

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

Correspondence regarding this publication should be forwarded to Unisys Corporation either by using the User Reply form at the back of this manual or by addressing remarks directly to Unisys Corporation, SPG East Coast Systems Documentation Development, Tredyffrin Plant, 2476 Swedesford Road, P.O. Box 203, Paoli, PA, 19301-0203, U.S.A.

UNISYS

Product Information Announcement

o New Release • Revision o Update o New Mail Code

Title

System 80 Models 8-20 OS/3 Operations Guide

This announces the release of a revision to this guide. It was previously released as the *System 80 Models 3-6 and 8-20 OS/3 Operations Guide*, UP-8859.

This guide provides all information required to operate System 80 models 8, 10, 15, and 20 running under the OS/3 operating system. The guide is intended for use by system operators and administrators.

Changes for OS/3 Release 14 include:

- An enhanced console screen that provides a menu bar, pull-down windows, and express command entries via the keyboard function key.
- An automatic boot procedure that automatically loads your supervisor each time an auto-IPL occurs.
- Addition of the NOWRITE and WRITE parameters for the SET IO command (used to write-protect your I/O devices).
- Additional job status informational messages (JS/SY command).
- Security and resource management generation requirements affecting the LIMITS command.
- A SET LX command that lets delegated console/workstation users view messages sent to logged off users.
- Procedures for downline loading a Telcon file to front-end processor or a remote concentrator DCPs.
- Two ICAM unsolicited messages that bring a switched line type up as either a manual dial line or an unattended line.
- Revised information for optimizing the disk cache facility.
- Expanded ONUERL procedures for running the program manually on an as-needed basis or automatically on a fixed, 7-day cycle.
- The deletion of all information related to System 80 Models 3 through 6.

To order additional copies of this document:

- United States customers should call Unisys Direct at 1-800-448-1424.
- All other customers should contact their Unisys Subsidiary Librarian.
- Unisys personnel should use the Electronic Literature Ordering (ELO) system.

Announcement only:	Announcement and attachments:	System: System 80
SAB, SAE, and MB00	ECC3 and MB01	Release: 14
		Date: November 1991
		Part Number: 7004 5208-000

PAGE STATUS SUMMARY
ISSUE: 7004 5208-000

Part/Section	Page Number	Update Level
Cover		000
Title Page/Disclaimer		000
PSS	iii	000
About This Document	v thru x	000
Contents	xi thru xix	000
1	1 thru 23	000
2	1 thru 18	000
3	1	000
4	1 thru 87	000
5	1 thru 11	000
6	1 thru 14	000
7	1 thru 38	000
8	1 thru 13	000
9	1 thru 40	000
10	1 thru 4	000
Appendix A	1 thru 4	000
Index	1 thru 10	000
User Reply Form		000
Back Cover		000

Part/Section	Page Number	Update Level

Part/Section	Page Number	Update Level

Unisys uses an 11-digit document numbering system. The suffix of the document number (1234 5678-xyz) indicates the document level. The first digit of the suffix (x) designates a revision level; the second digit (y) designates an update level. For example, the first release of a document has a suffix of -000. A suffix of -130 designates the third update to revision 1. The third digit (z) is used to indicate an errata for a particular level and is not reflected in the page status summary.

About This Document

Purpose

This guide is one of a series designed to instruct and guide the operator in the procedures required to operate the System 80 under the control of Operating System/3 (OS/3). It indicates how to initiate and control basic system operations. This guide also provides general reference material that may be used after you have become proficient with all basic operating procedures.

Scope

This guide describes the daily operational tasks normally performed by the system operator. These tasks include system start-up and initialization, job processing, control of interactive services, control of the integrated communications access method (ICAM), and use of I/O media system services.

Audience

The intended audience for this guide is the novice operator with a basic knowledge of data processing operations and the more sophisticated operator who has some experience with the Unisys operating system/3 (OS/3).

Prerequisites

The reader should have experience or training in the concept of data processing.

How to Use This Guide

Read the entire guide to familiarize yourself with the concepts, procedures, and instructions that it presents; then use it for reference as needed.

- The delta symbol Δ indicates a space:

```
SHUTDOWNΔDDP
```

- 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.

```
DELETEΔjobname
```

- Underlined letters in a command indicate that they are the only letters required to be keyed in to initiate processing of the command and its associated symbiont. For example, only the letters DE need be keyed in to initiate processing of the DELETE command and subsequent running of the delete symbiont. Its format is presented as:

```
DELETEΔjobname
```

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

```
CHANGEΔjobname, ⎧PRE ⎫
                ⎨HIGH⎬
                ⎪NOR ⎪
                ⎩LOW ⎭
```

indicates that PRE (P), HIGH (H), NOR (N) or Low (L) may be keyed in after the job name.

- 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, the ENTER command format

```
ENTERΔ'filename',QUEUE=RDR ⎡,HOLD= ⎧N⎫⎤
                           ⎣       ⎩Y⎭⎦
```

specifies that the filename and QUEUE=RDR parameters must be included (parameters 1 and 2). If you omit parameter 3, HOLD=Y is used.

Note: *Not all optional parameters have a default specification.*

- An ellipsis (...) indicates the omission of a variable number of entries.

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

- A colon (:) and parenthesis symbols () are indicates for several job initialization commands. These symbols are part of the applicable command syntax and must be entered to define the indicated parameter.

Related Product Information

Other current OS/3 publications that are helpful when operating the models 8-20 system include:

Note: *Throughout this guide, when we refer you to another manual, use the current version that applies to the software level in use at your site.*

Consolidated Data Management Programming Guide, **UP-9978**

Describes what data management is and how it handles the transfer of data between peripheral devices and programs. It also describes OS/3 data file formats.

Dump Analysis Programming Guide, **UP-9980**

Discusses the process of analyzing OS/3 dumps. Explains the purpose of various types of dumps, when and how to use them, their formats, and their interpretation.

Installation Verification Procedures (IVP) Operating Guide, **7004 5232**

Describes the procedures used to verify that OS/3 software products are properly installed and ready to use.

Interactive Services Operating Guide, **UP-9972**

Describes the use of interactive services terminals and their capabilities. Includes log-on and log-off procedures, ICAM sign-on procedures, and use of interactive commands and utilities.

Integrated Communications Access Method (ICAM) Utilities Programming Guide, **7004 4565**

Describes how to use the ICAM utility routines.

***Interactive Services Commands and Facilities Programming and Operations Quick-Reference Guide,* UP-9973**

Provides quick-reference information on interactive services commands, log-on and log-off procedures, and ICAM sign-on procedures.

***Job Control Programming Guide,* 7004 4623**

Describes the OS/3 job control procedures and options.

***Models 8-20 Installation Guide,* 7004 5505**

Describes procedures for installation and maintenance of OS/3 software in a model 8-20 environment.

***Models 8-20 Integrated Communications Access Method (ICAM) Operations Guide,* 7004 4557**

Describes how to prepare the ICAM symboint to support communications on model 8-20 systems.

***Spooling and Job Accounting Operating Guide,* 7004 4581**

Describes the spooling capabilities and controls provided for the system administrator, programmers, and operators.

***Supervisor Technical Overview,* UP-8831**

Describes concepts of the OS/3 supervisor.

***System Messages Reference Manual,* 7004 5190**

Describes the OS/3 system messages and operator actions.

Contents

Section 2. System Power-On and Initialization Procedures

Section 3. System Power-Off Procedure

Section 4. Job Processing Procedures

Section 5. Interactive Services

Section 6. Integrated Communications Access Method (ICAM) Procedures

Section 7. System Utility Services

Section 9. ONUERL Program

Section 10. Error Data Handling

Appendix A. Supervisor Modification Procedure

Index

User Reply Form

Figures

Tables

Section 1
System Definition

System 80 models 8-20 are interactive data processing systems which operate on Operating System/3 (OS/3), the multiprogramming software system. The versatility of System 80 models 8-20 permit you an almost unlimited variety of system configurations from which you can tailor one to your particular needs. The system is configured from the basic processor complex to maximum expansion with one expansion cabinet.

1.1. System Configuration

Diagrams showing minimum and maximum configurations of the System 80 models 8-20 are presented in Figures 1-1 and 1-2, respectively.

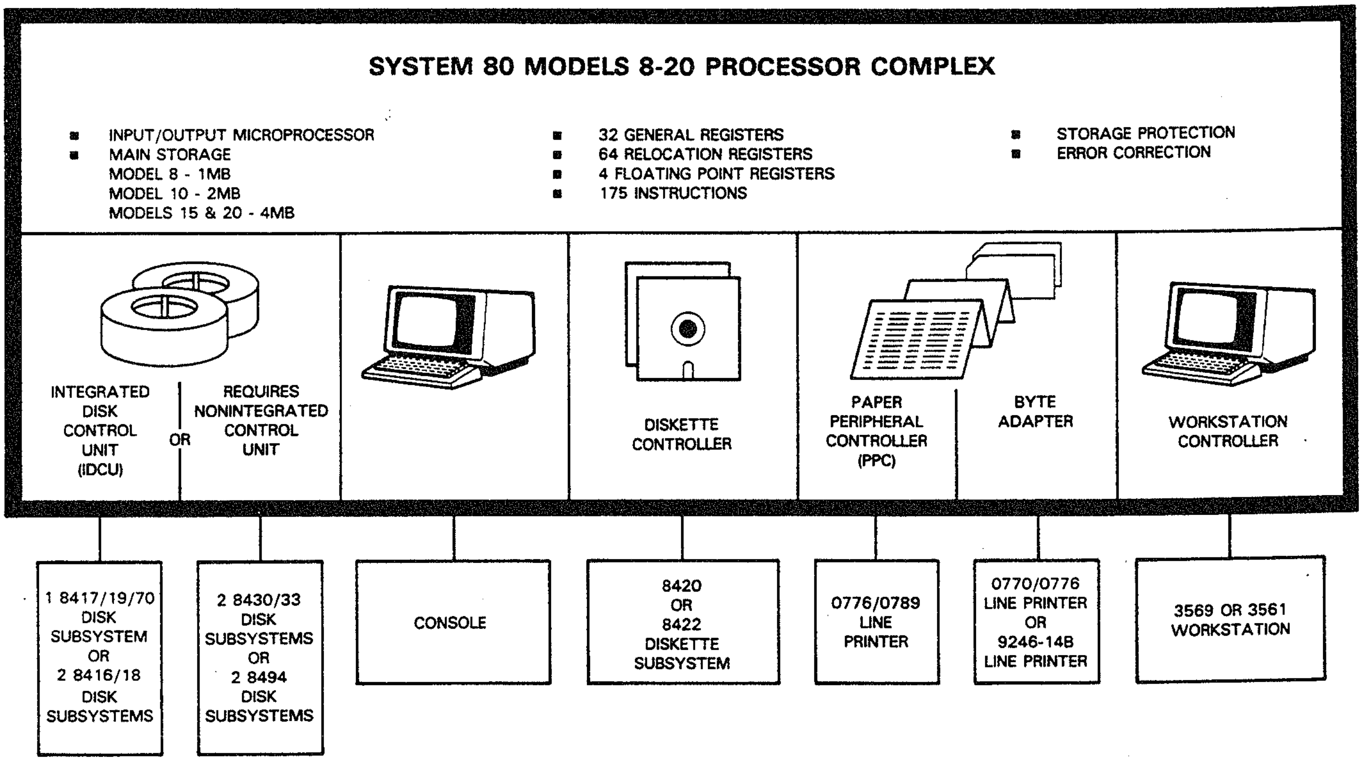
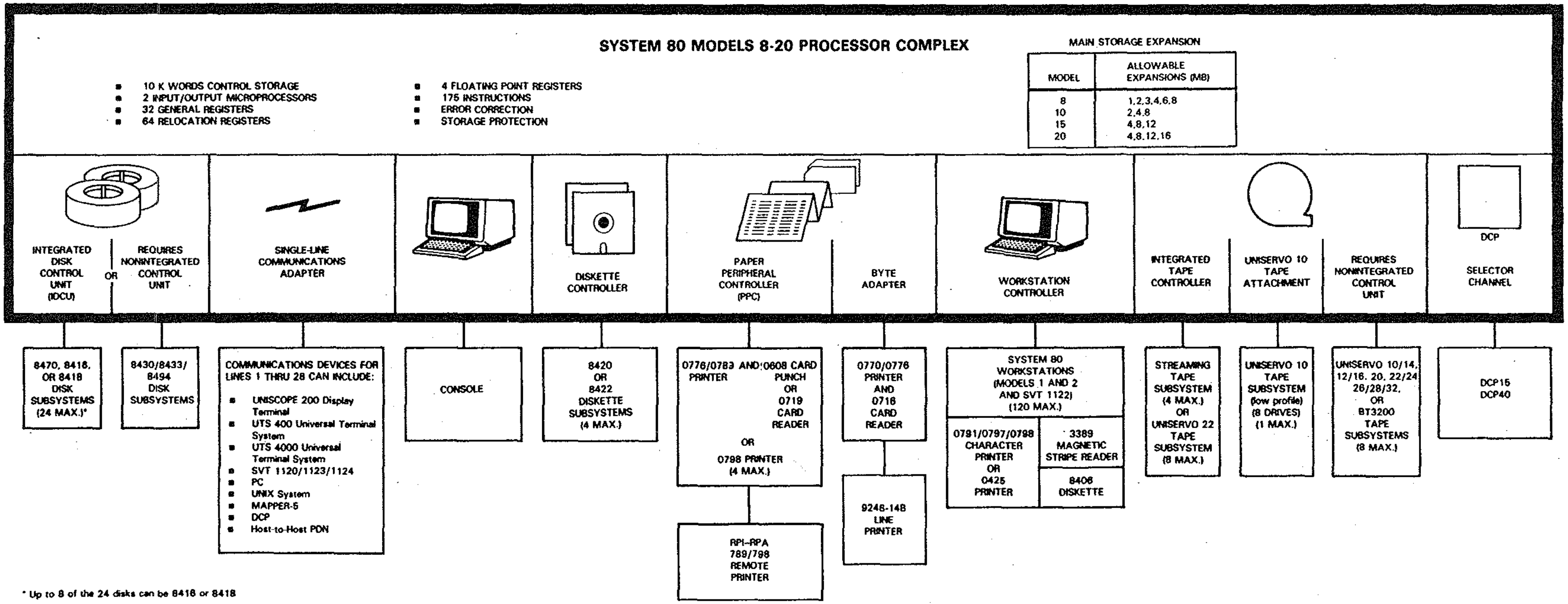

SYSTEM 80 MODELS 8-20 PROCESSOR COMPLEX
▪ INPUT/OUTPUT MICROPROCESSOR
▪ MAIN STORAGE
MODEL 8 - 1MB
MODEL 10 - 2MB
MODELS 15 & 20 - 4MB
▪ 32 GENERAL REGISTERS
▪ 64 RELOCATION REGISTERS
▪ 4 FLOATING POINT REGISTERS
▪ 175 INSTRUCTIONS
▪ STORAGE PROTECTION
▪ ERROR CORRECTION
INTEGRATED DISK CONTROL UNIT (IDCU)
OR
REQUIRES NONINTEGRATED CONTROL UNIT
DISKETTE CONTROLLER
PAPER PERIPHERAL CONTROLLER (PPC)
BYTE ADAPTER
WORKSTATION CONTROLLER
1 8417/19/70 DISK SUBSYSTEM OR 2 8416/18 DISK SUBSYSTEMS
2 8430/33 DISK SUBSYSTEMS OR 2 8494 DISK SUBSYSTEMS
CONSOLE
8420 OR 8422 DISKETTE SUBSYSTEM
0776/0789 LINE PRINTER
0770/0776 LINE PRINTER OR 9246-14B LINE PRINTER
3569 OR 3561 WORKSTATION

Figure 1-1. Minimum System 80 Configuration


SYSTEM 80 MODELS 8-20 PROCESSOR COMPLEX
10 K WORDS CONTROL STORAGE
2 INPUT/OUTPUT MICROPROCESSORS
32 GENERAL REGISTERS
64 RELOCATION REGISTERS
4 FLOATING POINT REGISTERS
175 INSTRUCTIONS
ERROR CORRECTION
STORAGE PROTECTION
INTEGRATED DISK CONTROL UNIT (IDCU)
OR
REQUIRES NONINTEGRATED CONTROL UNIT
SINGLE-LINE COMMUNICATIONS ADAPTER
DISKETTE CONTROLLER
PAPER PERIPHERAL CONTROLLER (PPC)
BYTE ADAPTER
WORKSTATION CONTROLLER
INTEGRATED TAPE CONTROLLER
UNISERVO 10 TAPE ATTACHMENT
REQUIRES NONINTEGRATED CONTROL UNIT
DCP
SELECTOR CHANNEL
8470, 8416, OR 8418 DISK SUBSYSTEMS (24 MAX.)*
8430/8433/8494 DISK SUBSYSTEMS
COMMUNICATIONS DEVICES FOR LINES 1 THRU 28 CAN INCLUDE:
UNISCOPE 200 Display Terminal
UTS 400 Universal Terminal System
UTS 4000 Universal Terminal System
SVT 1120/1123/1124
PC
UNIX System
MAPPER-5
DCP
Host-to-Host PDN
CONSOLE
8420 OR 8422 DISKETTE SUBSYSTEMS (4 MAX.)
0776/0783 AND 0608 CARD PUNCH OR 0719 CARD READER
PRINTER
OR
0798 PRINTER (4 MAX.)
RPI-RPA 789/798 REMOTE PRINTER
0770/0776 PRINTER AND 0716 CARD READER
9246-148 LINE PRINTER
SYSTEM 80 WORKSTATIONS (MODELS 1 AND 2 AND SVT 1122) (120 MAX.)
0791/0797/0798 CHARACTER PRINTER OR 0425 PRINTER
3389 MAGNETIC STRIPE READER
8406 DISKETTE
STREAMING TAPE SUBSYSTEM (4 MAX.) OR UNISERVO 22 TAPE SUBSYSTEM (8 MAX.)
UNISERVO 10 TAPE SUBSYSTEM (low profile) (8 DRIVES) (1 MAX.)
UNISERVO 10/14, 12/16, 20, 22/24 26/28/32, OR BT3200 TAPE SUBSYSTEMS (8 MAX.)
DCP15
DCP40

MAIN STORAGE EXPANSION

MODEL	ALLOWABLE EXPANSIONS (MB)
8	1,2,3,4,6,8
10	2,4,8
15	4,8,12
20	4,8,12,16

* Up to 8 of the 24 disks can be 8416 or 8418

Figure 1-2. Maximum System 80 Configuration

1.2. Central Hardware

The basic System 80 models 8-20 hardware consists of the processor complex, the console, disk and diskette subsystems, and a printer subsystem. It also includes the standard processor features and input/output controls shown in Figure 1-1. The central hardware for a maximum system can consist of the processor complex, an I/O expansion cabinet, and the additional integrated processor features and I/O controls shown in Figure 1-2.

The central processor complex contains all processors, I/O channels, and integrated peripheral system controls. The following central processor components are provided:

- High-performance control storage
- Expandable main storage
- 175 basic instructions, including instructions for floating-point, decimal, and system control
- 32 registers for program and system use, plus 4 floating-point registers and 64 relocation registers
- System control processor
- For model 8 systems, an input/output microprocessor (maximum of 2) supporting:
  - Diskette controller
  - Workstation controller
  - Paper peripheral controller
  - Integrated tape controller
  - Single-line communications adapter (SLCA)
- For model 8 systems, a channel controller (maximum of 2) supporting :
  - One byte MUX channel (for console, printer, and card reader support)
  - Up to five integrated selector channels (for disk, printer, tape, and distributed communications processor (DCP) support)
- For model 10/15/20 systems, a selector channel (maximum of 6) for disk, printer, tape, and DCP support
- For model 10/15/20 systems, an input/output processor (IOP) (maximum of 2) that provides support equivalent to the IOMP on the model 8

1.2.1. Control Storage

Control storage contains the macroinstructions that control operations of the central processor. The CPU executes and controls instructions and processes I/O interrupts, interval timer activities, and general interrupts.

1.2.2. Main Storage

The main storage unit stores data and commands. The minimum main storage values are shown in Figure 1-1. The main storage can be expanded (by adding another main storage unit). The increments vary by processor as shown in Figure 1-2.

Main storage provides error correction for single bit errors and error detection for double bit errors. It also provides a protection scheme for data and programs using keys. Main storage is accessed directly by the central processor (for programs), the system control processor, the input/output microprocessor, and the channel controller.

1.2.3. System Instructions

The system instructions control the operation of the central processor. A repertoire of 175 basic instructions, 31 of which are privileged, is provided with the system. The privileged instructions are used for system control and input/output operations. Nonprivileged instructions are used for general control, data manipulation, decimal operations, list processing, and floating-point operation.

1.2.4. System Registers

The system registers are provided for accumulating, addressing, and controlling operations and data within the system. The 32 general purpose registers may be used as accummulators in general arithmetic and logical operations. They are also used as base address and index registers in arithmetic and logical operations and in address generation. The 16 control registers are used to specify whether or not an operation can be performed in the system, or they may be used to provide special information required by a system facility. The four floating-point registers are available for floating-point operations when the floating-point instruction feature is included in the system. Main storage space is fully used by the system through dynamic storage relocation and the 48 relocation registers. Automatic relocation of all programs in main storage allows maximum loading of programs.

1.2.5. System Control Processor (SCP)

The system control processor (SCP) functions independently of the central processor to control the console, control panels, system initialization, maintenance, and diagnostic functions, system recovery, and remote maintenance.

1.2.6. Input/Output Microprocessor (IOMP) or Input/Output Processor (IOP)

The IOMP (for model 8) or IOP (for model 10/15/20) initiates, directs, and monitors the transfer of data between main storage and some of the peripheral I/O devices. The IOMP/IOP accepts instructions from the central processor (through user and system software), from the workstation, and from operator controls on the devices. 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 IOMP/IOP is used to support low-speed peripheral subsystems, such as those attached to a diskette controller, workstation controller, paper peripheral controller, or integrated magnetic tape controller. Up to eight controllers and a UNISERVO® 10 tape subsystem (low profile) can be attached to an IOMP/IOP through the shared direct memory access interface (SDMA). Data transfer rate with SDMA is 200K bytes/second.

Each IOMP/IOP can also support a single-line communications adapter (SLCA) with up to 14 communications lines. The multiple line communications multiplexer (MLCM) is the interface between the SLCA and the IOMP/IOP. MLCM microcode and IOMP/IOP hardware control buffer pool management, data chaining, and command chaining. The MLCM transfers data at the rate of 128K bytes/second.

The IOMP or IOP supports:

- Diskette controller
- Workstation controller
- Paper peripheral controller
- Integrated tape controller
- Single-line communications adapter

The following subsections describe these hardware components.

UNISERVO is a registered trademark of Unisys Corporation.

Diskette Controller

The 8420 or 8422 diskette subsystem is attached to the system through the diskette controller. The controller is capable of controlling up to four diskette drives - either two autoload drives and two manual drives or four manual drives. Your system must include at least one. The autoload type provides an automatic loading capacity of up to 20 standard diskettes. The controller uses microprocessors and interfaces the IOMP/IOP through shared direct memory access.

Workstation Controller

A workstation controller attaches up to eight workstations to the system. Additional workstation controllers can be attached to expand the number of workstations in the system to a maximum of 120.

The workstation controller accommodates three types of workstations on the System 80 models 8-20 - the model 1, the model 2, and the SVT 1122. All types may be located up to 5000 cable feet (1500 meters) from the processor cabinet. The controller uses dedicated buffers to allow data transfers at a rate of 19,200 bits/second. The model 2 workstation can directly connect to an 0791, 0797, 0798, or 0425 printer.

Paper Peripheral Controller

The paper peripheral controller attaches various peripheral printers and card processors to the system through the IOMP/IOP. The controller can support one or two printers and two card processors for a maximum of four devices operating concurrently. The controller accommodates the following types of devices: 0776, 0789, 0798 printers, 0608 card punch, and an 0719 card reader. The total print capacity of the paper peripheral controller is 1500 lines per minute. The System 80 models 8-20 support up to 12 paper peripheral controllers.

Integrated Tape Controller

The integrated tape controller connects up to four streaming tape subsystems or eight UNISERVO 22 tape drives to the system. The total number of drives the integrated controller can support is eight and only one controller is allowed per system.

Single-Line Communications Adapter (SLCA)

The SLCA provides the system with communications capabilities. Models 8-20 can support up to 14 SLCAs. You can have 14 low- or medium-speed half-duplex, 7 high-speed full-duplex, or 12 high-speed half-duplex SLCAs. Since your system can include two IOMPs or IOPs, each supporting 14 SLCAs, the total number of communications lines your system can have is 28. Each line can accept devices operating at speeds of up to 56 kilobits/second.

SLCAs connect remotely located workstations, terminals, and all other communications devices to the system. They perform special character recognition, integrity checks, data transfer control between main storage and devices, and other required control functions.

1.2.7. Channel Controller (Model 8 Only)

In model 8 systems, the channel controller initiates, directs, and monitors the transfer of data between main storage and those devices not connected via the IOMP. The system can include two channel controllers. The first channel controller supports the byte MUX channel and three integrated selector channels. A second channel controller can support up to two integrated selector channels.

Byte MUX Channel

The byte MUX channel connects the console to the system. The console includes two diskette drives for system initialization and maintenance. Via the byte adapter, the byte MUX channel also connects the 0770/0776 line printer and the 0716 card reader. The system can support only one byte MUX channel. In turn, the byte MUX channel can connect up to eight devices at a maximum data rate of 70 KB/second.

Integrated Selector Channel

An integrated selector channel controls high-speed I/O devices, such as disks, magnetic tapes, high-speed buffered printers, and distributed communications processors. It operates in burst mode with the subsystem for data transfers. There is a maximum of five integrated selector channels per model 8 system.

The 8416, 8417, 8418, 8419, and 8470 disk subsystems are connected to the integrated selector channel via an integrated disk control unit (IDCU). Each selector channel can support no more than three IDCUs (a maximum of six IDCUs per system) and 24 disks.

The 8430, 8433, and 8494 disk subsystems are connected to the integrated selector channel via nonintegrated (freestanding) disk control units. The system supports a maximum of 16 subsystems.

The integrated selector channel also connects, via nonintegrated controllers, UNISERVO 10, 12, 14, 16, 20, 22, 24, 26, and 28 tape units; 3200 series tape units; 0770 and 0776 printers; 9246-14B and 9246-25B printers; and DCP 15, 30, 40, and 50 units. The system supports a maximum of eight subsystems in any combination.

1.2.8. Selector Channel (Models 10/15/20 Only)

In model 10/15/20 systems, selector channels initiate, direct, and monitor the transfer of data between main storage and high-speed printer, disk, tape, and the DCP. A minimum system requires one selector channel; a maximum of six selector channels may be configured.

Each model 10/15/20 selector channel can be connected to I/O devices as defined for the model 8 integrated selector channel.

1.3. Input/Output Subsystems

The I/O peripheral subsystems available for use with System 80 include the console, disk and diskette subsystems, workstations, magnetic tape subsystems, printer subsystems, card reader/punch subsystems, and remote communications subsystems.

1.4. OS/3 Operating System

OS/3 is composed of a group of 11 major programs - supervisor, job control, data management, integrated communications access method (ICAM), utility programs, information management system (IMS), language processors, data base management system (DMS), application programs, diagnostic programs, and interactive services. (Figure 1-3).

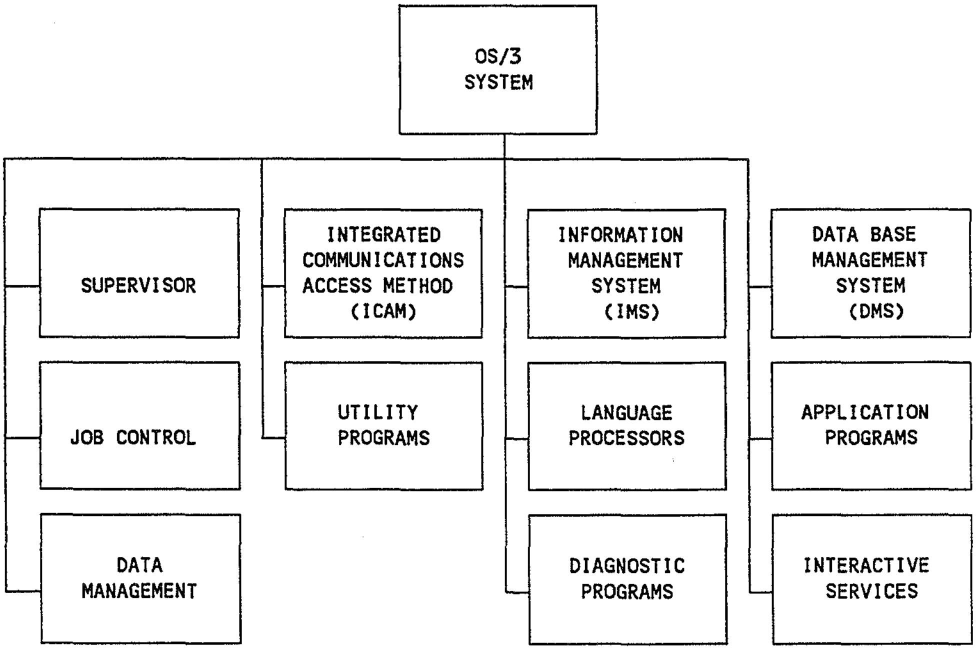


Figure 1-3. OS/3 Components

1.4.1. Supervisor

The supervisor is a resident program of the operating system; it interfaces with the user programs to provide the necessary control for the best use 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 these program elements:

- Interrupt control
- Priority control
- Transient control
- Physical I/O controls
- Resource allocation
- Task control
- Interrupt timer and day clock services
- Program management
- Console workstation management
- File services
- Program error handling
- Cooperative/symbiont operations
- Physical input/output control system (PIOCS)
- Debugging aids

1.4.2. 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 interactively or in batch mode by the programmer through the job control language. These sequenced control statements form the control stream that defines facility requirements of a job and directs the execution of the job. The job control statements, through the job control stream, function as an interface between the programmer and OS/3.

1.4.3. Data Management

Data management provides a consolidated data interface between the hardware-oriented I/O facility (card, tape, printer, and MIRAM disk files) and the user program. The data management facilities consist of logical input/output control stream (IOCS) modules, transient routines, declarative macroinstructions, and imperative macroinstructions.

1.4.4. Integrated Communications Access Method

The integrated communications access method (ICAM) may be used to control local workstations directly connected to the processor and other ICAM-supported terminals connected to the processor by means of communications lines. ICAM is loaded as a symbiont and acts as an extension to the System 80 supervisor.

Application programs communicate with ICAM by means of the following interfaces using imperative and declarative macroinstructions:

- CPI - communications physical interface
- DDI - direct data interface
- STDMCP - standard GET/PUT interface

1.4.5. 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, and 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

1.4.6. Information Management System

The information management system (IMS) is a transaction-oriented application package that 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.

1.4.7. Language Processors

Eight language processors are available with OS/3: basic assembly language (BAL), COBOL, FORTRAN IV, FORTRAN 77, Pascal, report program generator II (RPG II), BASIC, and ESCORT. Language processor input can be on punched cards, disk or diskette files, or magnetic tape; output can be recorded on disk or diskette files or on magnetic tape. All processor output is in a common system output format.

1.4.8. Diagnostic Programs

The diagnostic programs provided with OS/3 are hardware verification and 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.

1.4.9. Data Base Management System

The data base management system (DMS) is a collection of system programs that support 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.

1.4.10. Application Programs

Application programs are those specialized programs that are available to a user but 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).

1.4.11. Interactive Services

Interactive services provides 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
- Menu services
- Interactive dump/restore facility
- ESCORT
- BASIC

1.5. Console and Control Panel

As the system operator, most of the functions and procedures you perform are accomplished at the system console and system control panel. Following is a brief general description of these items. Note that two different system console versions are used; one for the model 8 and another for models 10, 15, and 20. Refer to the *Processor and Central Peripherals Operating Guide* (UP-9608) for detailed information on switch and indicator functions for both system console versions.

1.5.1. System Console

The console consists of a display monitor and a keyboard. The model 8 United States national keyboard is shown in Figure 1-4. Seven additional national keyboard styles are available.

The keyboard is used to communicate with the system via input commands and to select display frames. When you press the ESC key, a command input mode is initiated. You can enter commands by pressing an appropriate key while pressing and holding the ESC key. On the model 8 system, the LOCK/UNLOCK switch must be set to the UNLOCK position to allow command inputs. Table 1-1 summarizes console operations that can be performed with the ESC key. Refer to the *Processor and Central Peripherals Operating Guide* (UP-9608) for more detailed information.

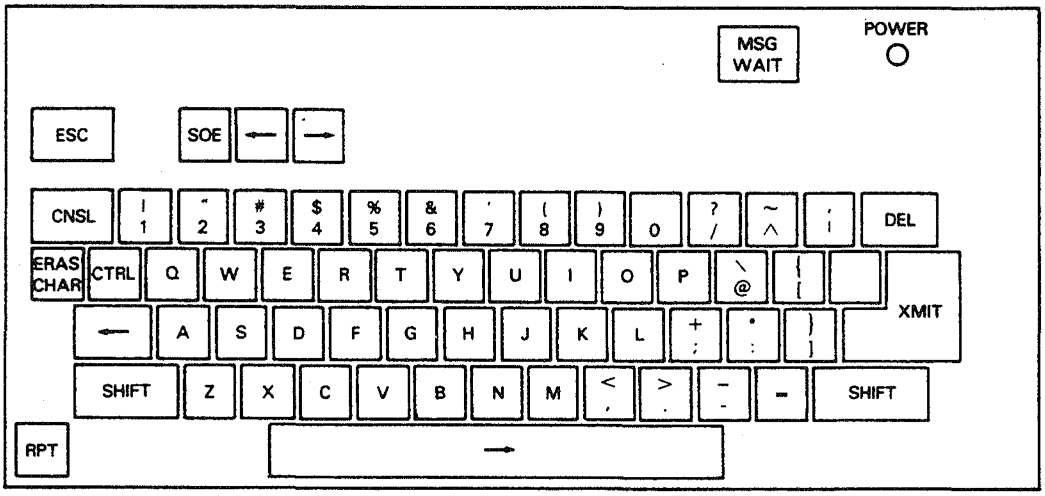


Figure 1-4. Model 8 Console Keyboard

Table 1-1. Console Control Selections

Key Input	Function
ESC and R	Switches processor from stop state to operational state so that software instructions can be executed.
ESC and S	Switches processor from operational state to stop state.
ESC and C	Puts console in system control mode and displays the configuration control frame. This frame is used to control auto IPL, autorecovery, and remote console modes. It also displays processor configuration data and the state of system control panel and console indicators.
ESC and M	Puts console in system control mode and displays the manual frame. This frame is used to perform manual IPL and IMPL functions, and to debug programs and operations.
ESC and A	Puts console in system control mode and displays the alter/display frame. This frame allows you to display and alter various registers, main storage locations, and the PSW.
ESC and P	Puts console in maintenance console mode and displays the service frame. This frame provides numerous error logging and maintenance-related functions.
ESC and T	Puts console in remote maintenance mode and displays the teleprocessing frame.
CNSL	Puts console in operator console mode and displays the console frame. This frame is used to exchange information between the operator and the operating system.

1.5.2. System Control Panel

The system control panel contains controls and indicators used for operator maintenance during system operation. On the model 8, this panel is located on the door of the console table. On models 10, 15, and 20, it is physically attached to the keyboard.

Figure 1-5 shows the model 8 system control panel. The model 10/15/20 system control panel is of a different design but provides similar functionality. Brief descriptions of the panel indicators and switches follow (refer to the *Processor and Central Peripherals Operating Guide* (UP-9608) for more detailed information on this panel).

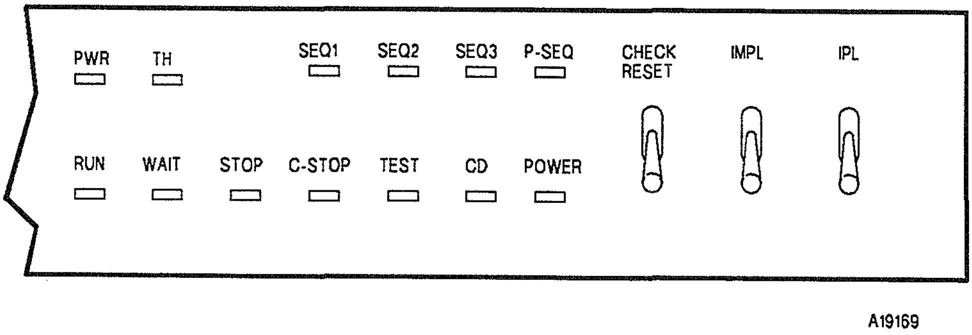


Figure 1-5. Model 8 System Control Panel

Indicator Lights

- PWR

  Indicates a fault in the dc power supply to the system. This light stays lit until the fault is corrected. When this light goes on, the system alarm buzzer also sounds.

- TH

  Indicates that the speed of the cooling fan in the system has dropped below a safe level. This light stays lit until the fan is fixed. When this light goes on, the system alarm buzzer also sounds.

- SEQ1, SEQ2, SEQ3

  Indicates that power has been applied to your system or that the initial microprogram load (IMPL) has begun. Once IMPL and initial program load (IPL) complete, these lights go off. Anytime an error occurs within the system control panel itself during IMPL, IPL, or system operation, these lights go on.

- P-SEQ

  Indicates that you are performing the power-on procedure. This light goes on as soon as you turn on the POWER switch behind the control panel door. It turns off when the P-OK signal is displayed on your console.

- RUN

  Indicates that your system is running jobs.

- WAIT

  Indicates that your system is in a WAIT state.

- STOP

  Indicates that your system is in a STOP state.

- C-STOP

  Indicates that your system is in a CHECK STOP state. This light stays lit until you perform system reset. When this light goes on, the system alarm buzzer also sounds.

- TEST

  Indicates that you are operating the system under other than normal conditions, such as processor STEP mode. This light stays lit until the abnormal condition is cleared.

- CD

  Indicates a remote console is connected. Your system receives the carry detect (CD) signal from the modem and turns this light on.

- POWER

  Indicates that your system is receiving power.

Switches

- CHECK RESET

  Stops the system alarm buzzer.

- IMPL

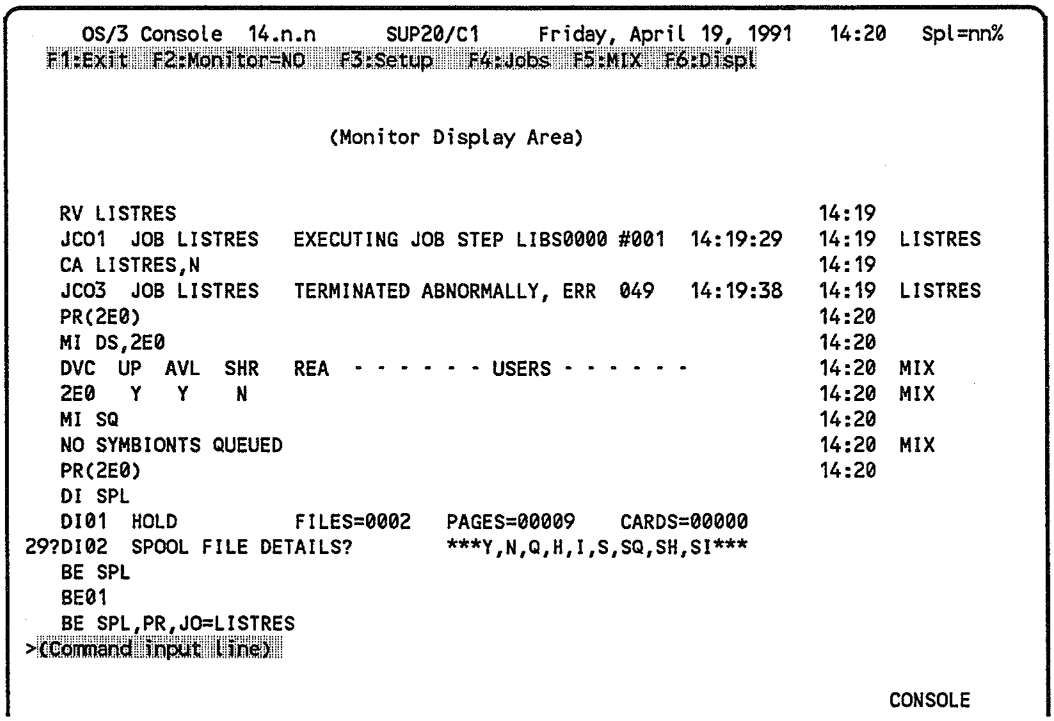
  Starts the service processor (SVP) and console initial microprogram load (IMPL) procedure. Note that additional IMPL functions must be performed using the manual frame display.

- IPL

  Specifies manual (MNL) or automatic (AUTO) initial program load.

1.5.3. System Console Screen

The console screen (Figure 1-6) is the normal operating screen from which you monitor and control system activities.



```
    OS/3 Console  14.n.n      SUP20/C1     Friday, April 19, 1991   14:20   Spl=nn%
  F1:Exit  F2:Monitor=NO  F3:Setup  F4:Jobs  F5:MIX  F6:Displ


                          (Monitor Display Area)


   RV LISTRES                                                     14:19
   JC01  JOB LISTRES   EXECUTING JOB STEP LIBS0000 #001  14:19:29  14:19  LISTRES
   CA LISTRES,N                                                   14:19
   JC03  JOB LISTRES   TERMINATED ABNORMALLY, ERR  049   14:19:38  14:19  LISTRES
   PR(2E0)                                                        14:20
   MI DS,2E0                                                      14:20
   DVC  UP  AVL  SHR   REA  - - - - - - USERS - - - - - -         14:20  MIX
   2E0   Y    Y    N                                              14:20  MIX
   MI SQ                                                          14:20
   NO SYMBIONTS QUEUED                                            14:20  MIX
   PR(2E0)                                                        14:20
   DI SPL
   DI01  HOLD          FILES=0002   PAGES=00009    CARDS=00000
29?DI02  SPOOL FILE DETAILS?        ***Y,N,Q,H,I,S,SQ,SH,SI***
   BE SPL
   BE01
   BE SPL,PR,JO=LISTRES
>(Command Input Line)

                                                                        CONSOLE
```

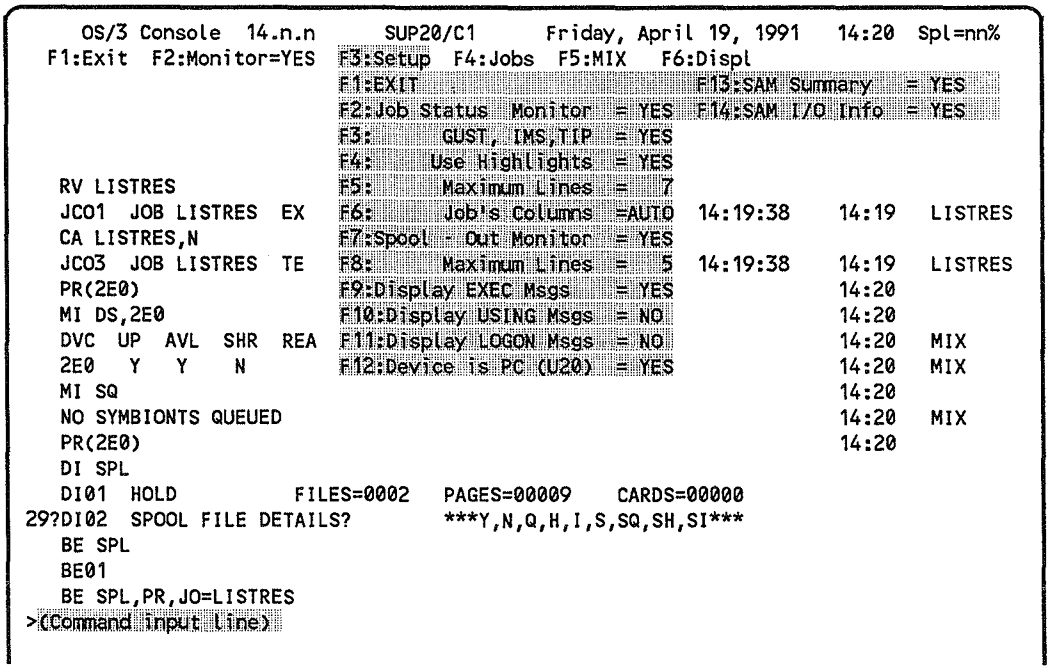
Figure 1-6. Sample Console Screen

The console screen is displayed at the completion of a successful system initialization sequence. The screen is divided from top to bottom into the following areas:

- Console header line (line 1, top of screen)
- Menu bar (line 2)
- Monitor display area (lines 3....n)
- Message display area (end of monitor area to line 23)
- Command input line (line 24)
- Status line (line 25)

The header line identifies the display as the console screen. It also identifies the system software level, supervisor name, ICAM network name (when ICAM is loaded), the day of the week, the current date and time, and the percentage of spooling being used. The time display is updated once every minute.

The menu bar lists the function keys (F1 through F6) that you use to select and display the pull-down menus associated with common command execution and system activity monitor control. Pressing one of the function keys shown in the menu bar causes the menu associated with that function key to appear directly below it on the screen (if applicable). This display area is called a pull-down menu window. (See the example shown in Figure 1-7.)



```
    OS/3 Console  14.n.n     SUP20/C1     Friday, April 19, 1991   14:20  Spl=nn%
 F1:Exit  F2:Monitor=YES  F3:Setup  F4:Jobs  F5:MIX   F6:Displ
                          F1:EXIT                     F13:SAM Summary   = YES
                          F2:Job Status  Monitor = YES F14:SAM I/O Info  = YES
                          F3:        GUST, IMS,TIP  = YES
                          F4:       Use Highlights  = YES
  RV LISTRES              F5:        Maximum Lines  =   7
  JC01  JOB LISTRES  EX   F6:        Job's Columns  =AUTO 14:19:38    14:19   LISTRES
  CA LISTRES,N            F7:Spool - Out Monitor    = YES
  JC03  JOB LISTRES  TE   F8:        Maximum Lines  =   5 14:19:38    14:19   LISTRES
  PR(2E0)                 F9:Display EXEC Msgs      = YES             14:20
  MI DS,2E0               F10:Display USING Msgs    = NO              14:20
  DVC  UP  AVL  SHR  REA  F11:Display LOGON Msgs    = NO              14:20   MIX
  2E0   Y   Y    N        F12:Device is PC (U20)    = YES             14:20   MIX
  MI SQ                                                               14:20
  NO SYMBIONTS QUEUED                                                 14:20   MIX
  PR(2E0)                                                             14:20
  DI SPL
  DI01  HOLD         FILES=0002   PAGES=00009   CARDS=00000
29?DI02  SPOOL FILE DETAILS?      ***Y,N,Q,H,I,S,SQ,SH,SI***
  BE SPL
  BE01
  BE SPL,PR,JO=LISTRES
>(Command input line)
```

Figure 1-7. Example of a Pull-Down Menu Window

The pull-down menus list the function keys associated with each operation or command that can be issued from that menu. To initiate an operation or to issue a command, press the appropriate function key displayed in the menu. The pull-down menu usually disappears after you complete your selection. There are a few choices that display a second pull-down menu. In all cases, the F1 key exits the current menu. The F1 function key also toggles the menu bar display off and on whenever pull-down menus are not being displayed.

User-defined function key operations are not recognized by the system when they conflict with the menu bar definitions. When such conditions exist, you should exit the menu bar. The system will then acknowledge all your function key definitions with the exception of the F1 key, which remains the same. The functions you can perform from the menu bar and their associated pull-down menus are listed in Table 1-2. The function key values shown in the first column of Table 1-2 represent the default value assigned to each key.

Table 1-2. Console Screen Function Key Assignments

Function Keys Menu Bar / Pull-Down Menu	Description
F1:Exit	Exits the current pull-down menu being displayed. Also turns (toggles) the menu bar display on and off when pull-down menus are not being displayed.
F2:Monitors=YES	Turns the monitor display area on (YES) and off (NO)
F3:Setup	Displays the Setup pull-down menu
F1:Exit	Exits the Setup pull-down menu
F2:Job Status Monitor =YES	Turns (toggles) the Jobs monitor on (YES) and off (NO)
F3:GUST, IMS, TIP =YES	Includes (YES) or excludes (NO) the TIP, IMS, and GUST jobs in the Jobs monitor
F4:Use Highlights =YES	Turns (toggles) display highlighting on (YES) and off (NO)
F5:Maximum Lines = 7	Defines the number of jobs (lines and header) displayed in the display area of the screen. The default value is 7 but can be increased or decreased by pressing the F5 function key to pull down a second menu. From the second menu, press the function key (F2-F22) that corresponds the new value you want specified. That actual number of lines displayed may be dynamically reduced if additional screen lines are required for other uses
F6:Job's Columns =AUTO	Automatically divides the display area into the appropiate number of columns based on the number of jobs. The display can be manually divided into 1- to 3-columns by specifying the values 1, 2 ,or 3

continued

Table 1-2. Console Screen Function Key Assignments (cont.)

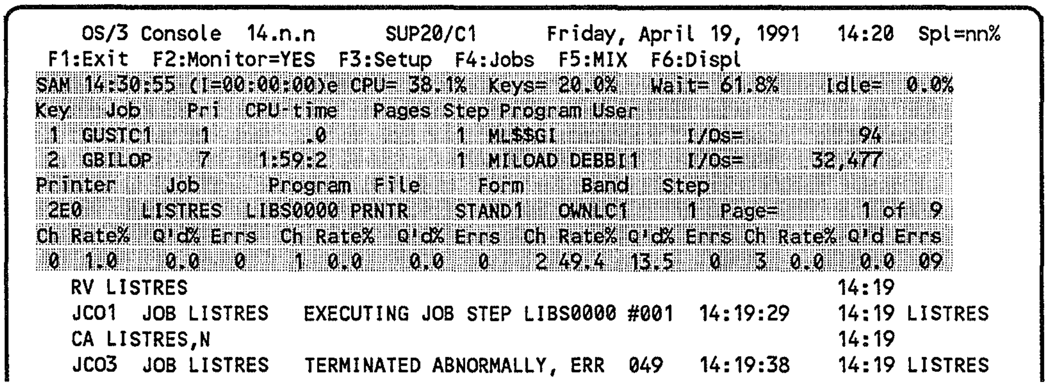
Function Keys Menu Bar / Pull-Down Menu	Description
F7:Spool - Out Monitor =YES	Turns (toggles) the output writer (spooling) monitor on (YES) and off (NO)
F8:Maximum Lines = 5	Defines the number of spooling jobs (lines and header) displayed in the monitor display area of the screen. The default value is 5 but can be increased or decreased by pressing the F8 function key to pull down a second menu. Select the Function key (F2 - F22) in the second menu that correspond to the new value and press that key
F9:Display EXEC Msgs =YES	Displays (YES) or inhibits (NO) the display of EXEC (JC01) and terminated normally (JC02) messages in the display area of the screen
F10:Display USING Msgs = NO	Displays (YES) or inhibits (NO) the display of USING messages in the display area of the screen
F11:Display LOGON Msgs = NO	Displays (YES) or inhibits (NO) the display of LOGON messages in the display area of the screen
F12:Device is PC(U20) =YES	Ignore. Not used on model 8-20 systems
F13:SAM Summary =YES	Toggles the display of the SAM summary line on (YES) and off (NO). When YES, summary line displayed if SAM active
F14:SAM I/O Info =YES	Toggles display of SAM I/O class data display information on (YES) and off (NO). When YES, information is displayed if SAM active and I/O class monitoring and display enabled
F4:Jobs	Displays the Jobs command pull-down menu
F1:Exit	Exits the Jobs command pull-down menu
F2:CJ Cancel Normal-No Dump	Selects the CJ command to soft cancel job; dump not taken. Also displays second pull-down menu in which you indicate the jobs to be canceled
F3:CJ Cancel Normal- Dump	Selects the CJ command to soft cancel job; dump taken. Also displays second pull-down menu in which you indicate the jobs to be canceled
F4:CA Cancel Hard -No Dump	Selects the CA command to hard cancel job; dump not taken. Also displays second pull-down menu in which you indicate the jobs to be canceled
F5:CA Cancel Hard - Dump	Selects the CA command to hard cancel job; dump taken. Also displays second pull-down menu in which you indicate the jobs to be canceled

continued

Table 1-2. Console Screen Function Key Assignments (cont)

Function Keys Menu Bar / Pull-Down Menu	Description
F6:PA Pause	Selects the PAUSE command to suspend processing of a job. Also displays second pull-down menu in which you indicate the jobs to be paused
F7:GO Un-Pause	Selects the GO command to reactivate a suspended job. Also displays second pull-down menu in which you indicate the jobs to be reactivated
F5:MIX	Displays the MIX command pull-down menu
F1:Exit	Exits the MIX command pull-down menu
F2:DA	Issues a MIX DA command to display specific job/symbiont information
F3:DS	Issues a MIX DS command to display peripheral device information
F4:EL	Issues a MIX EL command to display error log information
F5:FR	Issues a MIX FR command to display information about unused areas (free regions) of main memory
F6:MM	Issues a MIX MM command to display information about every main memory area
F7:SC	Issues a MIX SC command to display information about shared code modules currently residing in main memory
F8:SI	Issues a MIX SI command to display information pertaining to system operations
F9:SQ	Issues a MIX SQ command to display information pertaining to outstanding symbiont requests
F10:VI	Issues a MIX VI command to display the volume serial number for each disk and tape volume currently mounted
F6:Displ	Displays the Display command pull-down menu
F1:Exit	Exits the Displ pull-down menu
F2:Spooling	Issues a DI SPL command to display the status of completed spool files
F3:Job Status	Issues a DI JS command to display the status of jobs in the system for processing
F4:Job Queue	Issues a DI JBQ command to display the contents of all job scheduling queues
F5:Active Spool	Issues a DI SPL command to display the status of active spool files

The monitoring display area of the console screen starts at display line 3 and occupies a variable amount of screen space based on the number of monitors turned on and the number of items being tracked. The monitoring area is displayed whenever F2=YES. The information displayed is limited to the specific monitors turned on. In the following example, the monitoring display area uses eight lines to display status information for SAM, jobs GUSTC1 and GBILOP, and the output writer.



```
    OS/3 Console  14.n.n      SUP20/C1      Friday, April 19, 1991   14:20  Spl=nn%
  F1:Exit  F2:Monitor=YES  F3:Setup  F4:Jobs  F5:MIX  F6:Displ
SAM 14:30:55 (I=00:00:00)e CPU= 38.1%  Keys= 20.0%   Wait= 61.8%    Idle=  0.0%
Key  Job    Pri  CPU-time   Pages Step Program User
 1  GUSTC1   1         .0         1  ML$$GI             I/Os=          94
 2  GBILOP   7    1:59:2          1  MILOAD DEBBI1      I/Os=      32,477
Printer    Job      Program  File     Form     Band   Step
 2E0     LISTRES  LIBS0000 PRNTR    STAND1   OWNLC1     1  Page=        1 of  9
Ch Rate%  Q'd% Errs  Ch Rate%  Q'd% Errs  Ch Rate% Q'd% Errs Ch Rate% Q'd Errs
 0  1.0    0.0   0    1  0.0    0.0   0    2 49.4  13.5   0   3  0.0   0.0  09
   RV LISTRES                                                      14:19
   JCO1  JOB LISTRES   EXECUTING JOB STEP LIBS0000 #001  14:19:29   14:19 LISTRES
   CA LISTRES,N                                                    14:19
   JCO3  JOB LISTRES   TERMINATED ABNORMALLY, ERR  049   14:19:38   14:19 LISTRES
```

The message display area occupies the space from the bottom of the monitoring display area to the command input line. Messages are scrolled up from the bottom of the display. Messages contain a 13-character prefix that includes:

- The sequential message number
- The originating program name
- Special characters, such as:
  - ? for messages requiring an operator response
  - * for ACTION "GO" required message

You enter commands from the command input line near the bottom of the screen. Pressing the XMIT key enters your input.

Section 2
System Power-On and Initialization Procedures

Use the procedures in this section to:

- Power on and initialize your system from a cold startup.
- Automate supervisor loading.
- Reinitialize your system after the occurrence of an unrecoverable error or when you want to load a different supervisor.

2.1. General Information

Main power to the equipment comprising your system is controlled from circuit breakers in the main power distribution panel for your site. The circuit breakers at the rear of the central equipment complex cabinets and the peripheral cabinets are usually left in the ON position when the equipment is powered off. Operating power to these cabinets is controlled by the ON/OFF switches on the operator panels of the equipment. There may be times, however, when site requirements or emergency situations make it necessary to remove main power by use of the cabinet circuit breakers. Therefore, if the power-on procedure (2.2) in this section fails to apply power to your equipment, check to make certain that the cabinet circuit breakers and the circuit breakers in the main power distribution panel are set ON.

Each time operating power is removed from and reapplied to the system, it is necessary to perform the IMPL (initial microprogram load) and the IPL (initial program load) procedures. The IMPL procedure loads your system's control storage with the microcode from the diskette mounted in the diskette drive behind the processor cabinet door. This drive is reserved for the microcode diskette; the diskette should be mounted at all times. Once loaded, the microcode stays in control storage until you turn the system off or you choose to re-IMPL. The IMPL process is automatically performed whenever you power on your system.

The IPL procedure loads the resident portion of your operating system software. It can be initiated automatically or interactively by setting the IPL switch on the system control panel to the AUTO or MNL position. In the AUTO position, IPL automatically follows IMPL without operator intervention. In the MNL position, the system waits for you to manually initiate IPL after IMPL completes.

Supervisor loading usually requires operator intervention. That is, you tell the system what supervisor you want loaded. Then you define the characteristic and the load specifications for your supervisor by answering a series of menu driven questions. The system loads the supervisor after you input your specifications. Once your system is up an running, you have the option of automating the supervisor load process (2.3). You do this through interactive services. You simply log on to interactive services, call the CFIG program, and complete the information requested by the ABP (automatic boot procedure) menu. The CFIG program creates a file called $Y$CONFIG which defines the load criteria and specifications for your supervisor. From this point on, the system automatically loads your supervisor any time an automatic IPL is performed. The automated supervisor load process is set up so that you can interrupt it and load an alternate supervisor or opt for the default supervisor displayed in the menu portion of the IPL screen. However, once you interrupt the automatic procedure you must manually complete the loading process.

There are two day-to-day operations where system initialization is required. They are:

1. When a nonrecoverable error occurs while the system is running (such as a CHECK STOP, when the C-STOP indicator lights)

2. When you want to use a different supervisor (possibly on another load device).

In both of these situations, you must initialize the system (2.4). It is not automatic.

When you perform both the IMPL and IPL procedures, you're completely reinitializing the system. This isn't always required. In fact, more often than not, the only procedure you need is the IPL procedure. For instance, when you need an alternate supervisor, all you're changing is the resident portion of OS/3. Since you're not changing the microcode, it isn't necessary to IMPL.

Similarly, many nonrecoverable errors only require you to re-IPL. You rarely have to IMPL once the system is powered on. When a nonrecoverable error does occur, the system displays an error message on the manual frame, or lights an indicator on the processor cabinet. Refer to the error message in the *OS/3 System Messages Reference Manual*, 7004 5190. This manual explains what you should do. If it tells you to reboot the system, it still might only require the IPL procedure. However, it's a good rule of thumb to perform both the IMPL and IPL procedures in this case. The IMPL procedure is easy and fast and ensures that the system is properly loaded.

If an error occurs during either the IMPL or IPL procedure, always start again with the IMPL procedure.

2.2. System Power On and Initialization - Cold Start

Before proceeding, decide whether you want to perform an automatic or manual IPL. And, decide if you want to automate supervisor loading.

To perform an automatic IPL, set the IPL switch on the system control panel to the AUTO position. At the end of IMPL, the system automatically initiates the IPL process. The load device is the one defined in the configuration frame.

To perform a manual IPL, set the IPL switch to the MNL position. At the end of IMPL, the system displays the configuration frame allowing you to define an alternate load device.

To automate supervisor loading, initialize your system as described in the procedure that follows. After you have the system up and running, create the $Y$CONFIG file as described section 2.3. The creation of the $Y$CONFIG file automatically loads you supervisor whenever an auto-IPL is performed.

Proceed as follows to power on your system:

Note: *The power circuit breakers located at the rear of the CEC and peripheral cabinets are usually left in the ON position. This allows operating power for the equipment to be controlled from the power on / off switch on the operator panel of each unit of equipment. Perform steps 1 through 3 if setting the cabinet circuit breakers off is a site procedure for powering down the equipment.*

1. Set the MAIN POWER circuit breaker at the rear of the processor cabinet to the ON position.

2. Set the MAIN POWER circuit breaker at the rear of the I/O cabinet to the ON position.

3. Set the circuit breaker at the rear of the I/O expansion cabinet (when configured) to the ON position.

Caution

When performing step 4, it is recommended that the data files stored on your disk subsystems are protected before powering on the system. Otherwise, a portion of the recorded data may be accidently erased.

4. Power on peripheral devices (except diskette subsystems and workstations) according to the power-on procedures described in their respective operating guides. (Diskette subsystems power-on with CEC cabinets; the workstations are powered on as required by their users.)

5. Make certain that the POWER ON/OFF switch on the console monitor is in the ON (depressed) position.

6. Set the system LOCK/UNLOCK switch to the UNLOCK position (Model 8 only)

7. Make certain that the IPL switch is set for the type of IPL you plan to perform:

   - AUTO (automatic) position to IPL from load device defined in the configuration frame.

   - MNL (manual) position to IPL from an alternate load device.

8. Set the system POWER ON/OFF switch to the ON position.

   Observe the following:

   - The SEQ1, SEQ2, SEQ3, and P-SEQ lights on the system control panel turn on as the system steps through the power sequencing process.

   - The P-SEQ light turns off and the PWR light turns on when power sequencing is complete.

   - The console displays the following messages indicating that the IMPL procedure is in progress:

     ```
     HEALTH CHECK RUNNING
     BPU IMPL W08START
     BPU IMPL W09 START
     DMUX IOML START
     ```

   - The SEQ1, SEQ2, and SEQ3 lights turn off when the IMPL procedure completes.

   - If the IPL switch is set to position MNL, the STOP and POWER lights on the system control panel light and then the configuration frame appears on the console monitor. Go to step 9 to manually initiate the IPL process.

   - If the IPL switch is set to position AUTO, the system automatically initiates the IPL procedure. The system turns on the RUN, WAIT, and POWER lights on the system control panel and then displays the OS/3 initial program load screen console monitor.) Go to step 10.

Note: *If the system encounters an error during the automatic IPL procedure, it displays the configuration display frame. The cause of the error is displayed as a message in the upper-right corner of the frame. Go to step 9 to manually initate the IPL procedure.*

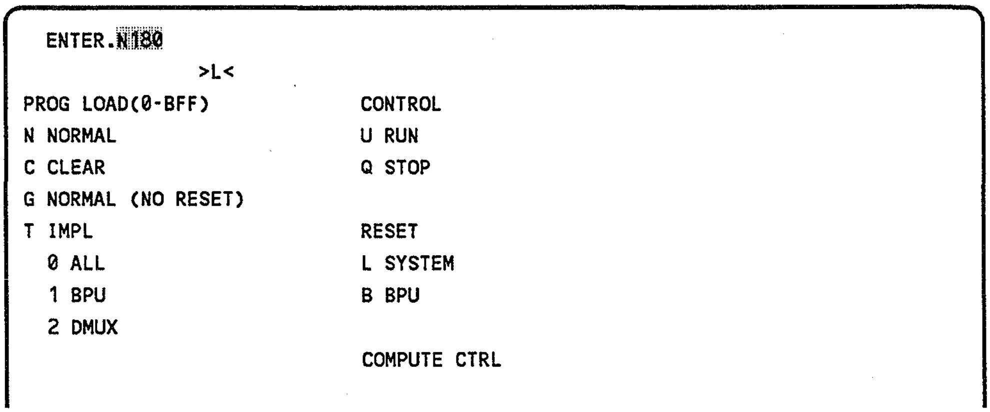
9. Proceed as follows to manually initiate IPL:

   a. Press the ESC key, then the M key to display the manual frame.

   b. Key in the character L and then press the XMIT key reset the system.

   c. Key in the character N (normal program load option) followed by the 3-digit address of the load device where your supervisor resides. (If you are loading from an autoload diskette, substitute the G option for N to prevent ejection of the diskette.)

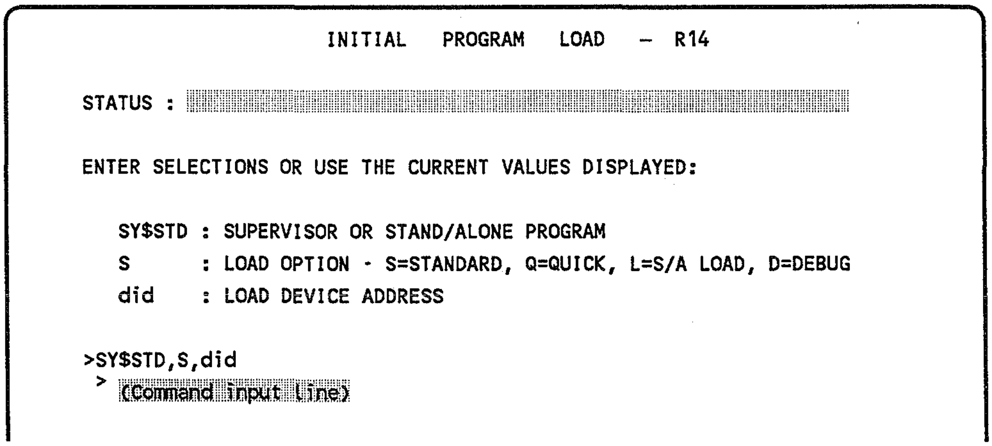
   Example: Your supervisor resides on a disk drive with the address of 180. Complete the manual frame as follows:



```
  ENTER.N180
                >L<
PROG LOAD(0-BFF)              CONTROL
N NORMAL                      U RUN
C CLEAR                       Q STOP
G NORMAL (NO RESET)
T IMPL                        RESET
  0 ALL                       L SYSTEM
  1 BPU                       B BPU
  2 DMUX
                              COMPUTE CTRL
```

   d. Press the XMIT key. The OS/3 Initial Program Load (IPL) screen replaces the manual screen on the console monitor. Continue with step 10.

10. Observe the OS/3 initial program load screen.



```
                    INITIAL   PROGRAM   LOAD   -  R14

   STATUS :

   ENTER SELECTIONS OR USE THE CURRENT VALUES DISPLAYED:

      SY$STD : SUPERVISOR OR STAND/ALONE PROGRAM
      S      : LOAD OPTION - S=STANDARD, Q=QUICK, L=S/A LOAD, D=DEBUG
      did    : LOAD DEVICE ADDRESS

   >SY$STD,S,did
    > (Command input line)
```

a. If you automated supervisor loading, the following message is displayed on the status line of the IPL screen:

```
IPL20 AUTO IPL IN XX SECONDS; DEPRESS MESSAGE WAIT TO HALT
```

To let the autoload process take place, take no action. The system automatically loads the supervisor defined in the $Y$CONFIG file when the countdown period reaches 0 (zero). System initialization is complete and the system is ready to process user jobs after the supervisor load completes. Go step 15.

To stop the autoload process, press the message wait key before the countdown period (XX) shown in the message reaches 0 (times out). You can now perform a manual supervisor load as described in step 11.

b. If you did not automate supervisor loading or elected to perform a manual IPL, the status line of the IPL screen is blank. The system takes no action. You must manually load your supervisor. Go to step 11.

11. To manually load your supervisor, you must enter its name, the load option, and the address of the load device where the supervisor resides. You have the option of accepting the default values displayed above the command input line on the screen or entering your own values. A description of the screen values follows:

```
supervisor-name,load-option,load-device-address
```

where:

`supervisor-name`
Specifies the name of the supervisor you want to load. When you create a supervisor, you usually assign it a name. If you elected not to name your supervisor, the system assigns it the default name SY$STD.

`load-option`

`S`
Specifies the STANDARD, default load option. This option is use primarily for the initial IPL after receiving a new release or after applying an SMC or SMP. When specified, it updates the table of symbiont sizes that is maintained on disk. The update process is time-consuming and is only necessary when a symbiont size has changed.

`D`
Specifies the supervisor DEBUG load option. (For details, see the *Supervisor Technical Overview*, UP-8831.)

`L`
Specifies the special supervisor load for stand-alone programs.

Q

Specifies the QUICK load option. This option performs IPL significantly faster than the others and should be used in most cases.

Note: *Do not use the Q option if an SMC has been applied or if a new supervisor or ICAM has been generated.*

`load-device-address`

This is the device address (did) where the supervisor resides. If you manually initiated the IPL procedure, the device address you entered on the manual frame is displayed.

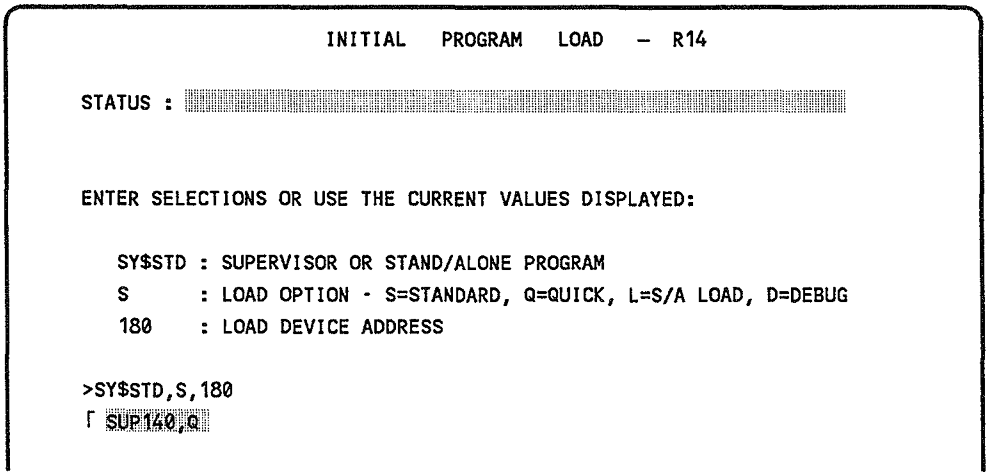
If IPL was initiated automatically, the address of the volume named as SYSRES on the configuration frame is displayed.

12. Enter the supervisor load values as follows:

    a. If you accept all the default values shown, press the XMIT key; no other entry is required. Go to step 13.

    b. If you don't accept the default values, enter the values you want.(The cursor is automatically positioned at the command input line for you to enter your values in the format shown.)

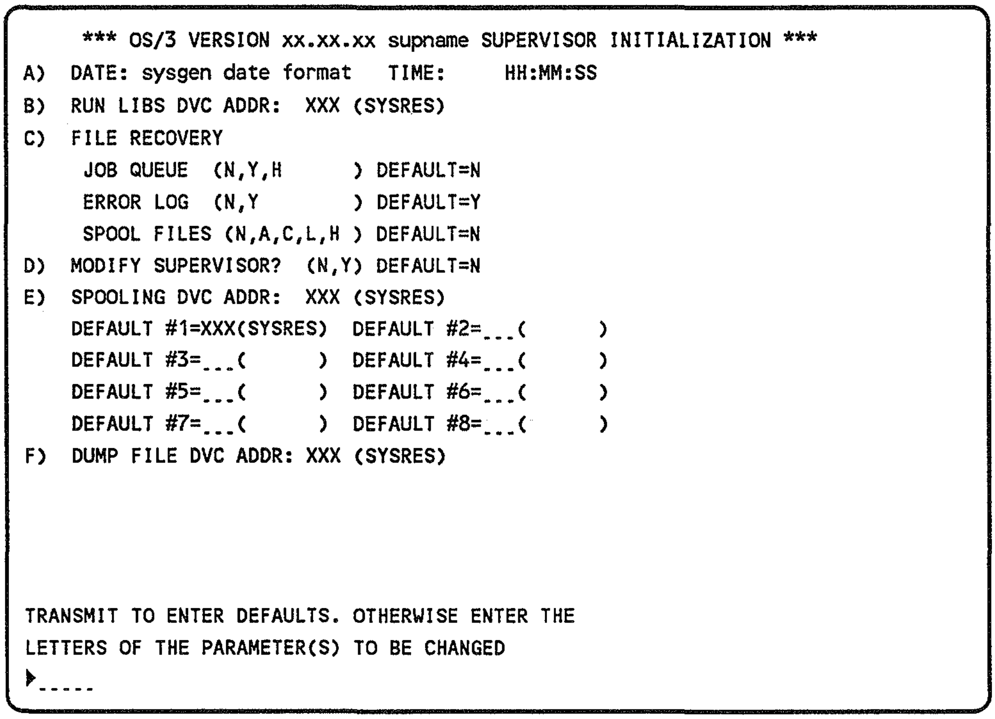
       For example, if the name of your supervisor is SUP140, it's located on device 180, and you want to use the quick load option, you complete the IPL screen as follows:



```
                    INITIAL   PROGRAM   LOAD   -  R14

  STATUS :


  ENTER SELECTIONS OR USE THE CURRENT VALUES DISPLAYED:

     SY$STD : SUPERVISOR OR STAND/ALONE PROGRAM
     S      : LOAD OPTION - S=STANDARD, Q=QUICK, L=S/A LOAD, D=DEBUG
     180    : LOAD DEVICE ADDRESS

  >SY$STD,S,180
  Γ SUP140,Q
```

    c. Press XMIT to enter your load values. Go to step 13.

13. Observe the supervisor initialization screen.



```
     *** OS/3 VERSION xx.xx.xx supname SUPERVISOR INITIALIZATION ***
A)  DATE: sysgen date format   TIME:     HH:MM:SS
B)  RUN LIBS DVC ADDR:  XXX (SYSRES)
C)  FILE RECOVERY
     JOB QUEUE  (N,Y,H      ) DEFAULT=N
     ERROR LOG  (N,Y        ) DEFAULT=Y
     SPOOL FILES (N,A,C,L,H ) DEFAULT=N
D)  MODIFY SUPERVISOR?  (N,Y) DEFAULT=N
E)  SPOOLING DVC ADDR:  XXX (SYSRES)
    DEFAULT #1=XXX(SYSRES)  DEFAULT #2=...(      )
    DEFAULT #3=...(      )  DEFAULT #4=...(      )
    DEFAULT #5=...(      )  DEFAULT #6=...(      )
    DEFAULT #7=...(      )  DEFAULT #8=...(      )
F)  DUMP FILE DVC ADDR: XXX (SYSRES)




TRANSMIT TO ENTER DEFAULTS. OTHERWISE ENTER THE
LETTERS OF THE PARAMETER(S) TO BE CHANGED
▶-----
```

This screen displays the system date and time and some of the system generation (SYSGEN) selections. From this screen, you can either accept all of the default values shown, or you can change them.

Notes:

1. *Line E is displayed only when loading a supervisor configured with spooling.*

2. *The default DVC address listed under line E are displayed only for systems with multivolume spooling files.*

a. To accept all the defaults on the screen, press XMIT. Then go to step 14.

b. To change any of the default values shown, key in the letter of the line on which the default appears in the space after the cursor. Then press the XMIT key to enter your change. (You can key in the line letters one at a time or all at once.) Go to step 14 after you have made your changes.

Read the descriptions that follow to help you understand the options listed for each line of the display.

- OS/3 VERSION Line

  Indicates the release version (*xx.xx.x*) of OS/3 and the supervisor you specified on the IPL screen. This line cannot be changed.

- Line A

  Shows the date and time supplied by the hardware clock. Usually, you'll have no need to change this information. However, if you do want to change it, key in letter A on the screen and press the XMIT key. The following message is displayed at the bottom of the screen:

  ```
  ENTER DATE, TIME YYMMDD, HHMMSS
  -- -- --, -- -- --
  ```

  Key in the date and time in the YYMMDD and HHMMSS format. Then press the XMIT key.

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

- Line B

  Shows SYSRES as the default address for the disk containing the system job run library $Y$RUN. If you want to locate $Y$RUN on a different volume, key in letter B and then press the XMIT key. The following message is displayed at the bottom of the screen:

  ```
  ENTER RUN LIBS DVC ADDR
  ---
  ```

  Enter the 3-digit device address of the disk volume to contain $Y$RUN. Remember, the volume you specify must be online for the system to be operational. If it is not, you'll receive two system messages on your console screen telling you to mount the volume.

- Line C

Shows the default values for the SYSGEN options concerning file recovery of the job queue, error log, and spool file (if you configured your system with spooling). To change any of these values, key in C on the screen and press the XMIT key. The following message is displayed at the bottom of the screen:

```
ENTER NEW VALUES FOR THE JOB QUEUE, ERROR LOG, SPOOL FILES: XXX
_ _ _
```

Enter the recovery option you want for each of the three files. When choosing these options, consider the following information concerning these three files:

- First blank space (Job Queue File)

  To delete jobs previously filed in the job queue, accept the default value of N for no recovery by not entering a value in the first blank space. To retain these jobs for processing, key in Y in the first blank space. To place them into hold status, key in H.

- Second blank space (Error Log File)

  To retain the present error log file and continue to list errors in the error log, accept the default value of Y for recovery by not entering a value in the second blank space. To clear all accumulated errors in the error log and start a new error log file, key in N for no recovery.

- Third blank space (Spool Files)

  If previously spooled input and output was processed before the system was turned off and the spool file is empty, accept the default value of N for no recovery by not entering a value in the third blank space.

  If you want to 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, key in A for all files in the third blank space. With this response, all spooled subfiles, whether complete or incomplete, are saved. This response (or H) must be specified to recover the console log file and workstation log file if configured into the system). Console and workstation messages that were not copied from the main storage buffer to the spool file are not recovered; copying is done only when the buffer has been filled.

  To recover only completed subfiles when the spool file is initialized, key in C for completed in the third blank space.

To recover only the user log directory when the spool file is initialized, key in L for log directory in the third blank space.

If you took a system dump of the previously loaded system, key in H for hot start in the third blank space. With this option, when the spool file is reinitialized, all spool subfiles (same as with A option), as well as console messages that have been accumulated in the 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 accumulated in the main storage buffer are not recovered, however.) Use this method of recovery, sometimes called a hot start, only if a system crash occurs. When there is a planned shutdown of the system and you plan to recover spool files later, breakpoint the console log, then select the A, C, or L spool recovery option when you reinitialize your system.

Press the XMIT key to enter your options.

- Line D

  Asks whether you want to modify the supervisor in any way that you specified at SYSGEN. If you do, key in D on the screen and press the XMIT key. The following message appears at the bottom of the screen:

  ```
  ENTER MODIFY SUPERVISOR OPTION (Y/N)

  _ _ _
  ```

  Your response to this query tells the system whether you want to modify your supervisor.

  If you do, key in Y for yes and press the XMIT key. The screen clears and your system displays output messages. Proceed with the appropriate operator action described in Appendix A. Then come back to this procedure.

  If not, key in N for no and press the XMIT key.

- Line E (Displayed when your supervisor is configured with spooling)

  Shows your SYSRES (or the volume set at SYSGEN) as the volume containing your spooling volume. If you want to specify another device as your spooling volume, or if you have multivolume spooling and want to specify more than one disk volume as your spool volume, enter E on the screen. Your system then displays one of two messages at the bottom of the screen depending on whether you have single-volume or multivolume spooling.

Single-Volume Spooling

```
ENTER DUMP FILE DVC ADDR

---
```

Key in the device address of the single volume other than SYSRES that you want to contain your spooling volume and press the XMIT key.

Multivolume Spooling

```
ENTER DUMP FILE DVC ADDR

---
```

Key in the device addresses of any of eight spooling devices in your system and then press the XMIT key. Remember that all volumes identified as spool volumes at IPL must be online for your system to be operational.

- Line F

  Shows SYSRES as the default address for the disk containing $Y$DUMP. If you want to specify an alternate dump file on another disk, key in the letter F on the screen and then press the XMIT key. The following message is displayed at the bottom of the screen:

  ```
  ENTER DUMP FILE DVC ADDR

  ---
  ```

  Key in the device address of the disk to contain $Y$DUMP and then press the XMIT key. During system initialization, $Y$DUMP will be allocated on this disk when necessary. However, SYSRES will always contain a minimum $Y$DUMP file for dumps taken during system initialization.

Note: *When you are using an alternate dump file and $Y$DUMP is extended on SYSRES, you can use a system job stream, SG$SADMP, to scratch the original $Y$DUMP and reallocate and format a minimum $Y$DUMP on SYSRES. The device containing the $Y$DUMP file does not need to be allocated for any user.*

14. Specify disk cache

    When your system displays the following message:

    ```
    CM01 ENTER THE NUMBER OF 1024 BYTE BLOCKS OF MEMORY FOR DISK
    CACHE OR NONE. VALID VALUES IN THE RANGE OF 160-8192?
    ```

Enter one of the following responses:

- NONE to disable disk cache operation.
- A 4-digit value, in the range of 160 to 8,192, specifying the number of 1,024-byte blocks to be assigned.

Note: *See Section 8 for more information on the disk cache facility.*

15. Observe the system ready status.

The system indicates that it has completed the initialization process and is now ready to process user jobs by displaying the following message, along with job queue and error log messages when appropriate. (If you have configured an alternate console, the message appears on the designated device.)

```
SYSTEM 80 OS/3 version-no supnam yy/mm/dd  hh:mm:ss
```

where:

`version-no`
Specifies the OS/3 software version loaded in the system

`supnam`
Specifies the name of the supervisor loaded into the system

`yy/mm/dd`
Specifies the year, month, and day used by the system and normally tracked by the hardware clock.

`hh:mm:ss`
Specifies the time in hours, minutes, and seconds used by the system.

Other messages that may be displayed are as follows:

```
JOB QUEUE RECOVERED- n JOBS QUEUED
```

This message is displayed when your supervisor is configured to retain or hold jobs previously filed in the job queue. It specifies the number of jobs in the scheduling queue.

```
ERROR LOG NOT RECOVERED
```

This message is displayed when your supervisor is not configured to retain the current error log. A new error log file is started.

2.3. Automating the Supervisor Load Process

Automating the supervisor load process requires no optional equipment or software. The function already exist in your system but must be activated to make it operational. The procedure is as follows:

1. Power on and initialize your system as described in section 2.2 if it is not already up and running.

2. Display the interactive services logon screen.

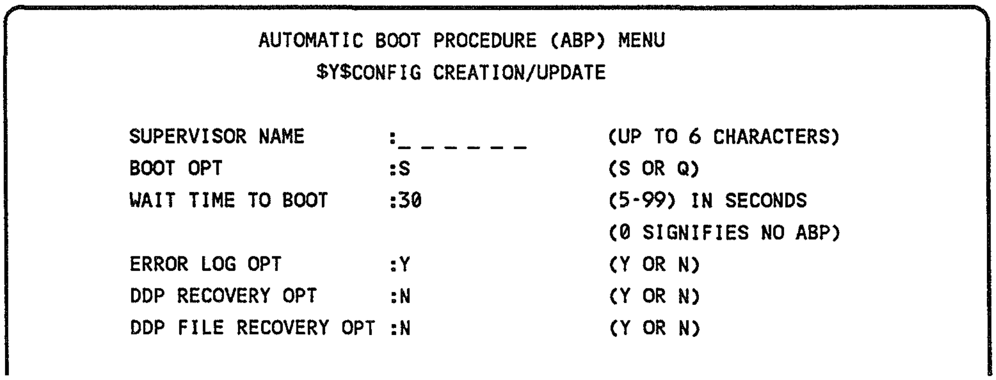
3. Enter the logon information requested by the screen.

4. Key in the command **CFIG** and then press the **XMIT** key to display the ABP (automatic boot procedure) menu for creating/updating the $Y$CONFIG file.

   Notes:

   1. *The system will not execute the CFIG command if security is present on your system. Your system administrator must enter his password-id before CFIG will execute.*

   2. *You must also provide a password to gain access to $Y$CONFIG if the file exists and is cataloged. The password must conform to the one used to catalog the file. If the file is not cataloged, pressing the XMIT key will display the ABP menu.*



```
                    AUTOMATIC BOOT PROCEDURE (ABP) MENU
                         $Y$CONFIG CREATION/UPDATE

          SUPERVISOR NAME      :_ _ _ _ _ _        (UP TO 6 CHARACTERS)
          BOOT OPT             :S                  (S OR Q)
          WAIT TIME TO BOOT    :30                 (5-99) IN SECONDS
                                                   (0 SIGNIFIES NO ABP)
          ERROR LOG OPT        :Y                  (Y OR N)
          DDP RECOVERY OPT     :N                  (Y OR N)
          DDP FILE RECOVERY OPT :N                 (Y OR N)
```

5. Change any default (or current) value in the ABP menu that is not acceptable then press the **XMIT** key to enter your input. Be aware that setting the WAIT TIME TO BOOT value to 0 seconds turns off the ABP function.

6. Log off interactive services.

The automatic supervisor load criteria is now setup. The supervisor loads automatically each time an auto-IPL occurs.

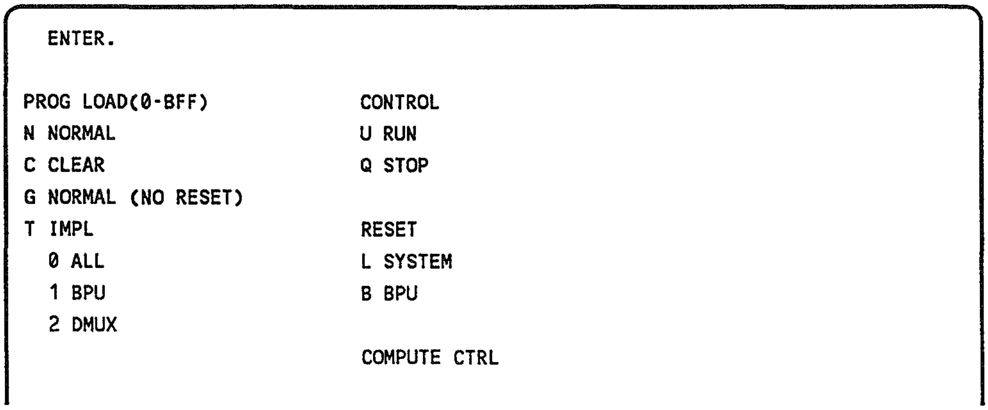
2.4. System Reinitialization

Use the following procedures to reinitialize the system after the occurrence of an unrecoverable error or when you want to load an alternate supervisor.

2.4.1. Initial Microprogram Load Procedure

Proceed as follows to manually initiate the IMPL procedure:

1. Press the ESC key, then the M key to display the manual frame.

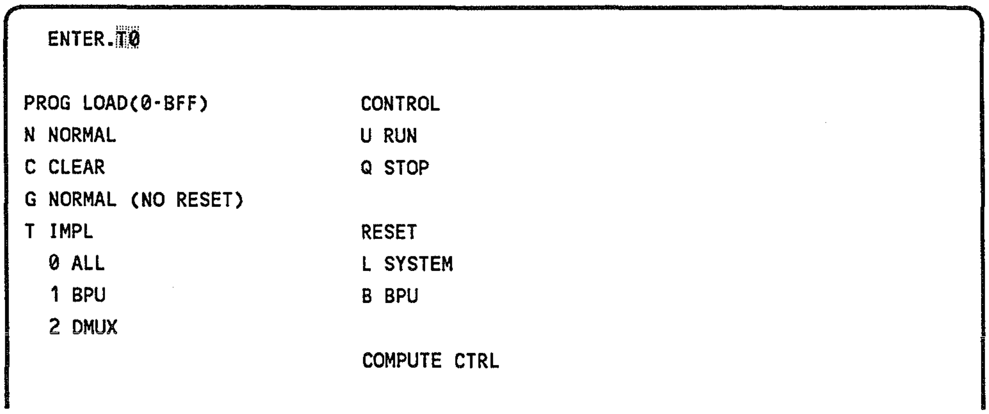


```
  ENTER.

PROG LOAD(0-BFF)             CONTROL
N NORMAL                     U RUN
C CLEAR                      Q STOP
G NORMAL (NO RESET)
T IMPL                       RESET
  0 ALL                      L SYSTEM
  1 BPU                      B BPU
  2 DMUX
                             COMPUTE CTRL
```

2. Enter the T option, followed by how much of the microcode you want to reload.

   You have the option of loading all the microcode (0), just the microcode for the BPU (1), or the microcode for the input/output system (2).

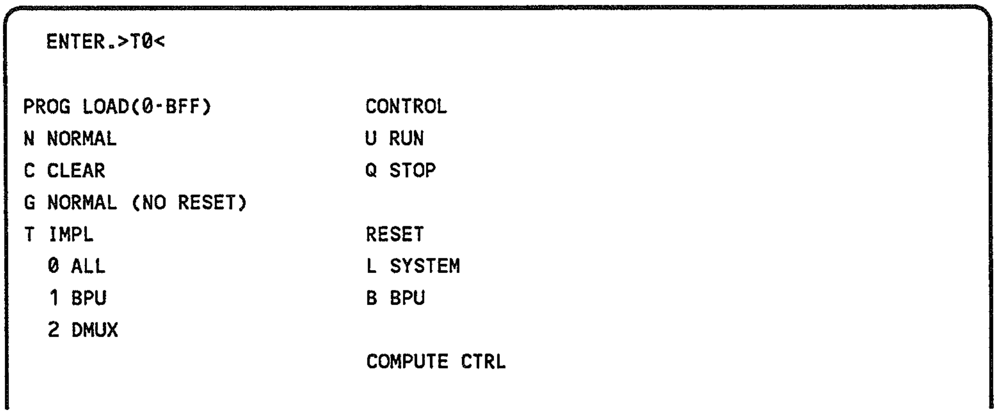
   For example, to load all the microcode, complete the manual frame as follows:



```
  ENTER.T0

PROG LOAD(0-BFF)             CONTROL
N NORMAL                     U RUN
C CLEAR                      Q STOP
G NORMAL (NO RESET)
T IMPL                       RESET
  0 ALL                      L SYSTEM
  1 BPU                      B BPU
  2 DMUX
                             COMPUTE CTRL
```

3. Press XMIT to enter your input.

   While the IMPL procedure is running, the SEQ1, SEQ2, and SEQ3 indicators light briefly. When the procedure completes, the system redisplays the manual frame. It displays your previous IMPL selection on line 2:



```
  ENTER.>T0<

PROG LOAD(0-BFF)             CONTROL
N NORMAL                     U RUN
C CLEAR                      Q STOP
G NORMAL (NO RESET)
T IMPL                       RESET
  0 ALL                      L SYSTEM
  1 BPU                      B BPU
  2 DMUX
                             COMPUTE CTRL
```

4. Go to 2.4.2 to perform the initial program load (IPL).

2.4.2. Initial Program Load Procedure

System IPL consists of two phases: the initiation and supervisor loading. The procedure for re-IPL is the same as that described in section 2.2, steps 9 through 15.

Keep in mind two thoughts:

- Because the disk control units for the models 8-20 need loadable microcode, you must IPL from the same control unit that contains the SYSRES devices.

- Make sure no user jobs are active. The system must be idle before you initiate the IPL procedure.

Go to section 2.2, step 9 to re-IPL your system.

2.5. IPL Information

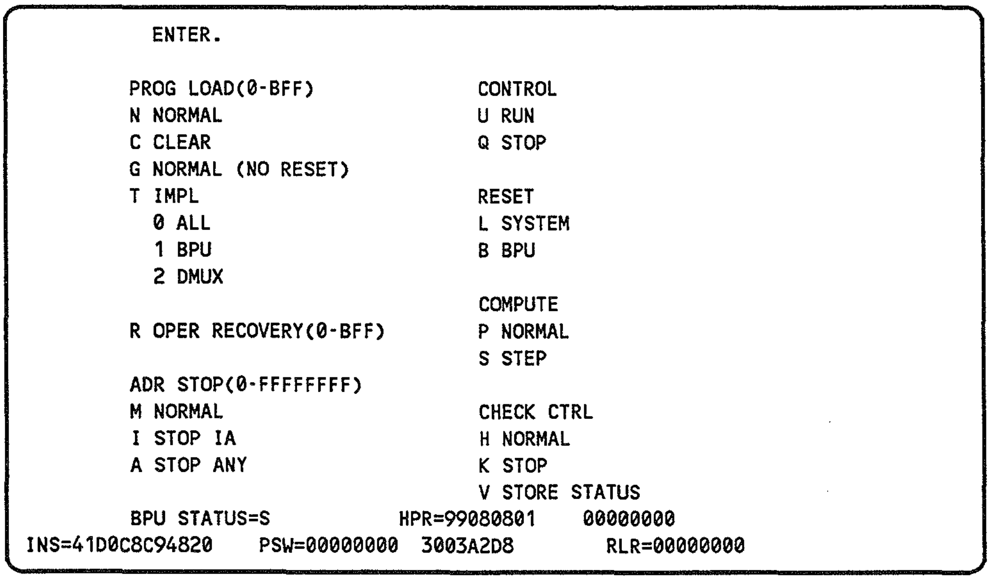
If the IPL process is not successful, IPL error message may be displayed on the console monitor, or the STOP or CHECK STOP indicators on the control panel light.

Look up the IPL error message in the *System Messages Reference Manual*, 7004 5190 for its meaning and the corrective action to be taken.

If the STOP or CHECK STOP indicators on the control panel light, your system has stopped and the load procedure has halted. In this case, you must display the manual frame to determine why the machine stopped. Do this by pressing the ESC key, releasing it, then pressing M for the manual frame.

The console screen clears at this point, then displays the following manual frame. On the second line from the bottom of the screen, the status message (BPU STATUS=S) appears. Directly under this line, the manual frame displays INS and the number of the instruction that caused your machine to stop and the program status word (PSW) identification. Check the instruction number in the *System Messages Reference Manual*, 7004 5190, for an explanation of why your machine stopped and the corrective action you should take.

The following sample frame display shows the BPU STATUS as stopped (S) and shows that instruction 41D0C8C94820 caused the IPL procedure to halt and the system to stop.



```
          ENTER.

        PROG LOAD(0-BFF)              CONTROL
        N NORMAL                      U RUN
        C CLEAR                       Q STOP
        G NORMAL (NO RESET)
        T IMPL                        RESET
          0 ALL                       L SYSTEM
          1 BPU                       B BPU
          2 DMUX
                                      COMPUTE
        R OPER RECOVERY(0-BFF)        P NORMAL
                                      S STEP
        ADR STOP(0-FFFFFFFF)
        M NORMAL                      CHECK CTRL
        I STOP IA                     H NORMAL
        A STOP ANY                    K STOP
                                      V STORE STATUS
        BPU STATUS=S          HPR=99080801     00000000
INS=41D0C8C94820    PSW=00000000  3003A2D8       RLR=00000000
```

If you configured your supervisor with the transient work area feature, your system should display the message TRANSIENT WORK AREA IS INITIALIZED as soon as the IPL procedure completes. If it doesn't, initiate the transient work area feature yourself at this time. To do so, issue the TW command (see 4.6.16).

Disk and tape devices can be identified as unavailable to the operating system and user applications at system generation (SYSGEN) time. This allows creation of supervisors that, by default, prevent access to specific devices. However, when loading a supervisor from a device generated as "down", the IPL software will automatically set that device to the "up" state. (Loadable controllers supporting devices that are generated down are also loaded during IPL.)

If a device generated as down is selected as a RUN, SPOOL, or DUMP device from the system initialization screen, a warning message is displayed indicating that the device is generated as down. You have the option of changing your device selection or retransmitting the screen unaltered to indicate that you want the IPL software to set the device up and proceed with system initialization. The IPL software does the same thing for devices generated as read-only at SYSGEN time.

During the IPL operation, the system automatically sets all devices or subsystems, except workstations, not online (not turned on) to not available. The devices or subsystems are not available for system use until you identify them as available via the SET IO command for a disk drive, until you mount a disk pack and initialize the drive. You can review the availability status for devices by using the MIX command. Jobs requiring more than the available devices are terminated with an R277 message.

Section 3
System Power-Off Procedure

3.1. System Power Off

The power-off procedure terminates system operations in an orderly manner and then removes operating power supplied to your system. The procedure does not include specific instructions for removing operating power to those peripheral subsystems that power on and off separately from your system. Instructions for powering off peripheral devices are contained in their respective operating guides.

Proceed as follows to power off your system:

1. Enter the SHUTDOWN command to terminate all system activity in an orderly manner.

Caution

When performing step 2, make certain that the data files stored on your disk subsystems are protected before powering off the system. Otherwise, a portion of the recorded data may be erased.

2. Power off all peripheral devices (those that power on and off separately from your system) as described in their respective operating guides.

3. Set the system POWER ON/OFF switch to the OFF position to remove dc operating power from the console, integrated disks, and diskettes subsystems. (The console screen clears and the POWER light on the console keyboard and all lights on the control panel go off.)

4. Be sure to leave the POWER ON/OFF switch on the console display in the ON (depressed) position.

Note: *Step 5 need not be performed unless you want to completely remove all power from the system due to an emergency or if the step is performed as a normal site requirement.*

5. Set the circuit breakers at the rear of the processor, the I/O cabinet, and the I/O expansion cabinet to the OFF position.

Section 4
Job Processing Procedures

After you successfully initialize your system, 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 console workstation or console to initiate running the job, using the system commands described in this section. When card input is used, press the RUN switch on the card reader after you enter the command and the I/O stop state message is displayed.

Notes:

1. *To demonstrate that all installed products in the system are operational, refer to the* Installation Verification Procedures Operating Guide *(7004 5232).*

2. *See the About This Document section for the conventions used to illustrate the commands and messages appearing in this guide.*

4.1. Keyin Procedure

4.1.1. Console Keyin Procedure

Before keying in any message or command on the console for transmittal to the system, you must press the MSG WAIT key on the console. Pressing this key:

- Notifies the system that you want to send a message or command to it

- Automatically opens a line on the console screen for display of the input command or message

- Gives the start of entry (▶) signal

- Sets the cursor (▨) to the position where typing is to start

After keying in the message or command, press the XMIT key on the console keyboard. This initiates transmittal of the message or command to the system. The accepted command (or unsolicited message used instead of a command, as we describe in 4.3.3) is either acted upon immediately or placed on a queue for future processing; it is 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 Reference Manual*, 7004 5190.

If console logging is configured in your system, the console log messages are recorded in a spool file for later printing.

4.1.2. Alternate Console Keyin Procedure

You can designate that a local workstation serve as the alternate console by system generation or the SET command. The local workstation takes the place of the hardware console and becomes the command entry and system control device.

Before keying in any message or command on the alternate console for transmittal to the system, you must press MSG WAIT on the keyboard. Pressing this key:

- Notifies the system that you want to send a message or command to it
- Automatically opens a line on the console screen for display of the input command message
- Positions the cursor at the beginning of the input line to allow you to type in your command

After keying in the command or message, press the XMIT key to enter it into the system. The system will exit the system line and repeat your input at the bottom of the screen. Input received is handled the same way as input from the regular console (see 4.1.1).

The spacebar is set as destructive so you need not clear the input line manually each time you use it. Just type in the new text on top of the previous command. The last command entered remains on the system command line so that you may edit and reenter it without having to rekey the entire command.

Although the device is physically a workstation, when it is functioning as the alternate console it cannot run programs or functions that allocate the entire screen (such as the editor or screen format services).

If the alternate console becomes unusable, the system will automatically reactivate the original console. Normally when the alternate console is active, input from the original console is rejected with a negative acknowledgement (NAK) error message.

If you specify a workstation at system generation to be the alternate console, the hardware console is still used to IPL the system. The system switches control to the alternate console when the system is successfully initialized. The hardware console screen is cleared and has the header line:

```
OS/3 INACTIVE CONSOLE - ALTERNATE CONSOLE IS DEVICE ddd
```

You can switch the alternate console to another device or back to the hardware console by using the command:

```
        ⎡did⎤
SET IO, ⎨ORG⎬,CON
        ⎣ALT⎦
```

where:

`did`
Is the device address.

`ORG`
Is the original hardware console.

`ALT`
Is the system generated.

`CON`
Is a console change.

4.2. 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:

```
▶command ⎡⎧(did)         ⎫⎤ Δ[command-parameters]
         ⎢⎨([did],label) ⎬⎥
         ⎣⎩(RDR,label)   ⎭⎦
```

where:

▶

Is the start-of-entry (SOE) symbol that must precede all lines. This symbol is automatically generated by OS/3 when the MSG WAIT 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 three-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, it must be formatted in data set label mode; a file identifier (label) is required to identify the specific file to be accessed on the diskette. The device address also must be included, unless the diskette is configured as the SYSRDR. The label, which may be a maximum of eight alphanumeric characters, is separated from the device address by a comma. The data set label diskette record size must be 128 bytes or less, and the records must be unblocked and unspanned.

`(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 that the diskette with a device address of 010 contains a file labeled DATANAM1 to be accessed by the command.

```
RUN(,DATANAM1) JOBNAM1
```

Specifies that the diskette configured as SYSRDR contains a file labeled DATANAM1 to be accessed by the command.

```
RUN(RDR,DATANAM1) JOBNAM1
```

Specifies that the input spool file contains a subfile labeled DATANAM1 to be accessed by the command.

`command-parameters`

Are optional positional parameters used to tailor the effect of the command being issued. Each specified parameter must appear in its own position. Commas are used to separate positions. For example, the following portion of the RUN command format indicates that two positional parameters are associated with the command; the comma separates the first from the second.

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

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 is considered the name of the job to be run.

Also, if a parameter takes 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 either of the following ways:

```
RUN MYJOB(JOBA),PRE

RUN (TEMPNAME)
```

In either case, the parentheses must be included.

4.3. Message Characteristics

4.3.1. Output Messages

Output messages are displayed on the console workstation or console to provide you with information, to direct you to take some action, or to ask you a question that requires a response. The messages that can be output to the console workstation or console by the components of OS/3 are described in the *System Messages Reference Manual*, 7004 5190, together with their associated operator responses, when appropriate. The format of an output message is:

```
nn ⎧?⎫                ⎧jobnamez⎫
   ⎨Δ⎬ message-text  ⎨synnnn  ⎬
   ⎩*⎭                ⎩command ⎭
```

where:

nn

Is a unique message number from 11 to 99 (numbers 1 through 10 are reserved for other system uses). This number serves as the message identification. Numbers 11 to 42 are used for questions.

?

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 console workstation or console screen until they are answered.

Δ

Identifies an output message that requires no reply or operator action; it gives information only. Input messages, solicited and unsolicited, must include a space between the message-id and message text.

*

Identifies an output message that requires you take some action. 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 (see 4.5.3).

message-text

Comprises the actual message content and is a maximum of 60 characters. The message text includes the message prefix which you use to find the message in the OS/3 manual documenting it - either the *System Messages Reference Manual*, 7004 5190 or the OS/3 manual describing the system program that displays the message.

jobnamez

Is the name of the user job sending the message. This name is the 8-character name taken from the job preamble.

synnnn

Is the name of the symbiont sending the message. This name is the 2-character symbiont identification followed by a four-digit binary job number inserted at task initiation. It is a six-character name altogether.

command

Is the interactive command followed by the interactive command ID.

4.3.2. Solicited Input Messages

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

```
nnΔmessage-text
```

where:

nn

Is the unique message number of the message you are responding to.

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.*

4.3.3. Unsolicited Input Messages

Unsolicited messages are those messages you input that are not in direct response to an output message that requires a reply. You may enter unsolicited messages from the console or from the workstation that initiated the job or symbiont. The format for all unsolicited input messages is:

```
┌00Δsynnnn        ┐
│00Δsymbname,     │
│00Δsymid         │Δmessage text
│00Δsymid(did)    │
{UNSΔjobname,     }
│UNSΔsymid,S      │
│UNSΔsynnnn       │
│UNSΔsymbname,S   │
└UNSΔsymid(did),S ┘
```

where:

`00`
Specifies that the unsolicited message is for a symbiont.

`UNS`
Specifies that the unsolicited message is for a user job.

`symbname`
Is the eight-character name of the symbiont receiving the message. For example, SL$$OW00 is the symbiont name of the output writer.

`symid`
Is the two-character symbiont identification used to invoke the symbiont receiving the message. For example, PR is the symbiont identification of the output writer.

`symid(did)`
Is the two-character symbiont identification currently used to invoke the symbiont receiving the message, along with the address of the device used or controlled by a specific copy of the symbiont in main storage. For example, PR(160) is the symbiont identification of the output writer using the printer that is at device address 160. 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.

`jobname`
Is the name of the user job receiving the unsolicited message.

`S`
Indicates a symbiont identification.

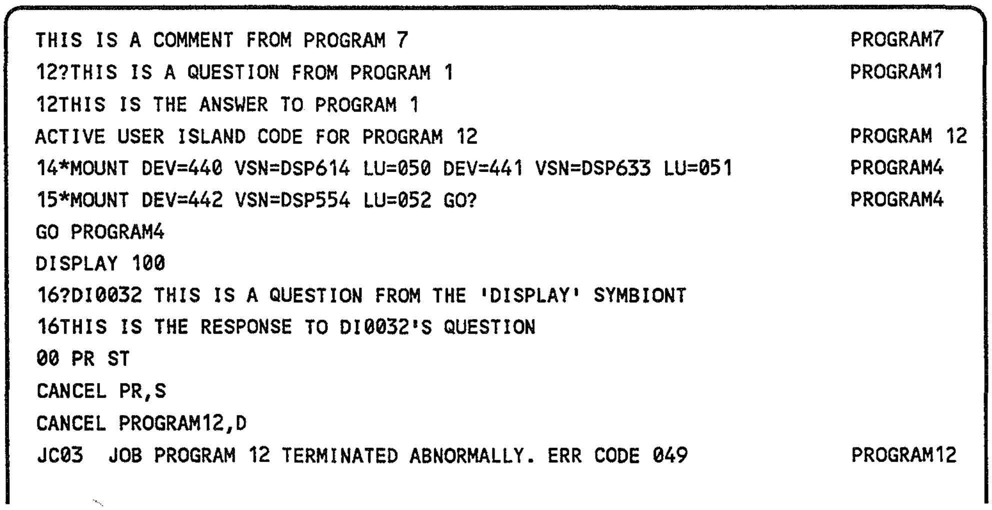
`synnnn`
Is the name of the symbiont sending the message. This six-character name is the two-character symbiont identification followed by a four-digit binary job number inserted at task initiation.

`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 active; only the first will get the message. Figure 4-1, line 11 is an example of an unsolicited keyin to a symbiont.*

Following in 4.4 and 4.5 are descriptions of 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.

Figure 4-1 shows typical messages, how OS/3 displays them, and how you respond to them. Note that the messages are not meant to be real; they simply show how you use the message characteristics described in this section.



```
THIS IS A COMMENT FROM PROGRAM 7                                  PROGRAM7
12?THIS IS A QUESTION FROM PROGRAM 1                              PROGRAM1
12THIS IS THE ANSWER TO PROGRAM 1
ACTIVE USER ISLAND CODE FOR PROGRAM 12                            PROGRAM 12
14*MOUNT DEV=440 VSN=DSP614 LU=050 DEV=441 VSN=DSP633 LU=051      PROGRAM4
15*MOUNT DEV=442 VSN=DSP554 LU=052 GO?                            PROGRAM4
GO PROGRAM4
DISPLAY 100
16?DI0032 THIS IS A QUESTION FROM THE 'DISPLAY' SYMBIONT
16THIS IS THE RESPONSE TO DI0032'S QUESTION
00 PR ST
CANCEL PR,S
CANCEL PROGRAM12,D
JC03  JOB PROGRAM 12 TERMINATED ABNORMALLY. ERR CODE 049          PROGRAM12
```

Figure 4-1. Typical Messages

4.4. Job Processing During the Daily Memory Refresh Routine

Model 10, 15, and 20 systems automatically run a memory refresh routine every day at midnight. When initiated, this routine accesses four megabytes of main storage and causes a hardware correction mechanism to correct any single-bit errors that are encountered. This operation increases overall system performance since it reduces the possibility that uncorrectable multibit errors will occur. Multibit errors require system recovery operations and may cause halts.

The memory refresh routine automatically starts at midnight and runs for approximately three minutes. During this time, you may notice a slight decrease in job processing performance.

Note: Refer to 4.3.1. for output message format definition.

When memory refresh is initiated, the following message is displayed at the system console:

```
nn MEMORY REFRESH INITIATED                             synnnn
```

When memory refresh is completed, the following message is displayed at the system console:

```
nn MEMORY REFRESH OF xxxxxxx-yyyyyyy COMPLETE           synnnn
```

where:

xxxxxxx
Is the starting main storage address.

yyyyyyy
Is the ending main storage address.

4.5. 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)
- Stop jobs under execution (terminate jobs)

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

1. PRE (preemptive) Queue

   Contains jobs to be executed first (i.e., before any jobs assigned HIGH, NOR, or LOW scheduling priority), even if the resources for any HIGH, NOR, or LOW priority jobs are available and the resources for any PRE jobs are not. PRE jobs are always executed first unless they're placed on hold via the HOLD command (see 4.5.2). If rollin/rollout is configured, a PRE job initiated for execution when sufficient main storage is not available may cause HIGH, NOR, or LOW 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. When rollin/rollout is configured, a maximum of 14 jobs can be in the preemptive queue.

2. HIGH Queue

   Contains jobs to be executed before any jobs assigned a NOR or LOW 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 or LOW priority jobs (unless they're placed on hold), even if the resources for the HIGH job are not available and the resources for the NOR or LOW jobs are. A maximum of 35 jobs can be in the high priority queue.

3. NOR (normal) Queue

   Contains jobs to be executed before any jobs assigned a LOW scheduling priority. NOR scheduling priority jobs are 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 (4.5.2). NOR scheduling priority jobs are always executed before the LOW priority jobs (unless they're placed on hold) even if the resources for the NOR job are not available and the resources for the LOW job are. A maximum of 70 jobs can be in the normal priority queue.

4. LOW Queue

   Contains jobs to be executed only when there are no jobs left in the PRE, HIGH, or NOR queues or when the queues are placed on hold. A maximum of 70 jobs can be in the low priority queue.

4.5.1. Job Initialization

Job control streams are read into the system by using one of the job initialization commands (FILE, RUN/RV, and SI/SC). 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. Input device (card reader, data set label diskette, or spool file)

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

3. $Y$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. You can override this specification by entering another scheduling priority in the job initialization command. The default is to use the normal priority queue.

Filing Job Control Streams (FILE)

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 can be from a card reader, a diskette drive (data set label diskette only), or the input spool file.

Note: *The FILE command cannot be issued from an enter stream.*

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).

Jobs filed from the card reader must terminate with a // FIN job control statement. When jobs are filed from diskette or the spool file, the // FIN job control statement is not necessary. Jobs input from diskette to the spool file must be single volume and formatted in data set label mode.

Format

```
FILE [ ┌(did)          ┐ ] Δ [ ┌: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. You can use a disk or a format label diskette for the alternate library file.

```
: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 or 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 or format label diskette to be used. 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 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.

```
:alt-filename,⎡⎡RES⎤⎤,write-password
              ⎢⎨RUN⎬⎥
              ⎣⎣vsn⎦⎦
```

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 or 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 or format label diskette 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.

Note: *If you specify a disk or diskette vsn for the alternate library file in the command when the volume isn't already mounted, the system displays a mount message suggesting an available device. The volume can be mounted on the available device or any other suitable device. In rare instances when the system cannot determine whether the volume is a disk or diskette, the message may suggest an available disk drive for a diskette volume. Ignore this inconsistency and mount the diskette volume on any suitable device.*

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.

Running Job Control Streams (RUN/RV)

Function

The RUN/RV commands read 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 (data set label diskettes only), or the input spool file.

The commands cause the job control stream to be written to the job run library ($Y$RUN) file and expanded to its *run* state (JPROCs are expanded), then scheduled for execution. $Y$RUN is a temporary file; the job is deleted when it is run.

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. 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; 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 by using the SI/SC commands. (SI/SC commands are described next in this subsection.)

The RV command initiates the reading of a prefiled job 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 RUN command initiates the reading of a job control stream that requires 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 an input device. You must include a job name when you enter an RV command.

Remember that, when a system card reader is placed online, the RUN command to read a job control stream from 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 console workstation or console. The RUN command can 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. Presuming that nothing is in the hopper when the second RUN command is executed, a hopper empty message results.

Format

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

   ⎡⎧:alt-filename                                       ⎫⎤
   ⎢⎪:alt-filename,⎧RES⎫                                 ⎪⎥
   ⎢⎪              ⎨RUN⎬                                 ⎪⎥
   ⎢⎨              ⎩vsn⎭                                 ⎬⎥
   ⎢⎪:alt-filename,⎡⎧RES⎫⎤,read-password)                ⎪⎥
   ⎢⎪              ⎢⎨RUN⎬⎥                               ⎪⎥
   ⎣⎩              ⎣⎩vsn⎭⎦                               ⎭⎦

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

Command Codes

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 the job is prefiled and you don't specify an input device, the first available card reader is assigned to the job, whether or not it is required. If you omit an input device and job name, the first available card reader is expected to contain the job to be run.

You must specify a label when the job control stream is on a diskette. If it is in the input spool file, you must specify RDR and a label (see 4.2). 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 and formatted in data set label mode.

`RV`

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

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. The job name is required with RV.

You include *new-name* to assign a new one- to eight-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 one- to eight-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 parameter 1 is omitted from the RUN command, the job is read and stored in a queue under the job name on the // JOB statement in the job control stream.

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 $Y$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 command which 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 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.

Parameter 3

Used to specify scheduling priority and/or time (hour and day).

`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.

`LOW`
: Places the job in the low scheduling priority queue to await execution.

`time`
: Is a four-digit number specifying a time of day in military format indicating when execution of the job is to begin.

`n`
: Is an integer from 1 to 9 specifying the number of days after today to start execution of the job.

Parameter 3 can be specified with priority only, time only, or priority and time. If 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.

Parameters 4 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 values must be supplied by the user requesting the job.

Notes:

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

2. *If you specify a disk or diskette vsn for the alternate library file in the command when the volume isn't mounted, the system displays a mount message suggesting an available device. The volume can be mounted on the available device or any other suitable device. In rare instances when the system cannot determine whether the volume is a disk or diskette, the message may suggest an available disk drive for a diskette volume. Ignore this inconsistency and mount the diskette volume on any suitable device.*

Examples

1. Operator keyin:

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

   Function requested:

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

2. Operator keyin:

   ```
   RV MYJOBA(NETPAY)
   ```

   Function requested:

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

3. Operator keyin:

   ```
   RV JOBABC:(CPYLIB,PUBDSK)
   ```

   Function requested:

   The job named JOBABC, filed in the alternate library file CPYLIB on volume serial number PUBDSK, is placed in the scheduling queue specified in the job control stream.

4. Operator keyin:

   ```
   RV MYJOB,2230
   ```

   Function requested:

   Execution of the job, MYJOB, will begin as soon as resources are available after 2230 hours (10:30 p.m.).

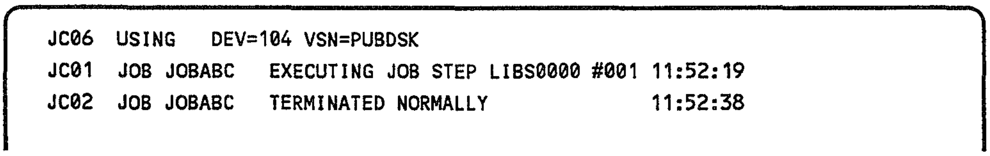
5. Operator keyin:

   ```
   RV MYJOB,P2230+1
   ```

   Function requested:

   The job, MYJOB, will be placed in the preemptive scheduling priority queue at 10:30 p.m. tomorrow.

Example:



```
JC06  USING   DEV=104 VSN=PUBDSK
JC01  JOB JOBABC   EXECUTING JOB STEP LIBS0000 #001 11:52:19
JC02  JOB JOBABC   TERMINATED NORMALLY              11:52:38
```

Explanation:

The system messages provide device information and the job step currently executing. However, the console does not display job slots, so you do not see your job occupying a job slot on the screen.

Running Saved Job Control Streams (SI/SC)

The SI/SC commands initiate the running of a job control stream from the $Y$SAVE MIRAM library file or from an alternate library file, then schedule the job for execution. In either case, the control streams have been saved in their expanded *run* state. A control stream is expanded in the $Y$RUN file when the RUN/RV 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 the $Y$SAVE or 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, data set label 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

```
┌   ┌┌(did          ┐┐┐
│SI ││([did],label) }│├Δjobname[(new-name)]
│   │└(RDR,label)   ┘││
{   └               ┘│
│                    │
└SC                  ┘
          ┌┌:alt-filename                               ┐┐┌ ┌PRE ┐┐[time + n]
          ││:alt-filename,┌RES┐                         ││├,│HIGH├┤
          ││              {RUN}                         ││││NOR │││
          ││              └vsn┘                         ││└ └LOW ┘┘
          {│                                            ├│
          ││:alt-filename, ┌┌RES┐┐,read-password        ││
          ││               │{RUN}│                      ││
          │└               └└vsn┘┘                      ┘│
          └                                              ┘
```

Command Codes

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.) 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.

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 name consists of one to eight alphanumeric characters.

You include *new-name* to assign a new one- to eight-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.

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.

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

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.

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

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.

Parameter 3

Used to specify scheduling priority and/or time (hour and day).

`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.

`LOW`
Places the job in the low scheduling priority queue to await execution.

`time`
Is a four-digit number specifying a time of day in military format indicating when execution of the job is to begin.

`n`
Is an integer from 1 to 9 specifying the number of days after today to start execution of the job.

Parameter 3 can be specified with priority only, time only, or priority and time. If parameter 3 is omitted, the scheduling priority assigned to the job via the job control stream is in effect.

Example

Operator keyin:

```
SI MYJOB(AVGYTD),H2230+1
```

Function requested:

The job called MYJOB, expanded and filed in $Y$SAVE, is 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 after 10:30 p.m. tomorrow.

4.5.2. 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) to execute that job. While waiting for these resources, you 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. You may also control, in a similar manner, 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 scheduling priority

Deferring Jobs Scheduled for Execution (HOLD)

The HOLD command permits you 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

```
      ┌     ┌          ┐ ┌ ┌        ┐ ┌          ┐                  ┐┐
HOLDΔ │ JBQ │ , ┌PRE  ┐│ │ │ , ┌OLD┐│ │ , ┌DDP  ┐│                  ││
      │     │   │HIGH ││ │ │   └NEW┘│ │   │LOCAL││                  ││
      │     │  {│NOR  }│{│ └        ┘ │  {│RBP  }│                  }│
      {     │   │LOW  ││ │            │   └WKSTN┘│                  │}
      │     │   └ALL  ┘│ │            └          ┘                  ││
      │     └          ┘ └[,UID=user-id][,HOST=host-id]             ┘│
      └jobname                                                       ┘
```

Positional Parameter 1

JBQ

Specifies the command applies to the jobs in a job scheduling priority queue, as further defined by PRE, HIGH, NOR, LOW 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.

LOW

Defers the jobs in the low 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 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.

Note: *When issued from a workstation, the HOLD console command will hold all OLD jobs on the job queue that were initiated by the same workstation, but will not hold NEW jobs entering the job queue. There is no mechanism at a workstation to hold NEW jobs.*

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 either initiated from a workstation or using the // OPTION ORI= or // OPTION MAS= job control statement.

Note: *All of the preceding 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 one- to six-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 other 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.

Scheduling Deferred Jobs (BEGIN)

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

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

Positional Parameter 1

JBQ

Specifies that the command applies to the jobs in a job scheduling priority queue, as further defined by PRE, HIGH, NOR, LOW 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.

LOW

Permits the jobs in the low 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 to be 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 the HOLD command earlier in this subsection.)

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 using the // OPTION ORI= or // OPTION MAS= job control statement to be scheduled for execution.

Note: *All of the preceding keyword parameters may be interchanged.*

UID=user-id

Permits all jobs associated with a particular workstation user-id to be scheduled in the priority queue defined in parameter 1. The user-id is a one- to six-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 other 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 the host ABC may be scheduled for execution.

The BE command produces the same screen display as the RN/RV command (see 4.5.1).

Deleting Jobs or Symbionts from Scheduling Priority Queues (DELETE)

The DELETE command permits you to delete jobs or symbionts 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.

You can delete all queued symbionts or specific symbionts by queue slot ID number. These ID numbers are provided in the MI SQ command display.

Note: *The DELETE command cannot be issued from an enter stream.*

Format

```
DELETE∆┌JBQ,┌PRE ┐┌┌  ┌DDP  ┐┐                          ┐┐[,LOG]
       │    │HIGH││ ,│LOCAL│                            │
       │   {NOR }{  {RBP  }                            }
       │    │LOW ││  │WKSTN│                            │
      {     └ALL ┘│└  └     ┘┘                           │
       │          └[,UID=user-id][,HOST=host-id]       ┘
       │SQ,┌id[id,...,id]┐                              }
       │   {ALL          }                             │
       └jobname                                        ┘
```

Positional Parameter 1

JBQ,

Specifies the command applies to the jobs in a job scheduling priority queue, as further defined by PRE, HIGH, NOR, LOW, 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.

LOW

Deletes the jobs in the low scheduling priority queue.

ALL

Deletes the jobs in all scheduling priority queues. When UID=*user-id* is specified in the command, ALL is the default condition.

SQ,

Specifies the command applies to symbionts in the outstanding symbiont queue, as further defined by id or ALL, where:

id

Is the queue slot ID number for the symbiont to be deleted. You obtain this ID number by executing the MI SQ command.

ALL

Deletes all symbionts from the outstanding symbiont queue. The MI SQ command lists all symbionts in the queue.

jobname

Specifies that a particular job is to be deleted 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 using the // OPTION ORI= or // OPTION MAS= job control statement.

Note: *All of the preceding 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 one- to six-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 that 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 residing in the high scheduling priority queue and originating from host CDE 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 deleted and their logs are printed. A system message tells you the number of jobs deleted.

Displaying Jobs in Scheduling Priority Queues (DISPLAY)

The DISPLAY command permits you to display the contents of any or all job scheduling queues on the console workstation or 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, then NOR, then LOW. 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
- Whether a hold local (HL), hold RBP (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 the command is from the system console (local or remote) with no other parameters. 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

```
                          ┌                                  ┐
                          │ ┌          ┐                     │
DISPLAYΔJBQ,┌PRE ┐        │ │ ,┌DDP  ┐ │                     │
            │HIGH│        │ │  │LOCAL│ │                     │
           {│NOR │}      {  │  │RBP  │ │                      }
            │LOW │        │ │  └WKSTN┘ │                     │
            └ALL ┘        │ └          ┘                     │
                          └[,UID=user-id][,HOST=host-id]     ┘
```

Positional Parameter 1

JBQ

Specifies the command applies to the jobs in a job scheduling priority queue, as further defined by PRE, HIGH, NOR, LOW, 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.

LOW
Displays the jobs in the low 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 using the // OPTION ORI= or // OPTION MAS= job control statement.

Note: *All of the preceding 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 one- to six-character alphanumeric identification that does not start with Y.

HOST=host-id
Displays all 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

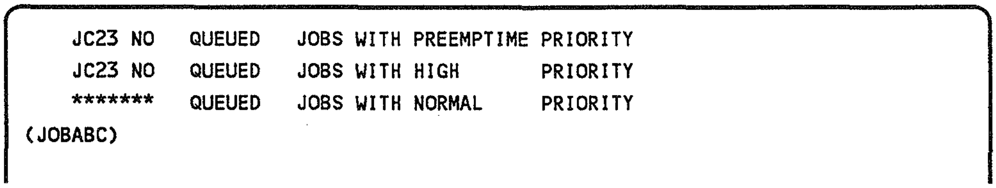
1. Operator keyin:

   ```
   DI JBQ
   ```

   Function requested:

   Display all jobs in all scheduling queues.

   Screen display:



```
    JC23 NO    QUEUED    JOBS WITH PREEMPTIME PRIORITY
    JC23 NO    QUEUED    JOBS WITH HIGH       PRIORITY
    *******    QUEUED    JOBS WITH NORMAL     PRIORITY
(JOBABC)
```

   Explanation:

   The job JOBABC is the only job presently in the system. It is in the normal priority queue and the parentheses indicate that the job is being held.

2. Operator keyin:

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

   Function requested:

   Display all locally and remotely entered jobs in the normal scheduling priority queue.

3. Operator keyin:

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

   Function requested:

   Display all jobs in all scheduling priority queues that are associated with the workstation user WKSTA2.

Changing a Job Scheduling Priority (CHANGE)

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

```
CHANGEΔjobname,⎡PRE ⎤
               ⎨HIGH⎬
               ⎢NOR ⎥
               ⎣LOW ⎦
```

Positional Parameter 1

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

Positional Parameter 2

`PRE`
Moves the specified job into the PRE scheduling priority queue.

`HIGH`
Moves the specified job into the HIGH scheduling priority queue.

`NOR`
Moves the specified job into the NOR scheduling priority queue.

`LOW`
Moves the specified job into the LOW scheduling priority queue.

Example

Operator keyin:

```
CH JOBABC,H
```

Function requested:

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 scheduling priority queue. The following message appears on your screen:

```
CH OF JOB JOBABC
```

4.5.3. Job Execution

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

- Suspend a job under execution
- Restart a job that has been suspended
- Raise or lower the switching priority level of a job being executed

Suspending a Job in Progress (PAUSE)

The PAUSE command suspends processing of a job. You may enter the command at any time, at a workstation or at the system console, and job processing suspends immediately or at the completion of a specified job step. If the job is between job steps, PAUSE (without a job-step-number parameter) takes effect at the beginning of the next job step. The PAUSE command permits you to mount a new volume on a tape, disk, or diskette drive, replace paper on the printer, or place more cards in the card reader. The suspended job is reactivated by the GO command. Furthermore, a suspended job cannot be cancelled until it is reactivated with the GO command.

Note: *A job paused by the console operator* cannot *be reactivated with a GO command from the workstation user. It can only be reactivated by the console operator.*

Format

```
PAUSEΔjobname[,job-step-number]
```

Positional Parameter 1

`jobname`
Specifies the name of the job whose processing is suspended.

Positional Parameter 2

`job-step-number`
Specifies the number of a job step that must complete before processing is suspended. This is an optional entry.

Examples

1. Operator keyin:

```
PA JOBA
```

Function requested:

Suspends JOBA processing immediately.

2. Operator keyin:

```
PA JOBA,02
```

Function requested:

Suspends JOBA following completion of the second job step.

Reactivating a Suspended Job (GO)

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, disk, or diskette drive. The GO command also is required as a response to a message from the system preceded by an asterisk (*).

Jobs started at a workstation and paused at the system console can be restarted only with a GO command at the system console. A GO command at a workstation on a job that was paused at the system console results in an NAK (negative acknowledgment) error message.

Format

```
GOΔjobname[,nn]
```

Positional Parameter

`jobname`
Specifies the job to be reactivated after execution was temporarily suspended.

`nn`
Is the number of the message requiring the GO command. This specification allows multitask jobs to suspend and start tasks other than the primary ones.

Example

Operator keyin:

```
GO JOBA
```

Function requested:

Reactivate job processing for the suspended job named JOBA.

Changing a Job Switching Priority (SWITCH)

The SWITCH command changes the switching priority level for a job under execution. The switching priority level is changed for either the currently executing job step or for the current job step and all subsequent job steps. If the priority level 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 higher 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.

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 priority levels that the highest or lowest priority task can be moved within the switching priority limits.

Format

```
SWITCHΔjobname,⎡+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 is in effect for all subsequent job steps. If omitted, the priority is only for the current job step.

Example

Operator keyin:

```
SW PAYJOB,-5
```

Function requested:

Lowers the switching priority level of the currently executing PAYJOB job step by five queues. The switching priority remains changed for the duration of the job step.

4.5.4. Job Termination

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

Canceling a Job in Progress (CANCEL)

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

Format

```
CANCEL△⎡jobname    ⎤ ⎡,⎧D⎫⎤
       ⎢symbiont,S ⎥ ⎢ ⎩N⎭⎥
       ⎨synnnn,S   ⎬ ⎣    ⎦
       ⎢sy(did),S  ⎥
       ⎣sy         ⎦
```

Positional Parameter 1

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

`synnnn`
: Is the name of the symbiont sending the message. This six-character name is the two-character symbiont identification followed by a four-digit binary job number inserted at task initiation.

`symbiont`
: Specifies the eight-character symbiont name.

`sy`
Specifies the two-character ID used to call the symbiont to be terminated.

`did`
Is the device address.

Positional Parameter 2

`D`
Specifies that a dump is to be 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 specified 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 cancelled.

Examples

1. Operator keyin:

```
CA JOBA,D
```

Function requested:

Terminates JOBA immediately, deletes the JOBA run library file, and provides a JOBA dump.

2. Operator keyin:

```
CA SU,S,N
```

Function requested:

Terminates the system utility symbiont immediately without a dump.

Soft Cancel of a User's Job (CJ)

The CJ command delays the standard cancel procedure until the job is executing under "normal" job conditions. This allows the operator to cancel a job without compromising a MIRAM file. Normal job execution requires:

- Execution under the job's key
- No wait flags set
- Not executing in external code

If this command does not terminate a job, the CANCEL command can be used. When CJ is issued, an unknown period of time may pass before the job actually cancels (for instance, the job may have been paused or yielded when the command was executed). A dump requested at the time of the command may not reflect the status of the command at that time; instead, it will reflect the job status when the cancel becomes effective.

Format

```
CJΔ⎧jobname      ⎫ ⎡ ,⎧D⎫ ⎤
   ⎪symbiont,S   ⎪ ⎣  ⎩N⎭ ⎦
   ⎨synnnn,S     ⎬
   ⎪sy(did),S    ⎪
   ⎩sy           ⎭
```

Positional Parameter 1

jobname
: Specifies the name of the job that will be terminated the next time it is executing under normal job conditions.

symbiont
: Specifies the eight-character symbiont name.

synnnn
: Is the name of the symbiont sending the message. This six-character name is the two-character symbiont identification followed by a four-digit binary job number inserted at task initiation.

sy
: Specifies the two-character ID used to call the symbiont to be terminated.

did
: Is the device address.

S
: Identifies the parameter entry as the name of a symbiont.

Positional Parameter 2

D
: Specifies that a dump is to be taken at the time when the job is effectively canceled, 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.

Examples

1. Operator keyin:

```
CJ JOBA,D
```

Function requested:

Terminates JOBA as soon as JOBA is executing under normal conditions, deletes the JOBA run library file, and provides a JOBA dump.

2. Operator keyin:

```
CJ JOBA,N
```

Function requested:

Terminates JOBA as soon as JOBA is executing under normal conditions, deletes the JOBA run library file, and does not provide a JOBA dump.

Stopping Execution of a Dump (END)

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

Format

```
ENDΔDUMP, ⎡jobname      ⎤
          ⎨symbiont,S   ⎬
          ⎢sy,S         ⎥
          ⎣sy(did),S    ⎦
```

Positional Parameter 1

DUMP
: Specifies that the execution of a dump is to stop.

Positional Parameter 2

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

symbiont
Specifies the eight-character symbiont name.

sy
Specifies the two-character ID used to call the symbiont to be terminated.

did
Is the device address.

Positional Parameter 3

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

Example

Operator keyin:

```
EN DUMP,JOBA
```

Function requested:

Terminates the execution of the dump being taken for JOBA.

Terminating a Job (STOP)

The STOP command terminates a specific job at the end of the current or specified job step. This command provides for orderly termination of the job.

Format

```
STOPΔjobname[,job-step-number]
```

Positional Parameter 1

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

Positional Parameter 2

job-step-number
Specifies the number of a job step that must complete before processing is suspended. This is an optional entry.

Examples

1. Operator keyin:

   ```
   ST JOBC
   ```

   Function requested:

   Terminates JOBC at the end of the currently executing job step.

2. Operator keyin:

   ```
   ST JOBC,02
   ```

   Function requested:

   Terminates JOBC at the end of job step 02.

4.6. Selected Occasion Operator Commands

During the course of processing a job, you 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 console workstation of all but outstanding output messages
- Change the system time or date
- Control software-detected hardware error logging
- Set an I/O device down or up as required for normal maintenance or device malfunction
- 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 $Y$DUMP file to an unlocked condition
- Set one or more blocks of main storage up or down
- Terminate system activity

- Specify resource management
- Verify the volume table of contents (VTOC)
- Set viewing privileges

The commands enabling you to perform these functions are described in the following subsections. Commands required to control interactive services, data communications, and system utility functions are described in Sections 5, 6, and 7.

4.6.1. Displaying Portions of Main Storage (DISPLAY)

The DISPLAY command displays selected areas of main storage at the console workstation or console. You usually enter the command when your system administrator needs a job address displayed for program debugging purposes.

```
DISPLAYΔaddr[,jobname]
```

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 name; otherwise, an absolute address is displayed.

Positional Parameter 2

jobname

Identifies the job name of the job in main storage whose relative address is displayed.

When no parameter 2 is entered, 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 console workstation or console:

```
addr[JOBNAME] contents-of-selected-addr Y,N,NXT?
```

where:

addr

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

JOBNAME

Identifies the address being displayed as a job-relative address and identifies the job region by the job's name. If JOBNAME 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:

```
addr[,jobname]
```

where:

addr

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

jobname

Identifies the job name of the job whose relative address is to be displayed.

When there are no more addresses to display, terminate the display symbiont by responding to the last display message with an N for none.

4.6.2. Displaying System Information (MIX)

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

Format

```
      ┌                                  ┐
      │   ┌                 ┐            │
MIXΔ  │DA │,{jobname       }│            │
      │   │ {symbiont-name }│            │
      │   └                 ┘            │
      │   ┌           ┐                  │
      │VI │,{did     }│                  │
      │   │ {c/ca/caa}│                  │
      │   └           ┘                  │
     {│SQ                                │}
      │SI                                │
      │DS[,did]                          │
      │SC                                │
      │FR                                │
      │MM                                │
      │EN                                │
      │EL                                │
      │HF (models 10/15/20 only)         │
      └                                  ┘
```

Positional Parameter 1

DA

Displays the following information on the console workstation or 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 the following information for the devices you specify in positional parameter 2 that have a mounted volume, or for all devices having a mounted volume, if you omit positional parameter 2:

- Device address of each device
- Volume serial number
- Mode setting of tape devices (U26 and U28 only)

SQ

Displays a list of outstanding symbiont requests, including unprocessed queued operator commands. (These are console commands that could not be processed immediately.) The displayed list includes a queue slot ID number that can be used to delete items from the outstanding symbiont queue (see DELETE command).

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 in decimal.

DS

Displays the following information on the console workstation or 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 if a diskette drive is available when the device requires maintenance; when the drive is in use, an I/O error occurs if its door is opened):

- 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)
- 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 (RUN processor, ICAM, etc.)
- RESERVED Region reserved for a job just scheduled
- BUF POOL Dynamic buffer pool. These storage pools are internally subdivided to provide dynamically allocated main storage for system software.

Note: *See Section 8 regarding disk cache storage allocations.*

- 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.

EL

Displays error log information, including error logging functions that are in effect, percentage of error records used, and number of records remaining.

HF

Displays the enabled or not enabled status of various model 10/15/20 hardware features (this parameter is not supported on other models). The displayed response indicates the model number (10, 15, or 20) and whether the following features are enabled (Y) or not enabled (N): bypass cache (BYP), floating point processor (FPP), instruction cache (ICM), and operand cache sets 1 through 4 (OC1 - OC4).

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 VI and DS parameters 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.

`caa`

Used with the VI parameter to indicate a device or group of devices. C specifies the channel; a, the address. One, two, or three characters may be used, as with did.

Example 1

Operator keyin:

```
MI DA
```

Function requested:

Display information pertaining to all active jobs and symbionts.

Screen display:

```
JOB NAME     JOB NO     PRI     DEV       TYPE          VSN/UID
RC$$IS00        0         0     312     3560 WRKSTN     LKT
```

The symbiont RC$$IS00 is recognized as a symbiont by a job slot number of 0. Its priority is 0, and it is allocated to device number 312, which is a 3560 workstation. LKT is the user-id.

Example 2

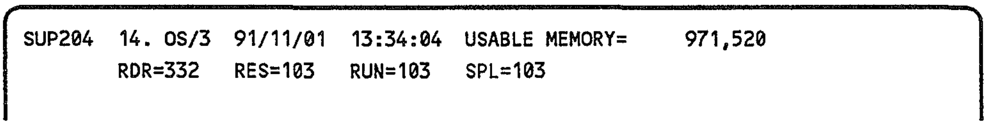
Operator keyin:

```
MI SI
```

Function requested:

Display system information.

Screen display:



```
SUP204  14. OS/3  91/11/01  13:34:04  USABLE MEMORY=     971,520
        RDR=332   RES=103   RUN=103   SPL=103
```

Supervisor SUP204 is running under release 14.0. The date and time is followed by the available main storage, which is 971,520 bytes. The reader device address is 332, the RES device address is 103, the RUN device address is 103, and the spool device address is 103.

Example 3

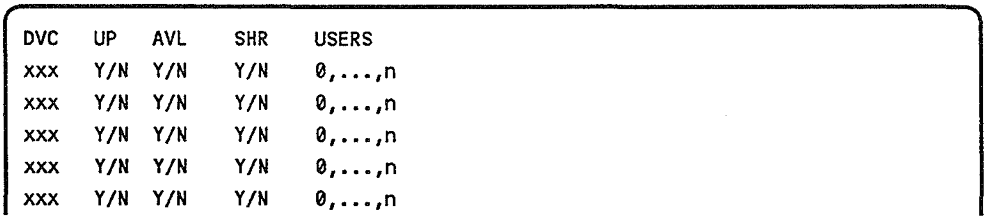
Operator keyin:

```
MI DS
```

Function requested:

Display information pertaining to all devices in the system.

Screen display:



```
DVC   UP   AVL    SHR    USERS
xxx   Y/N  Y/N    Y/N    0,...,n
xxx   Y/N  Y/N    Y/N    0,...,n
xxx   Y/N  Y/N    Y/N    0,...,n
xxx   Y/N  Y/N    Y/N    0,...,n
xxx   Y/N  Y/N    Y/N    0,...,n
```

The DVC column contains the device addresses of all the devices in the system. The next three columns tell you whether the device is up, available, and sharable. Also provided are the number of users of each device and the job slot numbers of all the jobs to which the device is allocated.

4.6.3. Reconstructing Console Display (REBUILD)

The REBUILD command clears all information from the console workstation, then restores the job number header lines and rewrites all outstanding question and action request output messages on the console workstation 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.

Note: *The REBUILD command does not restore the job number header lines on the console screen.*

4.6.4. Setting Simulated Day Clock (SET CLOCK)

The SET CLOCK command resets the time of day in the system-simulated day clock; it updates the hardware clock. The date and time are changed automatically at midnight of each day.

Format

```
SETΔCLOCK,hh:mm:ss
```

Positional Parameter 1

`CLOCK`
Sets the simulated day clock and hardware clock to the time specified in positional parameter 2.

Positional Parameter 2

`hh:mm:ss`
Specifies the time to set the simulated day clock and hardware clock as follows:

`hh`
Specifies hours (00 through 99).

`mm`
Specifies minutes (00 through 59).

`ss`
Specifies seconds (00 through 59).

4.6.5. Setting Date Field (SET DATE)

The SET DATE command resets the calendar date in the system information block date field, updates the hardware clock, and resets the job date for every job currently in main storage (except those jobs containing a // SET DATE job control statement). The date and time are changed automatically at midnight of each day.

Format

```
SET△DATE,yy/mm/dd[,yyddd]
```

Positional Parameter 1

DATE

Changes the calendar date in the system information block date field, the hardware clock, 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, the hardware clock, 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.

4.6.6. Setting Error Log (SET ELOG)

The SET ELOG command should not be used without first consulting your customer engineer. 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.

Format

```
SETΔELOG, ⎡ ⎧ON ⎫ ⎡ ,⎧ALL ⎫ ⎤ ⎤
          ⎢ ⎩OFF⎭ ⎢  ⎪COMM⎪ ⎥ ⎥
          ⎢       ⎢  ⎪MCHK⎪ ⎥ ⎥
          ⎢       ⎢  ⎪IO  ⎪ ⎥ ⎥
          ⎨       ⎢  ⎨TERM⎬ ⎥ ⎬
          ⎢       ⎢  ⎪CLOG⎪ ⎥ ⎥
          ⎢       ⎢  ⎪RSTK⎪ ⎥ ⎥
          ⎢       ⎢  ⎪MSE ⎪ ⎥ ⎥
          ⎢       ⎣  ⎩did ⎭ ⎦ ⎥
          ⎢   factor,⎧RSTK⎫   ⎥
          ⎣          ⎩MSE ⎭   ⎦
```

Positional Parameter 1

ELOG
: Specifies that the PIOCS error logging processor is to be set 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 logging of errors for the function specified by parameter 3.

factor
: Specifies a one- to three-character decimal value that is used to redefine main storage error (MSE) or retry stack error (RSTK) limits. The error is suppressed when its established limit is exceeded. The factor value entered is multiplied by 32 to redefine the millisecond time factor (F) in the error limit formula:

```
F*S:E
```

where:

F

Is the millisecond time factor to be redefined.

S

Is the size or number of MSE or RSTK errors.

E

Is the elapsed time (in milliseconds) since the last interrupt of the same type.

When F times S is greater than E, the reported type of suppressible machine error is disabled and a message to that effect is displayed on the console workstation or console screen.

Positional Parameter 3

ALL

Is valid only when ON is specified in parameter 2. ALL specifies that all loggable errors (i.e., communications, machine check, I/O device, and I/O termination record error logging) are to be logged for all devices in the system.

COMM

Turns on or off communications error logging as directed by parameter 2.

MCHK

Turns on or off machine check error logging as directed by parameter 2.

IO

Turns on or off all I/O device error logging as directed by parameter 2.

TERM

Turns on or off all I/O termination record error logging as directed by parameter 2.

CLOG

Turns on or off channel logout error logging as directed by parameter 2.

RSTK

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

ON

Turns on retry stack error logging.

OFF

Turns off retry stack error logging.

`factor`

Specifies that the factor specified in parameter 2 is to be used in the algorithm for establishing retry stack error limits.

<u>`MSE`</u>

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

`ON`

Turns on main storage error logging.

`OFF`

Turns off main storage error logging.

`factor`

Specifies that the factor specified in parameter 2 is to be used in the algorithm for establishing main storage error limits.

`did`

Represents a device address as one, two, or three hexadecimal characters specifying a channel, subchannel, and device, respectively. A 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 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,ON, the error logging function is turned on, but only for those parameter 3 specifications previously turned on.

Operator Considerations

When the ELOG file ($Y$ELOG) is almost full, the system displays the message:

```
LOG FILE IS NEARLY FULL
```

At this time you should run the system-supplied ONUERL program (Appendix A).

The ELOG file can become completely full only if the ONUERL program was not run when the nearly full message was displayed. If the ELOG file fills up, ELOG asks you whether you want to turn off error logging or wrap around to the beginning of the log file.

When the message

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

appears, respond using the solicited message format:

```
{O}
{W}
```

where:

```
O
```

Turns off the PIOCS error logging processor.

```
W
```

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

If ONUERL has already been run, respond with the letter W. If ONUERL has not been run, enter O to turn off logging, then initiate ONUERL.

Note: *It is recommended that you run ONUERL when the message LOG FILE IS NEARLY FULL is displayed. Allowing the ELOG file to wrap around causes loss of potentially valuable data.*

4.6.7. Discarding System Messages (FLUSH)

The FLUSH command allows you to discard system messages queued for delivery at your workstation. You can discard all queued messages or only those with a specified message prefix.

Format

```
FLUSH{*ALL      }
     {msg-prefix}
```

Parameters

```
*ALL
```

Discards all queued messages.

```
msg-prefix
```

Discards only messages with this prefix.

4.6.8. Setting Physical Unit Blocks (SET IO)

The SET IO command allows you to set specific bits in the physical unit blocks (PUB), which define operational characteristics and assignments of I/O devices. There is one physical unit block, comprising a three- or four-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 (NA). After IPL, any attention interrupt from a device causes that device to be set available (AV).*

Format

```
SETΔIO,did,⎡AV             ⎤ ⎡,⎧O⎫⎤
           ⎢CON            ⎥ ⎣ ⎩H⎭⎦
           ⎢DOWN           ⎥
           ⎢EON            ⎥
           ⎢EOF            ⎥
           ⎢FEA,type-code  ⎥
           ⎢HOME           ⎥
           ⎨NA             ⎬
           ⎢NOSHARE        ⎥
           ⎢NOWRITE        ⎥
           ⎢RDR            ⎥
           ⎢SHARE          ⎥
           ⎢TYPE,type-code ⎥
           ⎢UP             ⎥
           ⎣WRITE          ⎦
```

Positional Parameter 1

IO

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

Positional Parameter 2

did

Specifies a one- to four-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:

ALT	Alternate console specified in SYSGEN
caa	Alternate console specified by channel (*c*) and address (*aa*)
ORG	Original (hardware) console
100	SYSRES integrated disk
DDD	Indirect printer
nnn	Specific hardware device (channel/controller/device address)

nn. All devices on the channel/controller addresses specified

n. All devices on the channel address specified

Note: *Positional parameter 2 DDD can be associated only with positional parameter 3 UP or DOWN.*

Positional Parameter 3

AV

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

CON

Designates the specified device as the current console.

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.

The DOWN[,*n*] format is applicable only if the device identified in positional parameter 2 is an indirect printer. The *n* is a single-digit number indicating the indirect printer PUB on which the SET IO command is to be performed. If *n* is omitted, the first indirect PUB is assumed.

EON

Turns on error logging for the specified device.

EOF

Turns off error logging for the specified device.

FEA,type-code

Modifies the feature bytes of the device specified in positional parameter 2, where type-code is a one- to four-character field specifying the device, its options, and features desired. See the *Job Control Programming Reference Manual* (UP-9984) for the definition of type and feature codes.

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; it 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.

NOwrite

Specifies the device identified by positional parameter 2 (*did*) is write protected by the system.

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,type-code

Modifies the type bytes of the device specified in positional parameter 2, where type-code is a one- to four-character field specifying the device, its options, and features desired. See the *Job Control Programming Reference Manual* (UP-9984) for the definition of type and feature codes.

UP

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

The UP[,*n*] format is applicable only if the device identified in positional parameter 2 is an indirect printer. The *n* is a single-digit number indicating the indirect printer PUB on which the SET IO command is to be performed. If *n* is omitted, the first indirect PUB is assumed.

WRITE

Specifies the device identified by positional parameter 2 (*did*) is removed from its write protected state.

Notes:

1. *For more information on the indirect printer, see the* Installation Guide, *7004 5505.*
2. *Positional parameter 4 is required in order to set a device to available, not available, down, or up when your system is configured with stability monitoring.*

Positional Parameter 4

O

Indicates that the device status change specified in positional parameter 3 (AV, DOWN, NA, or UP) is required for operational reasons (for example, to set a device down to change forms or ribbons).

H

Indicates that the device status change specified in positional parameter 3 (AV, DOWN, NA, or UP) is required for hardware reasons (for example, to set a device to not available because the hardware is malfunctioning).

Example

Operator keyin:

```
SET I0,321,DOWN
```

Function requested:

Set the device identified by device number 321 down. Device 321 is a diskette. No screen display results after this command is issued. To determine if the device is now unavailable to user jobs, issue a MI DS command (see 4.6.2).

4.6.9. Reading a Mounted Volume Serial Number (AVR)

The AVR command initiates reading the volume serial number of a premounted prepped disk pack, diskette, or magnetic tape volume and storing it in the device physical unit block.

Format

```
AVRΔdid[,did][,did]
```

Positional Parameters 1 through 3

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

4.6.10. Displaying Job Status (DISPLAY JS/SY)

The DISPLAY JS command displays the status of jobs in the system at the console workstation or console. You can display the status of a specific job or all jobs in main storage, a specific job in a scheduling priority queue, a job being processed by the RUN or OCL processor, or all tasks attached to symbionts. The display includes the job name along with the CPU time used when the job is under execution, 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. It also includes the status of all subtasks attached to the job.

Format

```
DISPLAYΔ{JS,[jobname][,ALL]        }
        {SY[,symbiont-name][,ALL]}
```

Positional Parameter 1

JS
: Displays job status at the console workstation or console.

SY
: Displays at the console the status of all tasks attached to symbionts.

Positional Parameter 2

`jobname`

Displays the status of the job name specified. If omitted, the status of all jobs in main storage is displayed except for those on the job queues.

`symbiont-name`

Displays the tasks attached to the symbiont named. This name is the name displayed by the MIX DA command. If omitted, the tasks attached to all symbionts are displayed.

`ALL`

Displays the status of all tasks attached to a job or symbiont, instead of just the primary task.

Example

Operator keyin:

```
DI JS,MYJOB
```

Function requested:

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

Typical informational messages:

The following messages illustrate the information produced by the DISPLAY JS command when it is entered at various stages of jobs processing. The job name MYJOB is used for the purpose of example in the messages shown.

For the message:

```
MYJOB IN STEP 01(LNKEDT00)-PRI=10 CPU-TIME=00:01:43.874
```

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 processor for 1 minute, 43 seconds, and 874 milliseconds. If the job is proceeding, you can reenter DIΔJS for MYJOB and see an increase in the CPU TIME.

For the message:

```
MYJOB IN STEP 01(MYJOB0)-WTLOCK=JOBFILE
```

MYJOB is waiting to access the file JOBFILE. Another job accessing JOBFILE has defined a file share environment that is incompatible with the file share requirements specified by MYJOB. MYJOB must wait until its share requirements are compatible with the file share environment.

For the message:

```
MYJOB IN STEP 01(MYJOB0)-WAITING FOR TRANSIENT AREA
```

MYJOB requested a system function that requires a transient area. All transient areas are currently in use. MYJOB waits until a transient area is available for use.

For the message:

```
MYJOB IN STEP 01(MYJOB0)-JOB BEING MOVED (xxxx)
```

MYJOB is being acted upon by memory management and may be moved to a different address to allow better use of main memory. The values in parentheses are internal memory management values that are useful for software debugging.

For the message:

```
MYJOB IN STEP 01(MYJOB0)-WAITING UNTIL 12:00:00.000
```

MYJOB execution is suspended and will not restart until the time of day reaches that displayed in the message (12 noon in this example).

For the message:

```
MYJOB IN STEP 01(MYJOB0)-WAITING DUE TO PAUSE
```

MYJOB is paused by a PAUSE console command. MYJOB pauses until it is activated by a GO console command.

For the message:

```
MYJOB IN STEP 01(MYJOB0)-WAITING FOR IORB
```

MYJOB requires use of an input/output request block (IORB). All IORBs generated in the system are currently in use. MYJOB waits until an IORB is available for use.

For the message:

```
MYJOB IN STEP 01(MYJOB0)-WAITING FOR OPCOM SYSTEM MESSAGE
```

MYJOB issued a message to the system console or terminal. MYJOB waits until the supervisor processes the message.

For the message:

```
MYJOB IN STEP 01(MYJOB0)-WAITING DUE TO TASK PAUSE
```

A subtask of MYJOB issued a TPAUSE macro against the primary task of MYJOB. MYJOB pauses until a subtask issues a TGO macro against the primary task of MYJOB.

For the message:

```
MYJOB IN STEP 01(MYJOB0)-WAITING DUE TO TASK YIELD
```

The primary task of MYJOB issued a TYIELD macro to suspend execution. The primary task of MYJOB remains suspended until a subtask of MYJOB issues an AWAKE macro against the primary task to activate it.

For the message:

```
MYJOB IN STEP 01(MYJOB0)-WAITING FOR SHARED CODE LOAD
```

MYJOB requested a supervisor function be performed that requires a shared code module. The required shared code module is not currently in main memory. MYJOB waits until the required shared code module is loaded into main memory.

For the message:

```
MYJOB IN STEP 01(MYJOB0)-WAITING FOR DYNAMIC BUFFER
```

MYJOB requested a function that requires a dynamically allocated memory buffer for its processing. MYJOB waits until the supervisor has successfully allocated a storage buffer to MYJOB for its use.

For the message:

```
MYJOB IN STEP 01(MYJOB0)-CANCEL IN PROGRESS
```

MYJOB is in the process of being canceled from the system.

For the message:

```
MYJOB IN STEP 01(MYJOB0)-WAITING FOR OPERATOR REPLY
```

MYJOB issued a message to the system console or terminal that requires a reply. MYJOB waits until there is a reply from the console or terminal operator.

For the message:

```
MYJOB IN STEP 01(MYJOB0)-YIELDED TO ICAM
```

MYJOB issued a CYIELD macro and is waiting for activity from ICAM.

For the message:

```
MYJOB IN STEP 01(MYJOB0)-WAITING FOR MOUNT MSG
```

Job control or data management issued a message requesting that a volume be mounted on a specific tape or disk device for use by MYJOB. MYJOB waits until the volume is mounted and a console GO command is entered.

For the message:

```
MYJOB IN STEP 01(MYJOB0)-WAITING FOR SUB-TASK
```

The primary task of MYJOB suspended processing and waits until one or more of its subtasks completes processing before it begins executing.

For the message:

```
MYJOB IN STEP 01(LNKEDT)-ACTIVE AT PRIORITY 10
```

MYJOB is active in its first step, performing linkage editing. It is executing at priority level 10. SYSGEN option JOB ACCOUNTING is not configured in the supervisor currently loaded.

For the message:

```
MYJOB IN STEP 01(MYJOB0)-WAITING FOR 999xxxxx
```

MYJOB is waiting for a supervisor function to be performed on its behalf. All supervisor functions are displayed with the message that lists both the three-digit function number (999) and a five-character abbreviation (xxxxx) that describes it.

For the message:

```
MYJOB IN STEP 01(MYJOB0)-WAITING FOR LOCK (xxxxxxxx)
```

MYJOB is waiting because a system resource that it requires is currently locked by another job. The job waits until the system lock is released. The current system locks are as follows:

VTOC	System VTOC lock
PUBS	Physical unit block (PUB) allocation lock
JOB QTBL	Job queue table lock
VOL TBL	Volume use table lock
S.R.Q.	Symbiont request queue lock
$Y$SHARE	$Y$SHARE file lock
N.T.R.	System NTR lock
SCOPE	Scope table lock

For the message:

```
MYJOB IN STEP 01(LNKEDT00)-PRI=10 CPU-TIME=00:01:43.874
```

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 displayed.

For the message:

```
MYJOB IN STEP 02(LIBS0000)-WAITING FOR I/O #00005736
```

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 03 -IN STEP PROCESSOR
```

MYJOB is between job steps. Step 03 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 is not scheduled for execution because not enough main memory is available.

For the message:

```
MYJOB IS ROLLED OUT
```

MYJOB was removed from main memory to allow another job, which was scheduled with preemptive priority, to be loaded into main memory. MYJOB remains in the rollout status until there is enough main memory for its requirements.

For the message:

```
MYJOB JUST SCHEDULED-NOW BEING INITIATED
```

All system resources that MYJOB requires are available and assigned to it. MYJOB was scheduled and is currently being loaded into main memory to begin execution.

For the message:

```
MYJOB WAITING FOR QUEUED SYMBIONT xx
```

All the system resources needed by the symbiont in question for processing MYJOB are currently not available. The symbiont is placed on the symbiont request queue until the required system resources are available. Values for xx are:

RV	RV version of the run processor
RU	RU version of the run processor
SC	SC version of the save processor
SI	SI version of the save processor
SV	SV version of the save processor

4.6.11. Dumping the Contents of Main Storage (SYSDUMP)

The SYSDUMP command is used to dump the entire contents of main storage to the $Y$DUMP file on SYSRES. Use this command whenever a system dump is required without supervisor reloading (no manual IPL is required). After the contents of main storage are dumped, the job SYSDMP*nn* (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 SYSDMP*nn* job completes, to prevent other system functions (caused, for example, by an // OPTION SYSDUMP statement or by system errors) from also using the file (see the *Dump Analysis Programming Guide* (UP-9980).)

If you don't want a printout of the dump, enter NONE when the SYSDMP*nn* job asks what type of dump should be printed. If you delete the SYSDMP*nn* job from the job queue or cancel it before it sends you the SD01 DUMP OPTION message, you must enter the SET SY command to unlock the $Y$DUMP file.

You can't use the console workstation or 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
```

Notes

1. *If a nonrecoverable error occurs and you want a dump of everything in main storage at that time, press the ESC key, then the M key to display the manual frame. Then key in L for system reset and press the XMIT key. Next, key in U for run and press the XMIT key.*

   If you get an HPR of 999F, it means the $Y$DUMP file contains data from a previous main storage write operation that hasn't been processed by SYSDUMP or SYSDUMPO. You can overwrite the contents of the file by keying in U for run on the manual frame and pressing XMIT, or you can follow the next procedure.

2. *If there's information in the $Y$DUMP file that you want to preserve and a nonrecoverable error occurs, perform the IPL procedure in Section 2. This reinitializes the system without affecting the $Y$DUMP file. To print the contents of $Y$DUMP, run the SYSDUMP or SYSDUMPO job stream when the IPL procedure is completed.*

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

The SET SY command unlocks the $Y$DUMP file after an SY command (4.6.10) 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 set.

Positional Parameter 2

```
LOFF
```
Specifies the $Y$DUMP file is unlocked.

Note: *No dump is provided in stand-alone IPL.*

4.6.13. Setting Main Storage Condition (SET MEM)

The SET MEM command sets one or more blocks of main storage up or down. The length of one block is determined by the replacement element size (RES), that is, the size of the replacement element that contains the main storage block determined unusable. Normally, main storage is in an up (usable) condition. The system automatically determines when a main storage block becomes unusable. The system sets the block down and displays a listing on the console workstation or console screen of all main storage blocks currently set to down. The down block is unavailable for use by new jobs or symbionts until it is set to up via the SET MEM command. However, any jobs or symbionts currently running in the down block continue to run until completed.

The map of usable storage is kept in the error log. If you add main storage to your system or take a system-resident pack from a smaller system to a larger system and the error log is not initialized, the additional main storage will not be usable. To make the main storage usable, initialize the error log when you perform an IPL or use the SET MEM command to set the additional main storage up.

You enter the SET MEM command at the request of your system administrator to set a main storage block up or down as required. Normally, SET MEM is used to set a block to up after your Unisys customer engineer corrects the condition that made it unusable.

Format

```
SET∆MEM,┌DOWN┐,address ┌,┌no-of-blocks┐┐
        └UP  ┘         └ └1           ┘┘
```

Positional Parameter 1

MEM
: Specifies main storage is to be set up or down.

Positional Parameter 2

DOWN
: Sets one or more blocks of main storage down.

UP
: Sets one or more blocks of main storage up.

Positional Parameter 3

address
: Specifies the beginning address, in hexadecimal, of a block to be set up or down. This address must be on a RES boundary (a multiple of the replacement element size).

Positional Parameter 4

no-of-blocks
: Specifies the number of contiguous blocks, in decimal, set up or down beginning with the address specified in positional parameter 3. If omitted, one block is assumed.

4.6.14. Setting the Unattended Console Feature (SET UNCON)

The unattended console feature is a SYSGEN option that allows messages requiring operator intervention to be answered by the operating system if you do not respond in 15 minutes. These messages are informational only and pertain to device I/O errors. The unattended console feature decreases the need for operator intervention by requiring operator responses only in extreme cases. The SET UNCON command allows you to remove the previous SYSGEN option, reactivate it once it has been removed, and change the unattended console auto-answer time of 15 minutes.

Format

```
SET∆UNCON┌OFF ┐
         {ON  }
         └time┘
```

Positional Parameter 1

UNCON

Specifies that the unattended console feature enabled during SYSGEN is to be disabled, reactivated, or changed.

Positional Parameter 2

OFF

Specifies that the previously enabled (during SYSGEN) unattended console feature is to be removed.

ON

Specifies that the unattended console feature enabled at SYSGEN and removed by a previous SET UNCON command is to be reactivated.

time

Specifies the time in minutes for the unattended auto-answer. If this parameter is not specified, the time is 15 minutes.

4.6.15. Terminating System Activity (SHUTDOWN)

The SHUTDOWN command terminates system activity or distributed data processing (DDP) in an orderly manner.

Format

```
SHUTDOWNΔ⎧SYSTEM⎫
         ⎩DDP   ⎭
```

Positional Parameter 1

SYSTEM

Specifies that system activity will be terminated. 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. This command *cannot* be abbreviated; the entire word SYSTEM is required to protect the user against accidental keyin and shutdown.

DDP

Specifies that only distributed data processing will be terminated when its activity ceases. 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 (see 5.2.4).

4.6.16. Initiating Transient Work Area Feature (TW)

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 of 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=nnn]
```

Positional Parameter 1

`SZ=nnn`

Specifies the number (*nnn*) of 1,024-byte blocks for the work area, where *nnn* must be in the range of 32 to 250. Any value outside of this range causes your system to terminate the transient work area and display the message INVALID PARAMETER. If omitted, 65K is allocated for the work area by default.

You can initiate the transient work area whenever sufficient free main storage is available to satisfy the work area space requirements. However, you should initiate TW immediately after the IPL procedure to avoid possible fragmentation of main storage.

Once the transient work area is activated, the message TRANSIENT WORK AREA IS INITIALIZED appears. If you receive this message during the IPL procedure, it indicates that the work area in your supervisor generation has already been allocated to the specified size 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:

```
00ΔTWΔEOJ
```

The message TRANSIENT WORK AREA TERMINATED then appears.

4.6.17. Specifying Resource Management (LIMITS)

The LIMITS command lets you control the percentage of your system's main storage space used for symbionts, interactive processing, and batch processing. This enables you to maximize system turnaround time and workstation response. Resource management lets you control these activities:

- Number of batch jobs run from both the console and workstations
- Number of batch jobs run from the workstations only

- Number of logged-on workstations (including workstations logged on to both interactive services and OS/3 interactive)
- Number of ENTER processing tasks
- Number of RUN symbionts

You can issue the LIMITS command at any time during the session. However, all parameters of this command (except MAXLOGONS and UPTERMINAL) require that resource management be generated at SYSGEN. Otherwise, the parameters are considered invalid. MAXLOGONS and UPTERMINAL parameters are valid only when security is generated.

If your system is primarily a batch or interactive system, you can specify resource management parameters during system generation (SYSGEN). See the *Installation Guide*, 7004 5505, for a description of the resource management SYSGEN parameters.

If your system operations change throughout the day or week from batch to interactive, or you use a mixture of both, then use the LIMITS command to change your resource management controls as your processing needs change. When you reuse the LIMITS command, any parameters not specified remain unchanged.

Before using the LIMITS JOBMEM command to reduce the percentage of main storage available for batch jobs, be sure there is sufficient main storage for the jobs in the current running system. The MAXJOBS limit can be temporarily lowered to insure that no new jobs are initiated. After some jobs have terminated normally and the percentage of main storage in use falls below the new target level, then you can issue the LIMITS JOBMEM command and reset the MAXJOBS limit. This precaution helps avoid jobs from aborting due to insufficient memory.

Format

```
LIMITSΔ┌                    ┐┌                    ┐┌                    ┐
       │SYMBMEM=┌5-100┐     ││,INTMEM=┌5-100┐     ││,JOBMEM=┌0-100┐     │[,MAXJOBS=1-n]
       │        └NLMT ┘     ││        └NLMT ┘     ││        └NLMT ┘     │
       └                    ┘└                    ┘└                    ┘
        [,MAXWSJOBS=0-n][,MAXSWSJOBS=0-n][,MAXINTUSERS=0-255][,MAXENTERS=0-255]

        [,MAXRUNSYMBS=1-10]┌                    ┐┌                       ┐
                           │,MAXLOGONS=┌1-99┐   ││,UPTERMINAL=┌tttt┐     │
                           │           └5   ┘   ││            └*ALL┘     │
                           └                    ┘└                       ┘
```

Parameters

```
SYMBMEM=┌5-100┐
        └NLMT ┘
```

Specifies the percentage of available main storage your system uses for symbiont use. Either specify a percentage value between 5 and 100, or specify NLMT. NLMT specifies that resource management does not control that area of main storage allocated to symbionts. The value you specify can be greater than or less than the SYSGEN value.

```
INTMEM={5-100}
       {NLMT }
```

Specifies the percentage of available main storage your system uses for interactive use. Either specify a percentage value between 5 and 100, or specify NLMT. NLMT specifies that resource management does not control that area of main storage allocated for interactive use. The value you specify can be greater than or less than the SYSGEN value.

```
JOBMEM={0-100}
       {NLMT }
```

Specifies the percentage of available main storage your system uses for batch use. Either specify a percentage value between 0 and 100, or specify NLMT. NLMT specifies that resource management does not control that area of main storage allocated for batch use. The value you specify can be greater than or less than the SYSGEN value.

```
MAXJOBS=1-n
```

Specifies the number of jobs that can be executed concurrently. The value you choose cannot exceed the value you have specified for the SYSGEN JOBSLOTS parameter. The maximum number of jobs is 48.

```
MAXWSJOBS=0-n
```

Specifies the maximum number of jobs that can be executed concurrently from workstations only. The value you choose cannot exceed the value you have specified for the SYSGEN MAXJOBS parameter, or the last LIMITS command MAXJOBS parameter.

```
MAXSWSJOBS=0-n
```

Specifies the maximum number of jobs that can be initiated concurrently from any single workstation. The value you choose cannot exceed the value you have specified for the SYSGEN MAXWSJOBS parameter or the last LIMITS command MAXWSJOBS parameter.

```
MAXINTUSERS=0-255
```

Specifies the maximum number of interactive users that can be logged on at one time. This parameter changes the value of the ISINTLMT parameter defined at SYSGEN or set with a previous LIMITS command.

```
MAXENTERS=0-255
```

Specifies the maximum number of batch tasks initiated with the ENTER command that can execute concurrently in the system. This parameter changes the value of the ISBATCHLMT parameter defined at SYSGEN or set with a previous LIMITS command.

```
MAXRUNSYMBS=1-10
```

Specifies the maximum number of RUN symbionts that can be concurrently executed. This parameter changes the value of the MAXRUNSYMBS parameter defined at SYSGEN or set with a previous LIMITS command.

```
MAXLOGONS=⎧1-99⎫
          ⎩5   ⎭
```

Specifies the maximum number of invalid logon attempts before a terminal is deactivated. The default value is 5.

```
UPTERMINAL=⎧tttt⎫
           ⎩*ALL⎭
```

Specifies the terminal (*tttt*) to be reactivated. *ALL reactivates all terminals, whether they have been deactivated or not.

Examples

1. Let's say your system is used primarily for interactive jobs in the morning, for both batch and interactive jobs in the afternoon, and then only batch jobs in the evening.

   Upon user requests for a better balance between system turnaround time and workstation response time during each of these shifts, the operator issues the LIMITS command.

   First, before the morning shift, he issues:

   ```
   LIMITS JOBMEM=20,INTMEM=80,MAXRUNSYMBS=1
   ```

   This example shows a small value for the JOBMEM parameter, the parameter that specifies the percentage of main storage to be used for batch jobs. It shows a high value for INTMEM, the parameter that specifies the percentage of main storage to be used for interactive jobs. And it shows a low value for the number of RUN symbionts that can execute concurrently.

   Next, before the afternoon shift, the operator issues:

   ```
   LIMITS JOBMEM=70,INTMEM=70,MAXRUNSYMBS=2
   ```

   This example shows a balance of the percentage of main storage to be used for batch and interactive use. (Both JOBMEM and INTMEM are specified as 70). This time, the operator increases the number of RUN symbionts that can execute concurrently.

   Finally, before the evening shift, the operator issues:

   ```
   LIMITS JOBMEM=80,INTMEN=10,MAXRUNSYMBS=3
   ```

   This example shows a much higher percentage of main storage to be used for batch use (JOBMEM is specified as 80; but, INTMEM is specified as only 10.) The operator increases the number of RUN symbionts that can execute concurrently to 3.

2. The following example reactivates workstation C12 and changes the invalid logon limit to 3:

   ```
   LI  UP=C12,LOG=3
   ```

4.6.18. Verifying and Correcting the VTOC (VV)

The preventive maintenance procedures, VTOC verification and VTOC correction, are called using the VV system command. The VTOC verification routine determines whether volume table of contents (VTOC) corruption has occurred on an OS/3 disk. It performs a thorough analysis of a VTOC, looking for any inconsistencies between the labels in the VTOC. When it detects an inconsistency, it displays a message providing details on the error.

Use this routine under the following circumstances to detect VTOC problems as soon as possible:

- Initially, to verify that all volumes pass verification.

- After a system stop, to verify that all volumes in use at the time of the stop pass verification.

- When one of the error codes described under "Symptoms of Compromised VTOC" is experienced.

The VTOC correction routine can correct volume label errors (VV08) only. See the VV08 message description in the *System Messages Reference Manual*, 7004 5190, for a list of the specific error codes that are correctable.

It is not necessary to prevent other jobs from accessing the device specified when using the VTOC correction routine; however, you should use this facility only during periods of minimal activity in the system.

Before using the VTOC correction routine, you should make an alternate copy of the VTOC being corrected. (See Note 5, Figure 4-2.)

Format

```
VVΔ⎡did⎡,⎧END ⎫⎤[,ALT]⎤
   ⎢   ⎢ ⎪FIX ⎪⎥      ⎥
   ⎢   ⎢ ⎨ECnn⎬⎥      ⎥
   ⎢   ⎢ ⎪COPY⎪⎥      ⎥
   ⎣   ⎣ ⎩PACK⎭⎦      ⎦
```

Positional Parameter 1

`did`

Specifies a one- to four- character numeric field identifying the device address of the device whose VTOC is to be verified. If you omit this parameter, the system prompts you for the device address.

The VV command automatically locks the specified VTOC to prevent it from being accessed while it is being read or updated.

Positional Parameter 2

END

Terminates the routine when it completes the verification. If you omit this optional parameter:

- The system prompts you for another device address when the VTOC successfully passes verification.
- The routine terminates if verification fails.

FIX

Calls the VTOC correction routine and performs the following steps:

1. Complete VTOC verification
2. VTOC correction
3. Reverification

If the error encountered is not correctable, the system informs you that the error cannot be automatically corrected (VV22).

If you omit the FIX parameter, only verification is performed.

Note: *It is recommended that you specify the FIX parameter after you detect an error. It is also suggested that you correct an alternate VTOC (VVdid,FIX,ALT) as a first trial correction before correcting your VTOC file.*

ECnn

Causes a dump to be taken when the error code *nn* is encountered.

For example,

```
VV 190,EC18
```

causes a dump to be taken when VV08 EC18 is encountered. (See the *System Messages Reference Manual*, 7004 5190.)

The symbiont produces an error code 33A, calls SYSDUMP to print the dump, and continues processing. (Select the MINI SYSDUMP option.)

Use this debug facility to provide Unisys with additional documentation for an unexplained error condition.

COPY

Copies the VTOC to a file named ALT$VTOC. You must create this file before using the COPY parameter.

PACK

Indicates that the VTOC is to be packed (i.e., reorganized). Format 1 labels are searched on all ALLOCATE, SCRATCH, EXTEND, OPEN, and CLOSE file management operations. The pack process consists of moving the Format 1 labels to the front of the VTOC to reduce search times when performing these common operations.

First a verify is performed to guarantee that the VTOC passes verification. Then the VTOC is copied to the alternate VTOC file and the VTOC is reorganized. Should a problem occur during the pack process, the VTOC is restored to its original contents. Do *not* cancel VV while a pack function is in progress.

The alternate VTOC file (ALT$VTOC must be allocated prior to performing the pack.

You should use PACK only when the device in question is *not* being used by any other jobs (i.e., VV can get exclusive use of the device). If you pack the VTOC on a system disk (i.e., RES, RUN, or SPOOL), you must guarantee that the system is *completely* idle and, immediately upon completion of the pack, you must re-IPL the system because there may be internal tables that contain invalid (i.e., old) format label addresses. You can pack multiple system disks in succession without a re-IPL after each one, but you must pack the RES device last and then re-IPL immediately.

Positional Parameter 3

ALT

Performs the specified routine on the alternate VTOC (ALT$VTOC). The VTOC must be copied to the alternate file, using the COPY parameter, before the ALT parameter is used.

This parameter is optional. If omitted, the specified routine is performed on the original VTOC.

When you specify ALT, all VTOC verify messages (information and error) include the label ALT$VTOC at the end of the message.

Note: *It is suggested that you correct an alternate VTOC (VV did,FIX,ALT) as a first trial before correcting your VTOC file.*

The volume table of contents (VTOC) is a file that resides on every disk and serves as a directory for all files on that disk. The address of the VTOC is contained in the VOL1 label, which resides in a fixed location on each disk. Each physical record (i.e., sector) in the VTOC file is one of six types of format labels.

The following summarizes the various format label types, presented according to their logical progression.

- FORMAT 4 (F4) - Contains device-dependent information about the volume and a pointer to the F0 chain. There is only one F4 label.
- FORMAT 5 (F5) - Contains information regarding available space on the disk. There can be multiple F5 labels.
- FORMAT 0 (F0) - An unreserved label type available for label usage by the system as necessary. There can be multiple F0 labels that are logically linked together by pointers.
- FORMAT 1 (F1) - Contains information regarding the physical space allocated to a specific file (i.e., physical extents).
- FORMAT 2 (F2) - Contains supplementary F1 information that specifies how the physical space allocated to the file is subdivided (i.e., logical extents).
- FORMAT 3 (F3) - Contains additional F1 logical extent information.

Cause of Compromised VTOC

The VTOC is updated when a new file is added or deleted, or when an existing file requires that two or more labels in the VTOC be updated. If the update does not complete due to a system stop (HPR) or unrecoverable I/O error, the VTOC may be left in a compromised state, which can lead to files being destroyed.

Symptoms of Compromised VTOC

- Error codes 30, 31, or 32

  Caused by an unrecoverable I/O error during a VTOC update.

- Error code 3A

  A definite indication of a compromised VTOC. Disk space management (DSM) reports this error when it detects a label inconsistency.

- Error codes 35 or 36 or DM41

  Generally occurs when there are no available (F0) labels or no available space. In this case, these errors are generally not a symptom of a compromised VTOC. However, if they occur when there should be labels or space available, then it could be a sign of a compromised VTOC.

- Invalid data

  If invalid data (e.g., data from a different file) is being returned to a program, this could be sign of a compromised VTOC.

Error Conditions during Verification

Several conditions prevent the verification routine from performing. When any of these is encountered, it is identified by a VV07 message followed by a VV02 message prompting you to enter a device address. Correct the condition specified by the VV07 message and reenter the device address.

Volume and file type errors are identified by VV08 and VV09 messages, respectively. It is possible to get several VV08 or VV09 messages or a combination of both.

During verification, the routine attempts to identify all volume and file type error conditions. However, certain error conditions may be too serious for the routine to continue. Correct the conditions listed, then perform the VTOC verification routine again to identify any remaining VV08 and VV09 errors.

Refer to the *System Messages Reference Manual*, 7004 5190 for an explanation of VTOC verification messages and their error codes.

Corrective Action

The flow chart in Figure 4-2 shows the procedure to follow whenever a disk fails the VTOC verification routine.

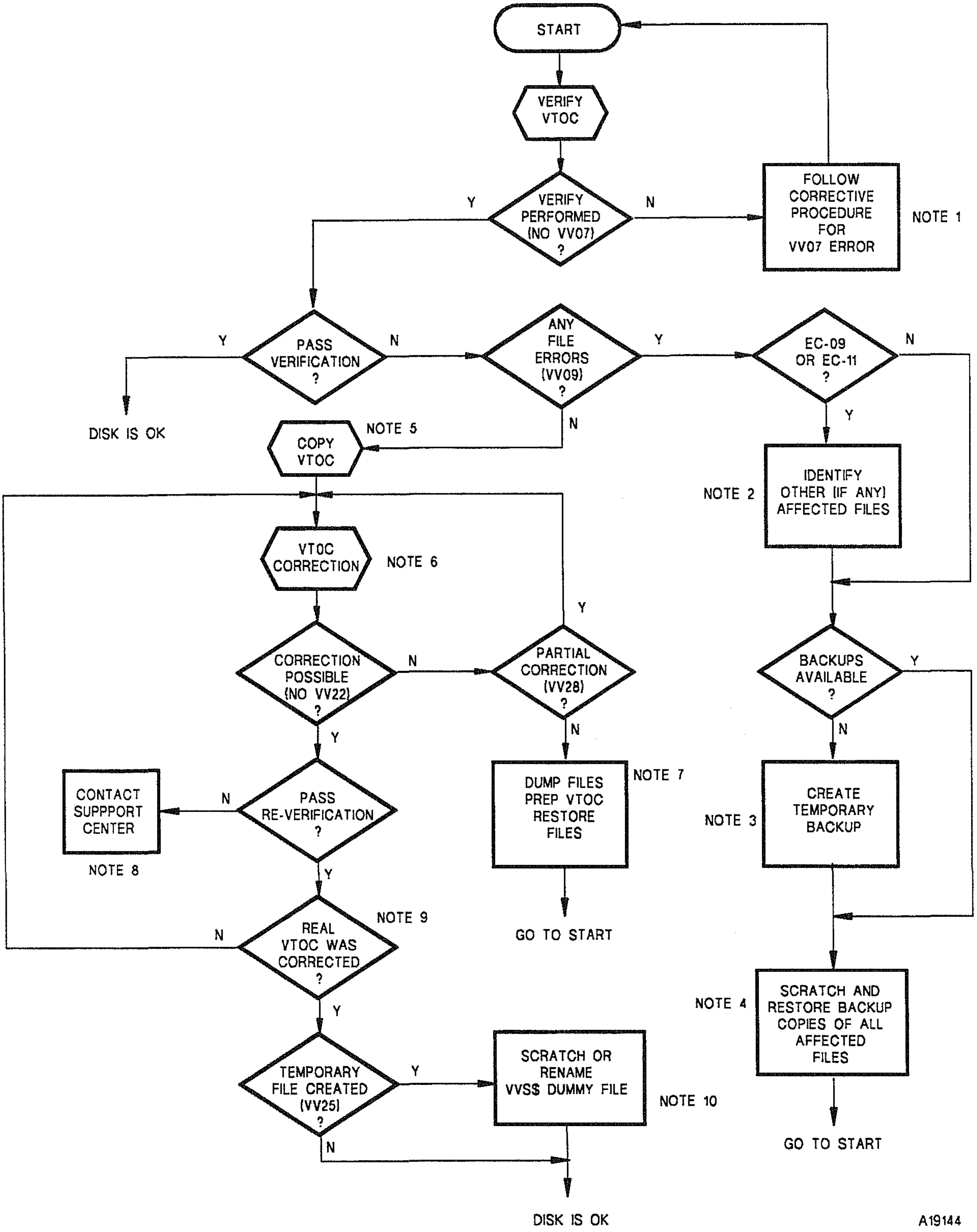

START
VERIFY VTOC
VERIFY PERFORMED (NO VV07) ?
Y
N
FOLLOW CORRECTIVE PROCEDURE FOR VV07 ERROR
NOTE 1
PASS VERIFICATION ?
Y
N
DISK IS OK
ANY FILE ERRORS (VV09) ?
Y
N
EC-09 OR EC-11 ?
N
Y
COPY VTOC
NOTE 5
NOTE 2
IDENTIFY OTHER (IF ANY) AFFECTED FILES
VTOC CORRECTION
NOTE 6
CORRECTION POSSIBLE (NO VV22) ?
N
Y
PARTIAL CORRECTION (VV28) ?
Y
N
BACKUPS AVAILABLE ?
Y
N
CONTACT SUPPPORT CENTER
NOTE 8
PASS RE-VERIFICATION ?
N
Y
DUMP FILES PREP VTOC RESTORE FILES
NOTE 7
GO TO START
NOTE 3
CREATE TEMPORARY BACKUP
REAL VTOC WAS CORRECTED ?
NOTE 9
N
Y
NOTE 4
SCRATCH AND RESTORE BACKUP COPIES OF ALL AFFECTED FILES
TEMPORARY FILE CREATED (VV25) ?
Y
N
SCRATCH OR RENAME VVS$ DUMMY FILE
NOTE 10
GO TO START
DISK IS OK
A19144

Figure 4-2. VTOC Verification/Correction Procedure (Part 1 of 2)

NOTES:

1. Some W07 errors are user errors, such as wrong device address or device not on-line.

2. If you get a W09 message with error code 09 or 11, other files besides the ones identified in the message may be affected.

   For example, when a cylinder is assigned to two different files, a W09 message is displayed for the second file. The verification routine cannot determine which file is corrupted. You must use the SU VTP listing to determine whether there are other files whose disk space allocations (physical or logical) overlap the allocations of the file named. If so, these files might have been affected also.

3. If you do not have a previously created backup copy of the affected files, create a temporary backup by copying the files onto another disk. Remember, however, that the integrity of this copy may be affected by the incorrect labels.

4. Restore all potentially corrupted files (identified by W09 messages or as in note 2 above) from a previously created backup copy using a copy program such as DATA utility or DUMP/RESTORE in file mode.

5. Using the COPY parameter, make an alternate VTOC for a trial correction.

6. Make two passes through the correction routine: the first, a trial correction on the alternate VTOC and the second, the correction of your VTOC file. If the trial correction is unsuccessful, your VTOC file is not affected.

7. Dump each file on the disk (except those identified by W09 messages or identified as in note 2) to another medium using the DUMP/RESTORE utility in file mode. Then prep the VTOC of the bad Disk and restore all of the files just dumped.

8. Should this situation ever occur, verify the disk using the ECnn parameter. Send the resulting dump and a copy of the old VTOC (on tape or diskette) to the Unisys Support center.

9. If the correction was performed against the alternate VTOC, correct your VTOC file.

10. Space that was not accounted for in the previous step is assigned to a temporary file (W$$DUMMY), giving you the opportunity to access that data. If you want to keep the data, rename the file. If not, you can scratch the file and return the space to the free space list.

Figure 4-2. VTOC Verification/Correction Procedure (Part 2 of 2)

Verifying the VTOC during AVR (SET VV)

The verify routine can be invoked explicitly via the system command VV or implicitly whenever an automatic volume recognition (AVR) is performed. AVR is performed under the following circumstances:

- During IPL
- Whenever a disk is set online
- Explicitly, via the AVR command

Automatic VTOC verification AVR time can be requested in two ways:

1. Via the VVAVR SYSGEN parameter. See the *Installation Guide*, 7004 5055, for details on the VVAVR parameter.
2. Via the SET VV command.

Format

```
SET VV,{ON }
       {OFF}
```

Positional Parameter 1

ON

Performs automatic VTOC verification at AVR time.

OFF

Does not perform automatic VTOC verification at AVR time. This overrides the VVAVR SYSGEN specification, described in the *Installation Guide*, 7004 5505.

Error Reporting to Unisys Support Center

If a volume that previously passed verification suddenly fails verification when there were no system stops or unrecoverable I/O errors to cause the VTOC to be compromised, report this occurrence to the Unisys support center. An analyst at the support center will record your information, assign you a User Communication Form (UCF) authorization number, and submit a UCF in your behalf.

When mailing support documentation to the support center, include the release level of your software and the UCF number assigned by the analyst Also enclose as much information as possible about the jobs that accessed the volume that was compromised, including:

- A formatted VTOC print (SU VTP) and a disk print of the VTOC (SU DD) taken both before and after the program is run

- The JCL for the jobs in question
- A console log and an error log
- Any additional information useful in duplicating the problem (for example, operational information, copies of programs, or copies of disks)

Procedures for reporting error conditions to a Unisys support center are provided in the System Technical Support Overview, 7002 3916.

4.6.19. Setting Memory Consolidation Conditions (SET MC)

The SET MC command enables or disables the memory consolidation (MEMCON) feature. MEMCON is enabled to allow movable jobs to be consolidated at the highest possible addresses, thus creating the largest possible free regions. All movable job task control blocks (TBCs) must be inactive prior to moving a job. In a system that is not fully utilizing its memory resources, memory consolidation may not be desirable. In this case, MEMCON should be disabled.

Ongoing MEMCONs at the time of the keyin are not affected by the command.

Format

$$\texttt{SET MC,}\begin{Bmatrix}\texttt{ON}\\\texttt{OFF}\end{Bmatrix}$$

Positional Parameter 1

`MC`

Specifies that the MEMCON condition is to be altered.

Positional Parameter 2

`ON`

Enables memory consolidation.

`OFF`

Disables memory consolidation.

4.6.20. Setting Job Immovability Conditions (SET IM)

The SET IM command enables or disables the job immovability (IMVJOB) feature. Some jobs present a problem to MEMCON because they become immovable. Normally IMVJOB is enabled to move those jobs to the high addressed portion of memory. This reduces the possibility that a particular job will become a roadblock to an ongoing memory consolidation. To perform this move, several jobs may have to be idled simultaneously, thus temporarily preventing any processing by those jobs involved. If this condition is not desirable to the operator, IMVJOB can be disabled by this command.

An ongoing move at the time of the keyin is not affected by the command.

ROLLOUT must have been configured for the current supervisor in order for IMVJOB to be enabled by this command.

Format

```
SET IM,{ON }
       {OFF}
```

Positional Parameter 1

`IM`
: Specifies that the IMVJOB condition is to be altered.

Positional Parameter 2

`ON`
: Enables job immovability moves.

`OFF`
: Disables job immovability moves.

4.6.21. Displaying Buffer Pool Information (DI BI)

The DI BI command displays system buffer pool information at the system console.

Format

```
DI BI
```

Example

The following is a sample system console display of buffer pool information for the DI BI command.

```
33 BR0022     BREAKPOINT TAKEN FOR CONSOLE LOG 88/09/08
DI BI
   EXPANSION REGION SIZE (BYTES)...................       4,096           DI0024
   RESERVE REGION SIZE (BYTES).....................         500           DI0024
   DISK CACHE BUFFER SIZE (BYTES)..................   2,050,048           DI0024
   TRANSIENT CACHE BUFFER SIZE (BYTES).............           0           DI0024
   TOTAL JOB/SYMBIONT BUFFER SIZE (BYTES)..........      59,272           DI0024
   TOTAL ALLOCATED BUFFERS (BYTES).................   2,109,320           DI0024
   CURRENT NUMBER OF BUFFERS ALLOCATED.............          43           DI0024
   LARGEST NUMBER OF BUFFERS ALLOCATED.............          49           DI0024
   LARGEST NO. OF JOB/SYMB. BUFFERS USED (BYTES)...      77,812           DI0024
   DISPLAY END                                                            DI0024
```

where:

EXPANSION REGION SIZE (BYTES)
Specified in a SYSGEN/IPL parameter.

RESERVE REGION SIZE (BYTES)
Specified in a SYSGEN parameter.

DISK CACHE BUFFER SIZE (BYTES)
Specified in a SYSGEN/IPL parameter/console command.

TRANSIENT CACHE BUFFER SIZE (BYTES)
Specified in a SYSGEN parameter/console command.

TOTAL JOB/SYMBIONT BUFFER SIZE (BYTES)
Is the total size of buffers currently allocated to jobs or symbionts.

TOTAL ALLOCATED BUFFERS (BYTES)
Is the total size of disk cache, transient cache, and job/symbiont buffers.

CURRENT NUMBER OF BUFFERS ALLOCATED
Is the number of buffer regions currently allocated in memory.

LARGEST NUMBER OF BUFFERS ALLOCATED
Is the largest number of buffer regions allocated in memory since the last IPL.

LARGEST NO. OF JOB/SYMB. BUFFERS USED (BYTES)
Is the largest number of buffer allocated to jobs or symbionts since the last IPL.

4.6.22. Enabling/Disabling the Console/Workstation Alarm Beeper (SET AL)

This command turns alarm beeper feature on or off at the console or workstation.

Format

```
SET ALARM, ⎧ONE ⎫
           ⎨ON  ⎬
           ⎩OFF ⎭
```

Parameters

ONE
Turns the alarm beeper feature on. Only a single warning beep is sounded.

ON
Turns the alarm beeper feature on. Continuous warning beeps are sounded.

OFF
Turns the alarm beeper feature off.

4.6.23. Enabling/Disabling the Console Video Clock (SET VC)

This command turns on or off the console time-of-day clock. This clock is located at the upper right corner of the console and is normally incremented every 10 seconds.

Format

```
SET VC,{ON }
       {OFF}
```

Parameters

ON
: Turns on the console time-of-day clock.

OFF
: Turns off the console time-of-day clock. The clock display goes blank and it is no longer incremented.

4.6.24. Changing Symbiont Execution Priorities (SET PS)

This command assigns an execution priority to a specific symbiont. The priority remains in effect until it is either changed by another SET PS command or the system is rebooted.

Format

```
SETΔPS,symbiont-name,n
```

Parameters

symbiont name
: Two- or three-character name identifying the affected symbiont.

n
: A value from 0 to 59 indicating the execution priority of the symbiont identified by parameter 1. This value cannot exceed the value specified by the PRIORITY parameter specified in SUPGEN.

4.6.25. Setting Console Message Viewing Privileges (SET CV)

This command allows privileged users to view the console messages sent to the operator in addition to their own messages.

Format

```
SETΔCV,{ON }
       {OFF}
```

Parameters

`ON`
Turns viewing privileges on.

`OFF`
Turns viewing privileges off.

4.6.26. Viewing Messages Sent to Logged Off Users (SET LX)

This command lets console and delegated console users view the screen messages sent to users who have logged off. The messages are displayed only at the console or terminal issuing the command. The messages sent to the logged off users are placed in the console log regardless of whether the SET LX command is issued.

Format

```
SETΔLX,{ON }
       {OFF}
```

Parameters

`ON`
Turns on message viewing function. (User issuing this command views messages directed to logged off users.)

`OFF`
Turns off message viewing function. (User issuing this command no longer receives messages directed to logged off users.)

Section 5
Interactive Services

You use 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 *Interactive Services Operating Guide* (UP-9972.) 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).

5.1. Interactive Services Commands

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 your command loads interactive services for terminal users. Interactive services commands are used to:

- Send messages to a workstation
- 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
- Display the log file
- Modify bulletin/log logon default setting and override values
- Load interactive services
- Terminate workstation and terminal user tasks or sessions

5.1.1. Sending Messages to Users (TELL)

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

Format

```
TELLΔ{ALL    },'text'
     {user-id}
```

Positional Parameter 1

ALL
: Displays the message on all active user screens.

user-id
: Specifies the one- to six-character alphanumeric identification of the user to receive the message display.

Positional Parameter 2

text
: Specifies the text of the message to be sent. The text may be a maximum of 88 characters long. Text must be preceded and followed by apostrophes.

5.1.2. Asking Questions of Users (ASK)

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 workstation screen.

Format

```
ASKΔuser-id,'text'
```

Positional Parameter 1

user-id
: Specifies the one- to six-character alphanumeric identification of the local or remote user to receive the message display.

Positional Parameter 2

text
: Specifies the text of the message to be sent. The text may be a maximum of 88 characters long. Text must be preceded and followed by apostrophes.

5.1.3. Displaying System Status (STATUS)

The STATUS command displays the volumes currently in use and the status of active workstations, terminals, jobs, and functions. It also displays the status of system resources in use and their availability and limitations.

Format

```
STATUSΔ [ ⎡JOBS              ⎤ ]
        [ ⎢FUNCTIONS         ⎥ ]
        [ ⎢RESOURCES         ⎥ ]
        [ ⎨TERMINALS         ⎬ ]
        [ ⎢TERMINALS[,uid]   ⎥ ]
        [ ⎢VOLUMES           ⎥ ]
        [ ⎣LIMITS            ⎦ ]
```

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.

TERMINALS,[uid]
: Displays a listing of all currently active (logged-on) terminals and associated user-ids that match the user-id you specify. The *uid* parameter must be between one and six alphanumeric characters.

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

LIMITS
: Displays the system resource management limits values and their current status.

Example 1

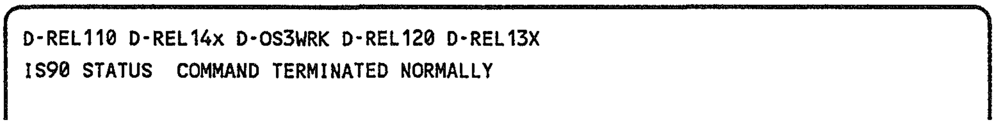
Operator keyin:

```
STA V
```

Function requested:

Display a listing of the tape, disk, and diskette volumes currently mounted on the system.

Screen display:



```
D-REL110 D-REL14x D-OS3WRK D-REL120 D-REL13X
IS90 STATUS  COMMAND TERMINATED NORMALLY
```

Disk volumes REL110, REL14X, OS3WRK, REL120, and REL13X are currently mounted on the system.

Example 2

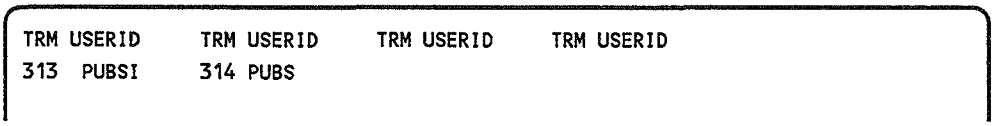
Operator keyin:

```
STA T
```

Function requested:

Display a listing of all the currently active workstations and terminals and their associated user-ids. In a secure system, display all terminals with at least one invalid log-on attempt and number of log-on attempts remaining for them.

Screen display:



```
TRM USERID     TRM USERID     TRM USERID     TRM USERID
313  PUBSI     314 PUBS
```

Workstations 313 and 314 are currently logged on under user-ids PUBSI and PUBS, respectively.

Additional screen display if security is active:

```
TRM             LOGON TRIES LEFT

  C12                  3
  (MAXLOGONS)          5
IS90 STATUS   COMMAND TERMINATED NORMALLY
```

Example 3

Operator keyin:

```
STA J
```

Function requested:

Display a listing of the jobs and symbionts currently active in the system.

Screen display:

```
JOB NAME   SIZE    CPU TIME  STEP   EXEC   JOB NO MASTER
RC$$IS00  006912       7.3    00   RC$$IS  00000
SL$$VT00  001536        .0    00   SL$$VT  00000  PUBS
SFGEN03   029696       2.9    01   SF$GEN  00003  PUBSI
I@DATA02  036864       4.8    01   DATA00  00002  PUBS
UNUSED MEMORY 0781312
IS90 STATUS   COMMAND TERMINATED NORMALLY
```

The jobs and symbionts currently active are listed in the first column. The SIZE column indicates the amount of storage each job is using. The CPU time is given next, followed by the job step number and the name of the program each job is executing. The JOB NO column is the job slot number of each job; MASTER provides the master user-id for the job or symbiont. The amount of unused memory is also shown.

Example 4

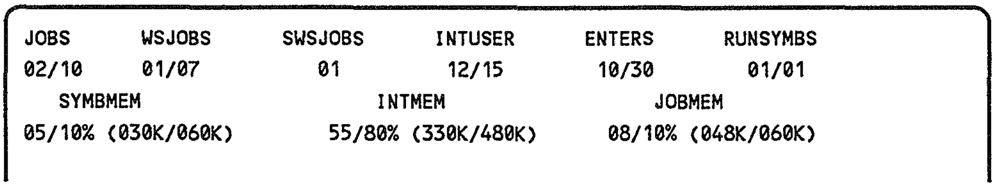
Operator keyin:

```
STA L
```

Function requested:

Display a listing of the current values and available limits for the resource management controls. These controls were set by the last LIMITS command or by resource management parameters specified during system generation.

Screen display:



```
JOBS       WSJOBS       SWSJOBS       INTUSER       ENTERS       RUNSYMBS
02/10      01/07          01           12/15         10/30          01/01
   SYMBMEM                        INTMEM                  JOBMEM
05/10% (030K/060K)          55/80% (330K/480K)       08/10% (048K/060K)
```

The values on this screen show that this user system will perform as follows:

`JOBS 02/10`
Can execute 10 jobs concurrently, but is now executing only 2.

`WSJOBS 01/07`
Can execute seven jobs concurrently from workstations only, but is now executing only one.

`SWSJOBS 01`
Can execute only one job from a single workstation at a time.

`INTUSER 12/15`
Can support up to 15 interactive users logged on at any one time, but only 12 users are logged on at this moment.

`ENTERS 10/30`
Can support 30 batch sessions executing concurrently, but is now supporting only 10.

`RUNSYMBS 01/01`
Can support one run symbiont executing at a time and is now supporting it.

`SYMBMEM 05/10% (030K/060K)`
Can use up to 10 percent (or 60K) of available main storage for symbiont use, but is now using only 5 percent (or 30K) for symbiont use.

`INTMEM 55/80% (330K/480K)`
Can use up to 80 percent (or 480K) of available main storage for interactive use, but is now using 55 percent (or 330K) for interactive use.

`JOBMEM 08/10% (048K/060K)`
Can use up to 10 percent (or 60K) of available main storage for batch use, but is now using only 8 percent (or 48K) for batch use.

Note: *If you specified NLMT for the SYMBMEM, INTMEM, or JOBMEM parameters, then the STATUS command listing would display NL as your upper limit for those three values.*

5.1.4. Displaying Log File (RECALL)

The RECALL command lets you display all or part of your log file at your workstation screen or at the system console. RECALL only uses the lines that are available on the screen for the recall.

Format

```
RECALLΔ⎧LASTΔnn          ⎫[,prefix]
       ⎩hh:mm:ss-hh:mm:ss⎭
```

Parameters

`LASTΔnn`
Indicates the number of messages (*nn*) in the log file that you wish to see.

`hh:mm:ss-hh:mm:ss`
Indicates that you wish to see the contents of the log file for a particular time period. All time must be specified in military time. The minutes (*mm*) and seconds (*ss*) are optional. You must, however, separate the beginning time from the ending time with a dash (-). To display all messages from a specific time until the end of the log file, specify only a beginning time. To display all messages from the beginning of the log file until a specific time, specify only an ending time. When keying in an ending time only, you must prefix it with a dash.

`prefix`
Indicates that you only wish to view those messages that begin with this prefix. The prefix can be from one to eight characters.

5.1.5. Modifying Bulletin and/or Log Values (SET IS)

The SET IS console command allows you to modify the bulletin/log logon default setting and override values during normal system processing.

Format

```
SETΔIS,⎡BULLDEF⎤,⎧YES⎫
       ⎢BULLOVR⎥ ⎩NO ⎭
       ⎢WLOGDEF⎥
       ⎣WLOGOVR⎦
```

Parameters

`BULLDEF` and `WLOGDEF`

Specify whether YES or NO appears as the default on the logon screen. These specifications are also valid for a user who logs on batch or does a smart logon (from system mode).

`BULLOVR` and `WLOGOVR`

Specify whether the interactive services user logging on can change the values that appear as defaults. These specifications also apply to batch and smart logons. If NO is specified and the user attempts to change the value, the following message appears:

```
IS136  USER NOT ALLOWED TO CHANGE BULLETIN AND/OR LOG VALUES
```

The user will remain logged on providing there are no other logon errors.

5.1.6. Starting Interactive Services (IS)

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.

- Enter the IS REMOTE unsolicited message (see 5.2.3) to start interactive services when ICAM and GUST are ready and the interactive services facility is active (already loaded) before being needed for terminals.

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 (5.2.4), 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

```
ISΔ[REMOTEΔSTART]
```

Positional Parameter

`REMOTE START`

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

5.1.7. Terminating User Tasks or Sessions (REMOVE)

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

Format

```
REMOVEΔ⎡task-id⎤
       ⎨user-id⎬
       ⎣ALL    ⎦
```

where:

`task-id`

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

`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 workstation or console screen to indicate when the user cannot be logged off.

`ALL`

Specifies all user sessions and batch runs are to terminate for each user-id that can be controlled by this user. All interactive functions 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 workstation or console screen.

5.2. Interactive Services Messages

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

- Restrict and release new workstation and terminal user sessions
- Control interactive services for terminals
- Terminate interactive services

5.2.1. Restricting New User Sessions (CLOSE Message)

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

Format

```
00ΔISΔCLOSE
```

5.2.2. Releasing New User Session Restrictions (OPEN Message)

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

Format

```
00ΔISΔOPEN
```

5.2.3. Controlling Interactive Services for Terminals (REMOTE Message)

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 10 for a description of the ICAM and GUST operator procedures.)

Format

```
00ΔISΔREMOTEΔ⎧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 workstation or 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.

You use 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 *Interactive Services Operating Guide*, UP-9972. These commands and messages enable you to exercise control over the interactive OS/3 operating system, all jobs within the system, and all terminal users.

5.2.4. Terminating Interactive Services (SHUTDOWN Message)

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 (5.1.6); it is not automatically loaded for workstation users via an interactive services command entry.

Format

```
00ΔISΔSHUTDOWN▶
```

Section 6
Integrated Communications Access Method (ICAM) Procedures

The integrated communications access method (ICAM) is an extension of the supervisor (a symbiont) that handles data communications tasks. Each symbiont can contain multiple network definitions or communications control areas (CCAs); each CCA can handle one or more communications lines.

At system generation time, the ICAM symbionts are tailored to each user's requirements. One or more ICAM symbionts can be configured during SYSGEN, each symbiont satisfying specific communications network requirements; or a single ICAM symbiont can be configured to satisfy all communications network requirements.

You 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, you must initiate the running of the global user service task (GUST) after loading ICAM.

The ICAM symbiont remains in main storage until GUST is shut down. Then ICAM shuts itself down unless you loaded ICAM with a KEEP operand. In this 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.

6.1. ICAM Operator Commands

6.1.1. Loading the ICAM Symbiont (Cn/Mn)

Key in the following Cn/Mn command from the console to load the ICAM symbiont. (ICAM symbiont names are established during the COMMCT phase of system generation.) Always load ICAM in an idle state to avoid main storage fragmentation.

Format

```
⎧Cn⎫Δ[KEEP]
⎩Mn⎭
```

where:

```
⎧Cn⎫
⎩Mn⎭
```

Is the ICAM synboint name as specified in the MCPNAME parameter defined during the COMMCT phase of system generation (n can be a value of 1 to 9).

`KEEP`

Keeps the ICAM symbiont loaded until you cancel it or ICAM suffers an unrecoverable error.

ICAM loading is complete when the following output message is displayed on the console workstation or console screen:

```
ICAM xx.x READY
```

where:

xx.x specifies the ICAM release level; for example, 14.0

Note: *Users can also load ICAM from a job control stream with the / / CC job control statement. This statement allows users to load ICAM as part of your information management system (IMS) or transaction information processing (TIP) execution job, or as part of a job that executes the global user service task (including the job executed as part of the automatic boot procedure).*

6.1.2. Changing the ICAM Name (SET IC)

Key in the following SET IC command to change the name of the ICAM symbiont that is currently loaded. Use this procedure when the remote batch output has output ready and the required ICAM symboint is not loaded.

Format

```
SETΔIC,⎧Cn⎫
       ⎨Mn⎬
       ⎩C?⎭
```

where:

`IC`

Specifies that you want to change the SPOOLICAM SYSGEN parameter.

```
⎧Cn⎫
⎨Mn⎬
⎩C?⎭
```

Specifies the ICAM symbiont name (C1-C9 or M1-M9) as defined at system generation. Specifying C? indicates that you want to be queried (requested) to supply the symboint name the next time output is ready and the required ICAM symboint is not loaded.

6.2. Initializing and Terminating the Global User Service Task (GUST)

You can initialize GUST after ICAM is loaded and before starting interactive services or executing user programs requiring global networks. Initialize GUST by running the job that executes the GUST program ML$$GI. (Ask your system administrator for the name of the job you must run to execute the ML$$GI program.) When you no longer require global network processing, enter an unsolicited message to shut GUST down.

6.2.1. Running the Global User Service Task Job

Enter the following RUN command from the system console to initiate ML$$GI.

Format

```
RUNΔjobname
```

where:

jobname
Names the prefiled job stream that executes the GUST ML$$GI program.

6.2.2. ML$$GI Program Operator Messages

When ML$$GI executes, it displays messages at the console workstation or console requesting you to provide information needed to initialize the global network. Respond to each message as indicated. (Only those messages pertaining to GUST initialization are presented here. See the *System Messages Reference Manual*, 7004 5190 for all GUST messages.)

For the message:

```
MC#420 ENTER NETREQ: CCA, PASS, RESTART, LINEREQ
```

GUST request you to enter the cca-name, password, restart, and line information associated with the global network to be initialized. When multinode global networks are used, ML$$GI automatically requests all virtual lines. Use the following format when keying your response:

Format

```
cca-nameΔ[[,password][,restart][,line-request]]
```

where:

cca-name
Names 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 one- to eight-character password.

`restart`
Determines the type of restart GUST requests for the network. Key in Y for warm starts (messages are recovered from existing disk files). Key in N for normal reinitiatization. That is, new disk files are created if required.

`line-request`
Specifies the lines for which line requests are issued. Response may be one of the following:

`ALL,Y`
Specifies that line requests are issued for all lines in the network. Any errors are treated as non-fatal and the network is brought up.

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

`*`
Displays the MC#421 message asking you to enter line request information. (See message MC#421 which follows.)

For the message:

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

Enter the lines for which line requests are 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`
Names the lines to be activated by means of line requests. GUST issues a LNEREQ statement for each line-name you specify. The line-name must be identical to that specified in the label field of the related LINE macroinstruction.

`blank`
Signals the GUST program to issue no specified line requests at this time. (VLINES and RWS lines are automatically requested during GUST processing.)

For the message:

```
MC#430 GUST ACTIVE FOR CCA nnnn
```

No response is required. This message informs you that the network you requested is loaded and ready for processing; nnnn is the name of the global network loaded. Message MC#25 or MC#20 will appear shortly after the MC#430 message.

For the message:

```
MC#25 nnnn ch/cu dddd CHANNEL INITIALIZED
```

When this message appears after the MC#430 message, it indicates that ICAM successfully communicated with the DCP. No response is required. Specified in the message are the name of the global network (nnnn), the DCP's channel/control unit addresses (ch/cu), and the label on the DCPCHNL macroinstruction (dddd).

Message

```
MC#20 nnnn ch/cu dddd CHANNEL DOWN ERROR ee
```

When this message appears after the MC#430 message, it indicates that ICAM could not establish communications with the DCP. In addition to the global network name (nnnn), DCP channel/control unit addresses (ch/cu), and DCPCHNL macroinstruction label, the message includes a 2-byte decimal error code (ee). Look up this code in the *Systems Message Reference Manual*, 7004 5190, for its explanation and recommended action to be taken.

6.2.3. Terminating the Global User Service Task

Key in the following unsolicited message to terminate the GUST job and end global network processing.

Format

```
00Δ{Cn}ΔGUΔS,network-name
   {Mn}
```

where:

`00`
: Specifies this message is destined for the supervisor.

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

Identifies 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 one- to four-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 shutdown request is accepted, the following message is displayed when the global user service task job is canceled:

```
MC#401 GUST SHUTDOWN COMPLETE
```

Note: *To resume communications processing with global networks or to start interactive services for terminals, you must repeat the procedures for loading ICAM and initializing GUST. Never use the CANCEL command to cancel the GUST job. If you do, the CANCEL command and the requested GUST shutdown command are rejected and the following message is displayed:*

```
MC#402 CANCEL REJECTED, ISSUE SHUTDOWN COMMAND
```

6.3. ICAM Operator Messages

Occasionally, the you must communicate with ICAM to change the communications environment or to advise ICAM of some external event. You use unsolicited messages to do this. The descriptions of the ICAM console messages are divided into three types:

- Standard messages - those not related to public data networks or remote batch processing
- PDN unsolicited messages - those related to public data networks
- RBP unsolicited messages - those having to do with remote batch processing

6.3.1. Standard Unsolicited Messages

Standard unsolicited messages to ICAM have the following format:

```
00Δ{Cn}ΔccΔf,{xxxx},nnnn
   {Mn}      {ccpp}
```

where:

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

Specifies the name of the required ICAM symbiont (C1-C9 or M1-M9).

cc

Is a two- to four-character command code.

f

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

xxxx

Is a one- to four-character name of a line or terminal as defined in the label field of a LINE or TERM macroinstruction.

ccpp

Specifies the input/output microprocessor (IOMP) channel number (*cc*) and the single line communications adapter id (*pp*) on the specified channel. Select your parameters from the following list:

Number of IOMPs	Channel No. (cc)	SLCA No. (pp)
1 IOMP	13	01 to 14
2 IOMPs	13 or 15	01 to 14

nnnn

Is the one- to four-character name of the network as specified in the label field of the CCA macroinstruction.

Table 6-1 lists the standard unsolicited messages ICAM provides for you. A typical ICAM response to an unsolicited message is:

```
NET1 TERM TRM1 MARKED UP
```

This response tells you that terminal TRM1 on network NET1 is now available.

Table 6-1. Standard ICAM Unsolicited Messages

Message	Description
00Δ{Cn}ΔUPΔL,xxxx,nnnn {Mn}	Marks line specified as available (up). Activates a communications line. Equivalent to a line request issued by a program. ICAM can poll, send output, and prepare to receive input. No traffic can flow over a nonactivated communications line.
00Δ{Cn}ΔDOΔL,xxxx,nnnn {Mn}	Marks line specified as unavailable (down). Stops all activity on a communications line. ICAM does not poll, send, or receive traffic. A switched line is disconnected. ICAM does not answer calls or dial out.
00Δ{Cn}ΔUPΔT,xxxx,nnnn {Mn}	Marks terminal (UNISCOPE or DCP) or remote workstation specified as available (UP). Enables ICAM to send output to a terminal previously marked down.
00Δ{Cn}ΔDOΔT,xxxx,nnnn {Mn}	Marks terminal (UNISCOPE or DCP) or remote workstation specified as unavailable (down). Stops ICAM from sending output to the terminal specified and aborts any current session at the terminal. If the last terminal in a poll group (UNISCOPE or remote workstation) is marked down, the entire poll group is placed on slow poll (the group is polled about once every minute). If input is entered at one of these terminals, a response is delayed until the next time the terminal is polled and the terminal marked up. When a remote workstation is marked down, its session as well as its partner's session is aborted. (Partners are defined by the primary/secondary specifications on adjacent TERM macroinstructions.)

continued

Table 6-1. Standard ICAM Unsolicited Messages (cont.)

Message	Description
00Δ{Cn}ΔUPΔP,ccpp,nnnn {Mn}	Marks port specified as available (UP). Useful when an ICAM symbiont contains more than one network and communications lines are shared between them. To reassign a line: - Down the port where the line is connected (specify the network currently using the line). - Using this command (UP port), reactive the port. (Specify the network to receive the line). - Issue an UP command to active the communications line. (Specify the network now using the line.)
00Δ{Cn}ΔDOΔP,ccpp,nnnn {Mn}	Marks port specified as unavailable (down). See the preceding message description.
00Δ{Cn}ΔCNΔL,xxxx,nnnn {Mn}	Notifies ICAM that dialing is completed on switched line specified. This is a response to a dialing message from ICAM to dial a manually dialed line. ICAM does not pass traffic on the switched line until it receives a response.
00Δ{Cn}ΔCNΔL,ALL,nnnn {Mn}	Notifies ICAM that dialing is completed on all switched lines. This is a response to one or more ICAM messages to manually dial switched lines.
00Δ{Cn}ΔSTΔL,llll,cccc {Mn}	Displays status of a communications line (llll) in a CCA (cccc).
00Δ{Cn}ΔSTΔL,*ALL,cccc {Mn}	Displays status of all communications lines in a CCA (cccc).
00Δ{Cn}ΔSTΔT,tttt,cccc {Mn}	Displays status of a terminal/workstation (tttt) within a CCA (cccc).
00Δ{Cn}ΔSTΔU,uuuu,cccc {Mn}	Displays status of a LOCAP (uuuu) within a CCA (cccc).
00Δ{Cn}ΔUUΔL,llll,cca- {Mn} name	Brings a switched line type up as an unattended line.
00Δ{Cn}ΔUMΔL,llll,cca- {Mn} name	Brings a switched line type up as a manual dial line.

6.3.2. Public Data Network (PDN) Unsolicited Messages

There are two kinds of public data networks: packet-switched and circuit-switched. The formats for entering unsolicited messages to ICAM vary with the type of public data network involved. The following paragraphs describe these formats. (See the *ICAM Operations Guide*, 7004 4557, for information about public data networks.)

Note: *In addition to the commands described in the following paragraphs, ICAM lines connected to public data networks use the standard ICAM UP and DOWN commands to activate and deactivate lines, terminals, or ports.*

How to Enter Packet-Switched PDN Unsolicited Messages

Packet-switched public data network unsolicited messages have the following format:

```
00Δ{Cn}ΔccccΔnnnn,llll[,dddd]
   {Mn}
```

where:

`00`
: Specifies this message is destined for the supervisor.

`Cn` or `Mn`
: Specifies the name of the ICAM symbiont to which this type-in is destined (C1-C9 or M1-M9).

`cccc`
: Specifies one of the following four-character command codes: STAT, CONN, DISC, or TEST.

  where:

  `STAT`
  : Causes the status of the level 2 link, the virtual circuits, or the subscribers to be displayed on the operator console. The llll parameter specifies the type of status requested.

  `CONN`
  : Reestablishes a connection to the level 2 link after it is disconnected.

  `DISC`
  : Drops (disconnects) a connection to the level 2 link.

  `TEST`
  : Initiates (for DDX only) the level 2 self-testing procedures.

`nnnn`
: Specifies the one- to four-character network name as specified in the label field of the CCA macroinstruction.

llll
: Specifies (for CONN, DISC, and TEST) the one- to four-character label of the VLINE macroinstruction defining the link. For the STAT command code, this parameter indicates the type of status to be displayed:

- Enter link name llll as instructed for the other command codes for the status of the link.
- Enter VCT to display the status of the virtual circuit.
- Enter SUBS to display the status of the subscriber.

dddd
: Is the one- to four-character label of the PDN macroinstruction that identifies the packet-switched public data network affected by this key-in. This parameter is optional and, if omitted, defaults to the label of the first PDN macroinstruction in the network definition.

How to Enter Circuit-Switched PDN Unsolicited Messages

The following formats are available for entering circuit-switched public data network unsolicited commands to ICAM.

In general, when the PDN keyword is omitted, the default is the label on the first PDN macroinstruction in the ICAM network definition. When the LDTE keyword is omitted, the default is the label on the first LDTE macroinstruction in the ICAM network definition.

Format 1

```
00Δ{Cn}ΔSTATΔnnn,{LDTE=ldte-name}[,PDN=pdn-name]
   {Mn}          {RDTE=rdte-name}
                 {TERM=term-name}
                 {CPS={cc }     }
                 {    {ALL}     }
```

Displays the status of the specified LDTE, RDTE, terminal (TERM), or call progress signal (CPS), where:

{Cn}
{Mn}
: Specifies the name of the ICAM symbiont for which this command is destined (C1-C9 or M1-M9).

nnnn
: Specifies the one- to four-character network name as specified in the label field of the CCA macroinstruction.

ldte-name
: Is the label on an LDTE macroinstruction in the ICAM network definition.

`rdte-name`
Is the label on an RDTE macroinstruction in the ICAM network definition.

`term-name`
Is the label on a TERM macroinstruction in the ICAM network definition.

`cc`
Specifies a valid call progress signal number.

`pdn-name`
Is the label on a PDN macroinstruction in the ICAM network definition.

Format 2

```
00Δ{Cn}ΔNETSΔnnn,{LDTE=ldte-name}[,PDN=pdn-name]
   {Mn}          {RDTE=rdte-name}
                 {CPS={cc }     }
                 {    {ALL}     }
```

Displays the network statistics for the specified LDTE, RDTE, or call progress signal (CPS).

Format 3

```
00Δ{Cn}ΔCONNΔnnn,RDTE=rdte-name[,LDTE=ldte-name][,PDN=pdn-name]
   {Mn}
```

Establishes a connection to the specified RDTE.

Format 4

```
00Δ{Cn}ΔDISCΔnnn,RDTE=rdte-name[,LDTE=ldte-name][,PDN=pdn-name]
   {Mn}
```

Disconnects from the specified RDTE.

Format 5

```
00Δ{Cn}ΔCHNGΔnnn,RDTE=rdte-name{,ADDR=address}[LDTE=ldte-name][,PDN=pdn-name]
   {Mn}                        { DSCO=t      }
                               { DSCI=t      }
                               { TIME=t      }
```

For the specified RDTE, changes:

a. Network address (ADDR=address) (maximum 16 digits)

b. For terminal connections

   - Time before disconnect following output (DSCO) (range 0-32767)

   - Time before disconnect following input (DSCI) (range 0-32767)

c. For host to host connections, the activity timer (TIME) (range 0-32767)

Format 6

```
00Δ⎧Cn⎫ΔCHNGΔnnn,CPS=cc⎧,NIR=n⎫[LDTE=ldte-name][,PDN=pdn-name]
   ⎩Mn⎭                 ⎪ TIR=n⎪
                        ⎨ NSR=n⎬
                        ⎪ TSR=n⎪
                        ⎩ TBD=n⎭
```

For the specified call progress signal (CPS), changes:

a. ICAM retry parameters where

NIR = number of ICAM retries (range 0-32767)

TIR = time between ICAM retries (range 0-32767)

b. SLCA retry parameters where

NSR = number of SLCA retries (range 0-63)

TSR = time between SLCA retries (range 0-255)

c. Time before disconnect (TBD) (range 0-254)

For further descriptions of these parameters and their usage, refer to the *ICAM Operations Guide*, 7004 4557.

6.3.3. Remote Batch Processing (RBP) Unsolicited Messages

You communicate with the remote batch processing ICAM symbiont by means of console commands in the form of unsolicited messages. These commands enable you to specify the functions you want performed. The format of a remote batch processing unsolicited message is:

```
00Δ⎧Cn⎫Δcc[,D]
   ⎩Mn⎭
```

where:

```
⎧Cn⎫
⎩Mn⎭
```

Specifies the two-character remote batch processing ICAM symbiont identifier (C1-C9) or (M1-M9).

cc

Specifies two-character command code.

D

Is an optional dump operand associated with the SH command.

The following command codes are accepted by remote batch processing. They are explained in detail in the *ICAM Utilities Programming Guide*, 7004 4565.

- SH

  Specifies shutdown. It removes the ICAM symbiont from the system when remote batch processing is the only active ICAM interface or idles remote batch processing when other ICAM interfaces are active in the same ICAM symbiont. If you specify the D option on the SH command, your system produces a dump during termination of the ICAM symbiont.
- RB

  Reactivates remote batch processing after it has been idled by an SH command.

- RD

  Starts the reading of batch jobs or data files from the central processor card reader under control of remote batch processing.

6.4. Downline Loading a Telcon File

The two procedures that follow let you downline load a Telcon file to a DCP front-end processor or a DCP remote concentrator. Additional detailed information concerning this subject is presented in the *Integrated Communications Access Method (ICAM) Utilities Programming Guide,* 7004 4565.

6.4.1. Loading Telcon File to Front-End Processor (DCP)

Perform the following sequence to downline load a Telcon file to a DCP front-end processor:

1. Load the ICAM symbiont.

2. Run GUST.

3. Initialize the line over which the DCP load sequence is to occur. (GUST automatically does this unless the line was previously marked down, in which case you must enter an unsolicited command to reinitialize the line.)

4. At the DCP front-end processor, initiate a Telcon load sequence. Refer to the *DCP / OS Operations Reference Manual,* 7831 5702, for information on initiating downline loading from a DCP.

Note: *To avoid reinitializing the line, it is recommended that you start the downline load procedure at the DCP before GUST is executed.*

6.4.2. Loading Telcon File to Remote Concentrator (DCP)

Perform the following sequence to downline load a Telcon file to a DCP remote concentrator:

1. Load the ICAM symbiont.

2. Run GUST.

3. Initialize the line over which the DCP load sequence is to occur. (GUST automatically does this unless the line was previously marked down, in which case you must enter an unsolicited command to reinitialize the line.)

4. Start the TELRCLD job stream by entering:

   ```
   RV TELRCLD
   ```

   The following message is displayed:

   ```
   ENTER NETWORK NAME, PASSWORD, LOCAP NAME
   ```

   Respond to this message with the network (CCA) name, the password (if any), and the word LOAD (the LOCAP name).

5. At the DCP remote concentrator, initiate a Telcon load sequence. Refer to the *DCP/OS Operations Reference Manual,* 7831 5702, for information on initiating a downline load from a DCP.

6. The TELRCLD job now displays the following messages to indicate the status of the load operation:

   ```
   RCLD Established a Session
   Remote Load of DCPID 156 Started   Load ELT PRCnnn
   Remote Load of DCPID 156 Complete  Load ELT PRCnnn
   ```

7. The TELRCLD job remains in memory after the load is completed. If desired, you can use it to load another remote concentrator. To terminate the TELRCLD job, enter:

   ```
   UNS TELRCLD TERM
   ```

Section 7
System Utility Services

The system utility symbiont (SL$$SU) is a multipurpose utility that allows you to perform many card, tape, disk, and diskette functions at the console workstation or console. To initiate a function, you include its function code as a parameter with the SU/TU command or enter it as a solicited message. The functions you can perform and their function codes are listed in Table 7-1. The SU/TU command and the functions for card, tape, disk, and diskete are described in 7.1 through 7.5.

Table 7-1. System Utility Functions

Function Code	Function Performed
	Card Functions
CC	Reproducing cards punched in Hollerith code
CCB	Reproducing cards punched in binary and Hollerith code
CCS	Reproducing and resequencing source programs
CT	Writing card to tape in unblocked format
CTR	Writing card to tape in blocked format
CP	Listing cards
CH	Listing cards containing compressed mode
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
TPR	Printing a tape in character and deblocked format
TRS	Locating a specific record 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 and Format Label Diskette Functions
DD	Printing a disk in unblocked format
DDR	Printing a disk in deblocked format
VTP	Printing the volume talb of contents of disk
SVT	Printing short format VTOC file
AVX	Displaying available disk extents on console screen
	Data Set Label Diskette Functions
DD	Printing a diskette in unblocked format
DDR	Printing a diskette in deblocked format
VTP	Printing the data set labels of a diskette

NOTE: If XXX is entered in place of the function code, all function codes are displayed.

7.1. System Utility Commands and Messages (SU/TU)

The SU/TU command loads the system utility symbiont. SU and TU can be used interchangeably for all functions. However, we recommend that you use TU for tape operations because it increases the buffer size for tapes from 8,189 to 32,767 bytes. You can include the required function as a parameter with the SU/TU command.

A spooling parameter also can be entered with the command, if spooling is configured in your system. (For the description of all spooling commands, see the *Spooling and Job Accounting Operating Guide*, 7004 4581.)

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 by responding with solicited messages. You use unsolicited messages only to terminate the symbiont or current symbiont functions on certain occasions.

Note: *The SU/TU command cannot be issued from an enter stream.*

Format

```
        ┌                      ┐
        │               ┌   ┐  │
┌SU┐Δ   │function-code ,│┌Y┐│  │
└TU┘    │XXX            ││N││  │
        │               ││R││  │
        │               │└H┘│  │
        │               └   ┘  │
        └                      ┘
```

Positional Parameter 1

`function-code`

Specifies the appropriate two- or three-character function code. (Function codes are listed in Table 7-1.) If omitted, the symbiont displays a message requesting that you enter a function code. All possible function codes that are recognized by either symbiont can be displayed on the console or console workstation 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 that the system utility output is not spooled.

`R`

Retains system utility output in the 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 the 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 any spooled output at the end of every SU function.

Operator Considerations

- When the symbiont is loaded, the following message is displayed:

```
nnΔSYSTEM UTILITY SYMBIONT LOADED                             SUNNNN
```

  where:

  nn
  : Is the two-digit message number

  SUNNNN
  : Is the name of the symbiont sending the message. This name is the two-character symbiont identification plus a four-digit binary job number inserted at task initiation.

- 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:

  a. With spooling:

```
nn?ENTER REQUIRED FUNCTION AND SPOOL OPTION [,Y,N,R,H];DEFAULT=Y   SUNNNN
```

  b. Without spooling:

```
nn?ENTER REQUIRED FUNCTION                                         SUNNNN
```

  Initiate the required function with the keyin:

  a. With spooling:

```
nnΔfunction-code,⎡Y⎤
                 ⎨N⎬
                 ⎨R⎬
                 ⎣H⎦
```

b. Without spooling:

```
nnΔfunction-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:

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

Enter the required spooling option with the keyin:

```
nnΔ⎧Y⎫
   ⎨N⎬
   ⎪R⎪
   ⎩H⎭
```

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

```
nnΔEOJ
```

The symbiont terminates and displays the message:

```
nnΔSYSTEM UTILITY SYMBIONT ENDED                                 SUNNNN
```

- You can also end the symbiont by pressing the MSG WAIT key on the console workstation or console and keying in the applicable unsolicited message:

```
00ΔSUΔEOJ
00ΔTUΔEOJ
```

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

- To terminate only the current function of the symbiont, key in the applicable unsolicited message:

```
00ΔSUΔEND
00ΔTUΔEND
```

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 previous unsolicited message to terminate the current function or to terminate the symbiont.*

7.2. 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 7.1:

   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 console workstation or console:

```
nnΔUSE READER did                                              SUNNNN
```

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 console workstation or console:

```
nnΔNO READER AVAILABLE                                   SUNNNN
```

Likewise, if the required output device is not available, the function aborts and the following message is displayed on the console workstation or console:

```
nnΔNO⎧PUNCH  ⎫AVAILABLE                                  SUNNNN
     ⎨TAPE   ⎬
     ⎩PRINTER⎭
```

If the required devices are available, the operation continues for each function code as described in 7.2.1 through 7.2.8.

7.2.1. 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.

7.2.2. 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.

7.2.3. 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 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

7.2.4. 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:

```
nn?CUUMMB OUTPUT TAPE B=BLK CNT                                  SUNNNN
```

You respond with the tape unit device address, mode setting, and block count characteristics of the output tape (as described in 7.3.1).

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:

```
nn?MOUNT NEXT OUTPUT TAPE. REPLY R OR END                        SUNNNN
```

Key in the following if the CT function is to be terminated:

```
nnΔEND
```

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

Key in:

```
nnΔR
```

For the message:

```
nn?DOES USER WISH TO PREP THIS TAPE? REPLY YES OR NO                  SUNNNN
```

Key in:

```
nnΔN
```

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

```
nnΔY
```

To prep the tape. The SU INT tape function is activated (see "Prepping a Tape (INT)" in 7.3.3). The CT function resumes when INT function is completed.

7.2.5. 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:

```
nn?BLK FACTOR                                                         SUNNNN
```

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

For the message:

```
nn?CUUMM OUTPUT TAPE B=BLK CNT                                        SUNNNN
```

You respond with the tape unit device address, and the mode setting of the output tape (see 7.3.1).

For the message:

```
nn?OUTPUT EXCEEDS ALLOCATED BUFFER                                  SUNNNN
```

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 "Tape Addressing" in 7.3.3.

7.2.6. 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.

7.2.7. 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.

7.2.8. Punching Cards from the System Console (JCP)

The JCP function punches job control or data cards entered through the console workstation or 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.

7.3. Tape Functions

Use the SU command if your tape block size does not exceed 8189 bytes; TU supports tape blocks up to 32,767 bytes.

7.3.1. Tape Addressing

The tapes to be used for a TU function must be identified using the tape unit ID. Communication by you is initiated by the following message:

```
cuummb
```

where:

cuu

Is device ID (channel and unit).

mm

Is mode setting of the tape. If *mm* is blank, 00, or not entered (e.g., response is 102), the SYSGEN mode settings are assumed. Commonly used mode settings are:

C0 for 1600 bpi (PE)
C8 for 800 bpi (NRZ)
D0 for 6250 bpi (GCR)
E9 for slow-speed operation (50 ips)
F9 for fast-speed operations (100 ips)

Refer to the *System 80 OS/3 Job Control Programming Guide*, 7004 4623, for information on mode settings.

b

Is block count characteristics of the tape. 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 block number (TBLK) printed for the TRS function is a display of the 3-byte block number prefixed to block-numbered tapes.*

7.3.2. 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 and the tape is forward spaced to the next block. If an error is bypassed, no information from the bad block is processed and the tape is forward spaced to the next block.

The messages:

```
nnΔTAPE ERROR ON INPUT TAPE                                   SUNNNN
nn?B-BYPASS, I-IGNORE, OR E-END FUNCTION                      SUNNNN
```

indicates a tape error has occurred and one of the following options may be replied:

```
nnΔBYPASS     (This reply is not applicable during a TRL function.)
nnΔIGNORE
nnΔEND
```

The message:

```
nnΔWARNING: TAPE BLK FILLS AVAILABLE BUFFER OR MAY BE TRUNCATED      SUNNNN
```

warns that a tape input or output record completely fills the allocated buffer. SU reads/writes a maximum of 8189 bytes and TU reads/writes a maximum of 32,767 bytes; any tape block exceeding this is truncated.

7.3.3. Tape Operating Instructions

To perform a tape function, proceed as follows:

1. As described in 7.2:

    a. Enter the TU symbiont command (or enter SU if the tape block size does not exceed 8189 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 this subsection.

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:

```
nn?CUUMMB-INPUT TAPE B=BLK CNT                                SUNNNN
```

Respond with the tape unit device address, and the mode setting and block count characteristics of the input tape (7.3.1).

```
nn?CUUMMB-OUTPUT TAPE B=BLK CNT                               SUNNNN
```

Respond with the tape unit device address, and the mode setting and block count characteristics of the output tape.

```
nn?#FILES 1-99                                                SUNNNN
```

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 for the number of files on a standard labeled tape.

If a high density tape is copied 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
- Rewinds and unloads the tape

For the message:

```
nn?MOUNT NEXT OUTPUT TAPE. REPLY R OR END                     SUNNNN
```

Key in:

```
nnΔEND
```

If the TT function is to be terminated.

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

Key in:

```
nnΔR
```

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:

```
nnΔSUPPLY VOLUME SERIAL NUMBER FOR NEW OUTPUT TAPE                SUNNNN
nn?XXXXXX                                                         SUNNNN
```

Key in:

```
nnΔvsn
```

where:

vsn
: Is the six-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.
```

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:

```
nn?END OF VOLUME? Y OR N                                          SUNNNN
```

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.

An example of tape copy with block count is as follows:

System message:

```
nn?CUUMMB - INPUT TAPE                                            SUNNNN
```

Operator response:

```
nnΔ100ΔΔB
```

System message:

```
nn?CUUMMB - OUTPUT TAPE                                       SUNNNN
```

Operator response:

```
nnΔ101ΔΔB
```

System message:

```
nn?#FILES 1-99                                                SUNNNN
```

Operator response:

```
nnΔ3
```

System message:

```
nn?END OF VOLUME? Y/N                                         SUNNNN
```

Operator response:

```
nnΔY
```

Printing a Tape in 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:

```
nn?CUUMMB - INPUT TAPE B=BLK CNT                              SUNNNN
```

Respond by entering the tape unit device address and the mode setting and block count characteristics of the input tape (see 7.3.1). 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.

```
nn?#BLKS OR END                                               SUNNNN
```

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.

Printing a Tape in 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.

```
nn?CUUMMB - INPUT TAPE B=BLK CNT                                      SUNNNN
```

Enter the tape unit device address, and the mode setting and block count characteristic of the input tape (see 7.3.1). If you specify block count, data is considered to begin in position 3, relative to position 0, for a length of blocksize minus 3 bytes.

If you specify B (for block count), the following message is displayed:

```
nn?#BLKS OR END                                                       SUNNNN
```

If the entire tape is to be printed, enter END. To print a specified number of blocks, enter a decimal number indicating the number of blocks to be printed. If the end of file (two tapemarks side by side) is detected before the specified number of blocks are read, the function is terminated.

```
nn?LOGICAL REC LNTH                                                   SUNNNN
```

Respond with the logical record length.

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.

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.

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
- Actual data needed

Procedure

You identify the scan field with the 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 (see 7.3.2). 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:

```
nnΔINVALID HEX CHAR                                                SUNNNN
```

An error message is printed when the scan argument is entered in hexadecimal and an invalid hexadecimal character was detected. You reenter the sequence to obtain the scan argument.

```
nnΔNO MATCH FND                                                    SUNNNN
```

This message is printed when an end-of-file mark (two tape marks back to back) is encountered before any finds were made. The function is terminated.

```
nnΔCONTINUE SCAN? Y/N                                              SUNNNN
```

This message is printed 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.

An example of a tape with block count that is to be searched for a tape record containing a key field of "4637275467" in the first 10 bytes of the record is as follows:

System message:

```
nn?CUUMMB - INPUT TAPE                                        SUNNNN
```

Operator response:

```
nnΔ10000B
```

System message:

```
nn?LOGICAL REC LENGTH                                         SUNNNN
```

Operator response:

```
nnΔ125
```

System message:

```
nn?LENGTH ARGUMENT (1-30|0/)                                  SUNNNN
```

Operator response:

```
nnΔ10
```

System message:

```
nn?STARTING DATA POSITION IN REC                              SUNNNN
```

Operator response:

```
nnΔ1
```

System message:

```
nnΔENTER IN HEX-H, CHAR-C                                  SUNNNN
```

Operator response:

```
nnΔC
```

System messages:

```
nnΔENTER 10 BYTES, 1 CHAR PER BYTE                         SUNNNN
nn?1...5...10...                                           SUNNNN
```

Operator response:

```
nnΔ4637275467
```

System message:

```
nn?CONTINUE SCAN? Y/N                                      SUNNNN
```

Operator response:

```
nnΔN
```

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

When a tape mark is encountered, a card with all * is punched. The function is terminated when two tape marks back to back are encountered. Tape error processing is described in 7.3.2.

For the message:

```
nn?CUUMMB - INPUT TAPE B=BLK CNT                           SUNNNN
```

Respond with the tape unit device address, and the mode setting and block characteristics of the input tape (as described in 7.3.1).

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 tape unit physical unit block (pub). It is, therefore, possible to prep multiple tapes with the same vsn.

Procedure

For the messages:

```
nn?CUUMMB - OUTPUT TAPE B=BLK CNT                          SUNNNN
```

Respond with the tape unit device address and the mode setting and block count characteristic of the output tape (refer to 7.3.1).

```
nnΔENTER NEW FILE LABEL                                    SUNNNN
nn?XXXXXXXXXXXXXXXXX                                       SUNNNN
```

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.

```
nnΔENTER OWNER I.D. OR SCRATCH                             SUNNNN
nn?XXXXXXXXXXXXXXXXX                                       SUNNNN
```

Enter up to 17 characters for a new owner identifier.

Example

System message:

```
nn?CUUMMB - OUTPUT TAPE                                    SUNNNN
```

Operator response:

```
nnΔ10100B
```

System messages:

```
nnΔENTER NEW VOL#                                                SUNNNN
nn?XXXXXX                                                        SUNNNN
```

Operator response:

```
nnΔ001036
```

System messages:

```
nnΔENTER NEW FILE LABEL                                          SUNNNN
nn?XXXXXXXXXXXXXXXXX                                             SUNNNN
```

Operator response:

```
nnΔTAPE1
```

System messages:

```
nnΔENTER OWNER I.D. OR SCRATCH                                   SUNNNN
nn?XXXXXXXXXXXXXXXXX                                             SUNNNN
```

Operator response:

```
nnΔMr. Smith
```

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:

```
nn?CUUMM - OUTPUT TAPE                                           SUNNNN
```

Enter the tape unit device address and the mode setting of the tape to be spaced forward (refer to 7.3.1).

Backspace 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:

```
nn?CUUMM - OUTPUT TAPE                                        SUNNNN
```

Enter the tape unit device address and the mode setting of the tape to be backspaced (refer to 7.3.1).

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 message:

```
nn?CUUMM - OUTPUT TAPE                                        SUNNNN
```

Enter the tape unit device address and the mode setting of the tape to be forward spaced (refer to 7.3.1).

```
nn?#BLKS                                                      SUNNNN
```

Enter the number of blocks to be advanced (maximum number is 9999).

Backspace to a Specific Record (BSR)

You use the BSR function code to backspace a specific number of blocks on your tape.

Procedure

For the message:

```
nn?CUUMM - OUTPUT TAPE                                        SUNNNN
```

Enter the tape unit device address and the mode setting of the tape to be backspaced (refer to 7.3.1).

```
nn?#BLKS                                                      SUNNNN
```

Enter the number of blocks to be backspaced (maximum number is 9999).

Write Tape Marks (WTM)

You use the WTM function code to write tapemarks on your file.

Procedure

For the message:

```
nn?CUUMM - OUTPUT TAPE                                            SUNNNN
```

Enter the tape unit device address and the mode setting of the output tape (refer to 7.3.1).

Rewind Tape (REW)

You use the REW function code to rewind your tape to load point.

Procedure

For the message:

```
nn?CUUMM - OUTPUT TAPE                                            SUNNNN
```

Enter the tape unit device address and the mode setting of the tape to be rewound (refer to 7.3.1).

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:

```
nn?CUUMM - OUTPUT TAPE                                            SUNNNN
```

Enter the tape unit device address and the mode setting of the tape to be rewound (refer to 7.3.1).

Erase 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:

```
nn?CUUMM - OUTPUT TAPE                                    SUNNNN
```

Enter the tape unit device address and the mode setting of the tape to be erased (refer to 7.3.1).

7.4. Disk and Format Label Diskette Functions

The disk and format label diskette functions that can be performed are described in the following subsections.

7.4.1. Operating Considerations

Consider the following when operating with the system utility on disks or format label diskettes:

- An end-of-file record is a disk record on which data length is in binary zeros (not applicable on IDA disks).

- The system utility 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.

7.4.2. Disk and Format Label Diskette Operating Instructions

Proceed as follows to perform a disk or format label diskette function:

1. Place the subject disk/diskette volume on an available unit.

2. As described in 7.1:

   a. Enter the SU symbiont command.

   b. Enter the function code 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).

7.4.3. Printing a Disk or Format Label Diskette in Unblocked Format (DD)

You use the DD function code to print your disk pack or format label diskette in character and hexadecimal format. The DD function does not deblock your logical records.

Procedure

For the messages:

```
    nn?ENTER DVC ADDRESS                                                SUNNNN
```

Enter the disk or diskette unit device address of the disk pack or format label diskette to be displayed.

```
    nn?CCCHH - BEGIN OR FILE-ID                                         SUNNNN
```

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.*

```
    nn?CCCHH - END                                                      SUNNNN
```

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

```
    nn?NO OF TRACKS TO PRINT UP TO 9                                    SUNNNN
```

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 end extent specified in the format label. For DD, a format label diskette is considered to have 74 cylinders, with 2 heads (0 and 1) in each cylinder. (Laced files are discussed in the *System Service Programs (SSP) Operating Guide* (UP-8841) in the section describing diskette prep.)

Example 1

The data on cylinder 3, head 4 through cylinder 6, head 6 on device 300 is to be printed.

For the message:

```
    nn?ENTER DVC ADDRESS                                                SUNNNN
```

Key in:

```
nnΔ300
```

For the message:

```
nn?CCCHH - BEGIN OR FILE-ID                                  SUNNNN
```

Key in:

```
nnΔ00304
```

For the message:

```
nn?CCCHH END                                                 SUNNNN
```

Key in:

```
nnΔ00606
```

Example 2

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

For the message:

```
nn?ENTER DVC ADDRESS                                         SUNNNN
```

Key in:

```
nnΔ440
```

For the message:

```
nn?CCCHH - BEGIN OR FILE-ID                                  SUNNNN
```

Key in:

```
nnΔSEQUENTIAL DISC
```

For the message:

```
nn?NO OF TRACKS TO PRINT UP TO 9                             SUNNNN
```

Key in:

```
nnΔ3
```

7.4.4. Printing a Disk or Format Label Diskette in Deblocked Format (DDR)

You use the DDR function code to print your disk pack or format label diskette in both character and hexadecimal formats. This function is similar to the DD function, except that the printing is reformatted according to the logical record length that is entered.

Procedure

For the message:

```
nn?ENTER DVC ADDRESS                                              SUNNNN
```

Enter the disk/diskette unit device address.

For the message:

```
nn?CCCHH-BEGIN OR FILE-ID                                         SUNNNN
```

Enter in decimal the first cylinder (CCC) and head (HH) to be printed or the file identifier (up to 44 characters) as 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-id is entered.

If cylinder and head are entered, SU displays these messages:

```
nn?CCCHH-END                                                      SUNNNN
```

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

```
nn?ENTER LOGICAL RECORD LENGTH                                    SUNNNN
```

Enter in decimal the logical record size to determine the print format.

```
nn?ENTER BLOCKSIZE                                                SUNNNN
```

Enter in decimal the logical block size. However, if the block size is not an exact multiple of the record size, the following message will be displayed:

```
BLOCKSIZE NOT MULTIPLE OF RECORD SIZE
```

Reenter the record size/block size sequence.

If a file-id is entered, SU displays these messages:

```
nn?NUMBER OF TRACKS TO BE PRINTED UP TO 9                         SUNNNN
```

Enter the number of tracks to be printed. Note that printing begins with the low cylinder and head numbers of the first extent and continues for the requested number of tracks, or until end of file (EOF) is encountered, whichever comes first. Printing is not confined to the extents specified in the format label.

```
nn?ENTER LOGICAL RECORD LENGTH                                    SUNNNN
```

Enter in decimal the logical record size to determine the print format. The block size will be taken from the VOL1 label.

7.4.5. Printing the Disk or Format Label 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 the available space left on your volume and other information regarding your volume
- File information only, giving you the extent and other information regarding the file

VTP edits and prints the volume table of contents (VTOC) information for the requested volume. It will not process other than OS/3-created VTOCs. Use the DD function to print a VTOC from a disk or diskette not created by OS/3.

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:

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

Key in one of the following, where vsn is the volume serial number of the disk pack or format label diskette whose VTOC is to be printed.

For full VTOC listing:

```
nnΔvsn,ALL
```

For a listing of only the device information:

```
nnΔvsn,DI
```

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

```
nnΔvsn,FILE-ID
```

To terminate the VTP function:

```
nnΔEND
```

To terminate SU:

```
nnΔEOJ
```

Notes:

1. *The device address of the disk pack or format label diskette may replace the vsn in the preceding messages.*

2. *When a VTP function terminates, SU produces the following message:*

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

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

Disk and Format Label Diskette VTP Listing Summary

The information listed by VTP is either taken directly from the disk or diskette 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 the *Consolidated Data Management Macroinstructions Programming Guide*, 7004 4607, for details.)

Table 7-5. Summary of Disk/Diskette VTP Information

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
AVAILABLE EXTENT DATA	
START ADDRESS	DL$XT5
END ADDRESS	DL$XC5 (calculated)
EXTENT 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/CYLINDERS	DL$TPC2
LOW HEAD	DL$FLH2
FILE TYPE	DL$FT1
LAST ACCESS DATE	DL$ACDTE
LAST ACCESS TIME	DL$ACTME (milliseconds)
LAST UPDATE DATE	DL$UPDTE
LAST UPDATE TIME	DL$UPTME (milliseconds)
UPDATE JOBNAME	DL$UPID
UPDATE ACCESS CODE	DL$UPACC
PRIOR UPDATE DATE	DL$PUPD
PRIOR UPDATE TIME	DL$PUPT (milliseconds)
PRIOR UPDATE JOBNAME	DL$PUPID
PRIOR UPDATE ACCESS CODE	DL$PUPAC

continued

Table 7-5. Summary of Disk/Diskette VTP Information (cont.)

Field Heading	Field Label
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$SEP2
TOTAL # RECORDS	(calculated: total number of records for sequential FIXBLK and FIXUNB files, and for VARUNB files.)
LOGICAL EXTENT INFORMATION	
PCA NO	(0 through 7)
START CCC' HH	DL$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
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

*Not applicable to MIRAM files on PCA basis.

continued

Table 7-5. Summary of Disk/Diskette VTP Information (cont.)

Field Heading	Field Label
(M)IRAM FILE INFORMATION	
KEY LOCATION	CL$XILOC
KEY LENGTH	CL$XILOC+2
DUPLICATES ALLOWED	CL$XILOC+3,X'80'
CHANGES ALLOWED	CL$XILOC+3,X'40'
DATA RECORD COUNT	CL$COUTR
RECORD SIZE	CL$DREC
INDEX BUFF SIZE	CL$CSIZ

7.4.6. Printing a Short Format VTOC for a Disk or Format Label Diskette (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:

```
nn?ENTER DVC OR VSN OR END                                    SUNNNN
```

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

If SVT is attempted on a data set label diskette, the following message is displayed.

```
SVT VALID ON FORMAT LABEL VOLUMES ONLY
```

SVT Listing Summary for Disk and Format Label Diskette

The various information listed by SVT is either taken directly from the disk or diskette 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 the *Consolidated Data Management Macroinstructions Programming Guide*, 7004 4607, for details.)

Table 7-6. Summary of SVT Information for Disk and Format Label Diskette

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$XU1
CUM CCC HH	(calculated: cumulative)
AIN	DL$SA1

7.4.7. Displaying the Available Disk and Format Label Diskette Extents (AVX)

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

Procedure

For the message:

```
    nn?ENTER DEVICE ADDRESS                                              SUNNNN
```

Enter the did of the disk or diskette drive containing the disk or diskette whose available extents are to be displayed.

If AVX is attempted on a data set label diskette, the following message is displayed:

```
    AVX VALID ON FORMAT LABEL VOLUMES ONLY
```

AVX Listing Summary

The various information listed by AVX is either taken directly from the disk or diskette labels or calculated from data contained in the labels. A summary of the available disk or diskette extent information listed by the AVX function is provided in Table 7-7. (Refer to the *Consolidated Data Management Programming Guide*, 7004 4607, for details.)

Table 7-7. Summary of AVX Information for Disk and Format Label Diskette

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. Data Set Label Diskette Functions

The data set label diskette functions that can be performed are described in the following subsections.

7.5.1. Data Set Label Diskette Operating Instructions

Proceed as follows to request a data set label diskette function:

1. Place the subject data set label diskette volume on an available diskette unit.

2. As described in 7.1:

   a. Enter the SU symbiont command.

   b. Enter the DD or DDR function code either as a command parameter or as a solicited message response to the ENTER REQUIRED FUNCTION message.

   c. Enter the spooling option; otherwise, default is Y (applicable if spooling is configured).

7.5.2. Printing the Data Set Label Diskette in Unblocked Format

You use the DD function to print a single-sided or double-sided data set label diskette in unblocked format.

Single-Sided Diskette in Unblocked Format (DD)

The DD function code prints your data set label diskette in character and hexadecimal format. The DD function does not deblock your logical records.

Procedure

For the messages:

```
nn?ENTER DVC ADDRESS                                      SUNNNN
```

Enter the diskette unit device address of the data set label diskette to be displayed.

```
nn?TTRR-BEGIN                                             SUNNNN
```

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

```
nn?TTRR-END                                               SUNNNN
```

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

Double-Sided Diskette in Unblocked Format (DD)

The DD function code prints your data set label diskette in character and hexadecimal format. The DD function does not deblock your logical records. Your diskette is considered to have 74 cylinders with 2 heads (0 and 1) in each cylinder.

Procedure

For the messages:

```
nn?ENTER DVC ADDRESS                                      SUNNNN
```

Enter the diskette unit device address of the data set label diskette to be displayed.

```
nn?CCCHH - BEGIN                                          SUNNNN
```

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

```
nn?CCCHH - END                                            SUNNNN
```

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

Example

The data on cylinder 3, head 4 through cylinder 6, head 6 on device 300 is to be printed.

For the message:

```
nn?ENTER DVC ADDRESS                                              SUNNNN
```

Key in:

```
nnΔ300
```

For the message:

```
nn?CCCHH - BEGIN                                                  SUNNNN
```

Key in:

```
nnΔ00300
```

For the message:

```
nn?CCCHH END                                                      SUNNNN
```

Key in:

```
nnΔ00601
```

7.5.3. Printing the Data Set Label Diskette in Deblocked Format

You use the DDR function to print a single-sided or double-sided data set label diskette in deblocked format.

Single-Sided Diskette in Deblocked Format (DDR)

The DDR function code prints your data set label diskette in character and hexadecimal format. The DDR function deblocks your logical records.

Procedure

For the messages:

```
nn?ENTER DVC ADDRESS                                              SUNNNN
```

Enter the diskette unit device address of the data set label to be displayed.

```
nn?TTRR-BEGIN                                                     SUNNNN
```

Enter in decimal the first record (RR) and track (TT) to be printed.

```
nn?ENTER LOGICAL RECORD LENGTH                                   SUNNNN
```

Enter in decimal the logical record size to determine the print format.

Double-Sided Diskette in Deblocked Format (DDR)

The DDR function code prints your data set label diskette in character and hexadecimal format. The DDR function deblocks your logical records. Your diskette is considered to have 74 cylinders with 2 heads (0 and 1) in each cylinder.

Procedure

For the messages:

```
nn?ENTER DVC ADDRESS                                             SUNNNN
```

Enter the diskette unit device address of the data set label diskette to be displayed.

```
nn?CCCHH-BEGIN                                                   SUNNNN
```

Enter in decimal the first cylinder (CCC) and head (HH) to be printed. Double-sided diskettes have 72 cylinders and 2 heads, 00 and 01.

```
nn?CCCHH-END                                                     SUNNNN
```

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

```
nn?ENTER LOGICAL RECORD LENGTH                                   SUNNNN
```

Enter in decimal the logical record size to determine the print format.

7.5.4. Printing the Data Set Label Diskette Volume Table of Contents (VTOC)

You use the DD function to print the diskette volume table of contents (that is, the data set labels).

Single-Sided Diskette VTOC (DD)

The DD function code prints your single-sided data set label diskette VTOC.

Procedure

For the message:

```
nn?ENTER DVC ADDRESS                                             SUNNNN
```

Enter the diskette unit device address of the data set label diskette whose VTOC is to be printed.

For the message:

```
nn?TTRR-BEGIN                                                   SUNNNN
```

Key in:

```
nnΔ0008
```

For the message:

```
nn?TTRR-END                                                     SUNNNN
```

Key in:

```
nnΔ0026
```

Double-Sided Diskette VTOC (DD)

The DD function code prints your double-sided data set label diskette VTOC.

Procedure

For the message:

```
nn?ENTER DVC ADDRESS                                            SUNNNN
```

Enter the diskette unit device address of the data set label diskette whose VTOC is to be printed.

For the message:

```
nn?CCCHH-BEGIN                                                  SUNNNN
```

Key in:

```
nnΔ00000
```

For the message:

```
nn?CCCHH-END                                                    SUNNNN
```

Key in:

```
nnΔ00001
```

Section 8
Disk Cache Facility (DCF)

The OS/3 disk cache facility (DCF) increases system performance by reducing the number of physical I/Os to disk devices. I/Os are slow relative to processor speed. The number of physical I/Os is reduced by buffering disk data in main storage (the cache buffer) so that the data is retrieved from memory rather than disk.

When a read is done of a particular area on a disk, the disk cache facility (DCF) reads into a reserved cache storage area all or a portion of the track being read (depending on the segment size when DCF was initialized). Consequently, in subsequent reads of that same disk area, a physical I/O is unnecessary, since that data is already present in storage.

The advantage of DCF is that a reduction in the number of physical I/Os required results in:

- Reduced queuing delays (as I/O requests must wait to be issued in turn)
- Reduced disk seek time (as the disk accessor moves to the proper cylinder)
- Reduced latency (as the disk rotates so that the disk read/write heads reach the proper record)
- Reduced data transfer time (from disk to storage)
- Reduced process time overall

Main storage disk cache is supported via software. The software is included in the system microcode file ($Y$MIC) and is called via the system definition file ($Y$SDF) to maintain consistency between all System 80 models. Refer to the *Installation Guide*, 7004 5505, for information on the system microcode and system definition files.

You can selectively activate or remove individual disk drives to DCF (8.3 and 8.4) or turn it off completely (8.5). Also, you can bypass the caching of specific files using the CACHE parameter of the DD job control statement and the CACHE IOGEN parameter. (See the *Consolidated Data Management Programming Guide*, UP-9978, and the *Installation Guide*, 7004 5505.) Information to help you optimize DCF performance is given in section 8.7.

To use DCF, your system must be configured with the disk cache feature. Following are descriptions of DCF features. Refer to the *System Messages Reference Manual*, 7004 5190, for an explanation of messages and error codes.

8.1. Initializing Disk Cache (CM)

DCF is normally initialized during IPL. However, if it was not initialized during IPL or was shut down, you can initialize it manually by entering the CACHE MEMORY command at the system console. When DCF is initialized, all devices are activated or removed from DCF according to their CACHE IOGEN parameter specifications. Refer to the *Installation Guide*, 7004 5505, for details on this parameter.

Format

```
CM
```

8.1.1. Specifying Cache Buffer Size (CM SIZ)

When cache is initialized at IPL time, the size of the cache buffer is determined by one of the following methods:

- If the last four digits of the cache manager microcode name in $Y$SDF file are not zero, they specify the cache size. For example, CA130256 indicates a cache size of 256KB.

- If the last four digits of the cache manager microcode name in $Y$SDF file are zero (CA130000), a CM01 message prompt asks for a buffer size.

The CM SIZ command lets you override the cache buffer size when you initialize DCF.

Format

```
CM SIZ=n
```

where:

n

Specifies the number of 1,024-byte blocks allocated to the cache buffer. Valid specifications are 160 to 8,192. Note that larger cache buffers generally result in increased performance.

See 8.7 for more information on cache buffer and segment size.

8.1.2. Specifying Cache Buffer Segment Size (CM SEG)

The cache buffer is divided into fixed size segments. The amount of data that DCF reads from a disk is equal to the cache buffer segment size. You can specify the cache buffer segment size by entering the CM SEG command.

Format

```
CM SEG=n
```

where:

n
: Specifies the segment size in 1,024-byte blocks. Valid specifications are 2 to 24. This command overrides the cache segment size previously defined through the CACHESEGSIZE SUPGEN parameter, as described in the *Installation Guide*, 7004 5505.

If you don't specify the CM SEG command, the size specified in the CACHESEGSIZE SUPGEN parameter is used. If you don't specify either parameter, the default values used are as follows:

Cache Buffer Size	Default Cache Buffer Segment Size (in 1,024-byte blocks)
	-
160 - 255K	3
256 - 511K	6
512 - 8,192K	12

Note that using a smaller segment size causes a greater number of segments in your cache buffer. Generally, sequential processing performs better with large segments, and random processing performs better with a larger number of small segments. You can experiment to find the combination that is best suited to your environment. Use the statistics that DCF provides (see 8.6) to help tune these specifications.

The SIZ and SEG options can be separated by a comma and specified together on the CM command.

Format

```
CM SEG=15, SIZ=1000
```

8.2. Resegmenting the Existing Cache Buffer (CM RESEG)

With the unsolicited CM RESEG command, you can resegment the cache buffer that you allocated during DCF initialization with a different segment size. You can use this command without shutting down or reinitializing DCF. However, if you want to change the buffer size, then you must reinitialize DCF. The RESEG command resets the cache statistics to zero.

Format

```
00 CM RESEG=n
```

where:

n

Specifies segment size in 1,024-byte blocks. Valid specifications are 2 to 24.

Note: *Resegmenting the cache buffer resets all cache statistics but does not affect the activated or removed status of the disks on the system.*

8.3. Selective Caching

Normally, all drives and files are cached. The performance of cache memory may be optimized by using cache memory only with the most frequently accessed drives and files. Use selective caching to remove the less frequently used drives and files.

8.3.1. Removing Drives from DCF (CM REM)

You can remove a disk drive from DCF via the CACHE IOGEN parameter (also see the *Installation Guide*, 7004 5505, or via the unsolicited CM REMOVE command. Note that when a disk drive is shared by two processors, the drive must be removed from DCF in any processor that writes to it.

Format

```
00 CM REMOVE dvc#[,dvc,...,dvc#]
```

where:

dvc#

Specifies the device address (drive number) of the drive to be removed from the DCF. You can specify more than one address.

8.3.2. Removing Files from DCF

You can bypass the caching of specific files using the CACHE parameter of the DD job control statement and the CACHE IOGEN parameter. For more information, see the *Consolidated Data Management Programming Guide*, UP-9978, and the *Installation Guide*, 7004 5505.

8.4. Activating Drives to DCF (CM ACT)

You can activate a drive to DCF via the CACHE IOGEN parameter (see the *Installation Guide*, 7004 5505) or via the unsolicited CM ACTIVATE command.

Format

```
00 CM ACTIVATE dvc#[,dvc#,...,dvc#]
```

where:

dvc#

Specifies the device address (drive number) of the drive to be activated to DCF. You can specify more than one address.

8.5. Removing DCF from the System (CM SHUT)

Remove DCF from the system at any time using the unsolicited CM SHUTDOWN command:

Format

```
00 CM SHUTDOWN
```

8.6. Displaying DCF Statistics (CM STA, STARES)

You can monitor DCF operations by displaying DCF statistics. You can display DCF statistics on demand or at periodic, timed intervals. You can also display DCF statistics for a specific drive.

To display DCF statistics immediately, use the unsolicited cache commands CM STA and CM STARES.

Formats

```
00 CM STA
```

Displays statistics on the system console.

```
00 CM STARES
```

Displays statistics on the system console and resets all of the counters including disk device statistics maintained by DCF. It can be used to monitor DCF operations during a specific period of time.

8.6.1. Displaying and Turning Off Timer Statistics (CM STATIME, STARESTIME, TIMER)

Formats

```
00 CM STATIME=n
```

Displays statistics on the system console repeatedly.

where:

```
n
```

Specifies the time interval between displays. Valid values are 1 to 999 minutes.

```
00 CM STARESTIME=n
```

Repeatedly displays statistics and resets all counters maintained by DCF. This command can be used to monitor DCF operations for a specified time interval.

where:

```
n
```

Specifies the time interval between displays. Valid values are 1 to 999 minutes.

```
00 CM TIMER=OFF
```

Turns off the timer you set using the STATIME or STARESTIME command.

8.6.2. Displaying Statistics on a Specified Drive (CM STADISK, STARESDISK)

Formats

```
00 CM STADISK=did
```

Displays statistics for the specified disk on the system console.

where:

```
did
```

Specifies the device address of the disk.

```
00 CM STARESDISK=did
```

Displays statistics and resets the counters for the specified disk.

where:

```
did
```

Is the device address of the disk.

Note: *If you generate an attention interrupt, AVR a disk, or mount another disk, the counters for that disk's device statistics are reset.*

8.6.3. Interpreting DCF Statistics

Figure 8-1 shows samples of the statistics displayed at the system console.

```
00 CM STA
   CM16 READ HIT RATE             73.0%                                  CM0005
   CM17 SEARCH READ HIT RATE      99.5%                                  CM0005
   CM18 UNREFERENCED I/O RATE      0.3%                                  CM0005
   CM19 CACHE BUFFER SIZE         1024k                                  CM0005
   CM20 SEGMENT SIZE (x 1024)        12                                  CM0005
   CM21 TOTAL NUMBER DISK I/O     46170                                  CM0005
   CM22 NUMBER READ HITS          30320                                  CM0005
   CM23 NUMBER WRITES              1846                                  CM0005
   CM24 NUMBER SEARCH EQUALS        248                                  CM0005
   CM25 NUMBER SEARCH HI/EQ        1121                                  CM0005
   CM26 NUMBER READ ERRORS            1                                  CM0005
   CM27 NUMBER READS NOT CACHED    8945                                  CM0005
   CM28 NUMBER SEARCH HITS         1116                                  CM0005
   CM29 NUMBER WRITE THROUGHS      1407                                  CM0005
   CM30 NUMBER UNREFERENCED I/OS    140                                  CM0005
   CM31 NUMBER (CACHE=NO) I/OS       20                                  CM0005
   CM32 READS OVER HALF SEGMENT      50                                  CM0005
   CM33 NUMBER READS              41508                                  CM0005
```

Figure 8-1. DCF Statistics Displayed at the System Console

- Hit rates

  A hit occurs when a read or search/read operation does not have to access the disk because the data is in the cache buffer. Hit rates are the most significant indication of cache performance. The search hit rate is the number of search hits divided by the number of search HI/EQs. (Search equal commands are not cached.)

- Read errors

  Read errors are I/O errors encountered by DCF. When a read error occurs, the I/O is reissued directly into the program's I/O buffer, rather than into the cache buffer. The reissued I/O may be successful.

- Write-throughs

  A write-through occurs when the data being written resides in the cache buffer. The buffer is updated and the data is written to the disk.

- Unreferenced I/Os

  Unreferenced I/Os are a count of the number of segments that were reused without ever having been referenced by read or search/read commands. Segments are reused when space is required for an I/O. The first segments reused are those that were never referenced through a read, search/read, or write (LRU).

- Unreferenced I/O rate

  The unreferenced I/O rate is the number of unreferenced I/Os divided by the number of reads.

- Reads over half segment

  This count is the number of reads that are greater than half the segment size and, therefore, are not cached. This count is *not* included in the hit rate calculations.

- Reads not cached

  DCF automatically bypasses read or search/read commands that are multitrack or without data transfer. This count is included in the hit rate calculations.

- (CACHE=NO) I/Os

  This count records the number of read or search/reads that bypass DCF because of the selective noncaching by file option (see 8.3.2). This count is not included in the hit rate calculations.

Note: *If a read is larger than half of the segment size, it is counted as a read over half segment even when CACHE=NO has been specified.*

8.7. DCF Supportive Information

The following information provides you with background and illustrative information to help you use the disk cache facility optimally.

8.7.1. Cache Modules

There are two modules that support disk cache, the cache symbiont (SL@@CM) and the cache manager (CAxx0000). The cache symbiont is responsible for loading the cache manager, initializing the cache buffer, and processing unsolicited cache commands. Once the manager is loaded and initialized, the symbiont becomes inactive until an unsolicited cache command is entered. The manager is responsible for managing the cache buffer and handling I/O requests to the disk.

8.7.2. Main Storage Layout with Disk Cache

When disk cache is initialized at IPL, the cache buffer is established at the high end of main storage. The cache buffer contains the cache manager, a search/read buffer, and cache segments. The search/read buffer is used by search/read high or equal commands, which will be explained later.

Note: *8470, 8494, and M9720 disks have equivalent logical track sizes.*

The size of the search/read buffer is that of an 8470 or 8417 disk track depending on whether or not an 8470 (or equivalent) disk is configured. The manager is located in the beginning of the cache buffer and is followed by the search/read buffer. If the cache buffer size is greater than 511K, DCF automatically allocates two search/read buffers. Also, if the segment size configured is greater than or equal to the largest track size for a disk on the system, DCF does not build separate search/read buffers. DCF uses any available cache segments for search/read operations.

The rest of the cache buffer is divided into fixed size segments. These segments are used by disk cache to read blocks of data. Each segment is associated with a disk track and device address. It may take several segments to contain a track of data, depending on the segment size. If the segment size is larger than a track, then part of the segment will be unused.

Segments are not reserved for specific disks on the system, but are available for use by any disk, based on a least recently used (LRU) algorithm. Section 8.7.6 explains how the segment size is determined.

Use the following formula to calculate the number of segments in the cache buffer.

$$\text{\#Segments} = \frac{\text{(cache buffer size * 1024) - (cache manager + search/read buffer(s))}}{\text{(segment size * 256)}}$$

The following example applies this formula. The cache manager is approximately 8K.

If:

cache buffer size = 700K
segment size = 48 sectors
search buffer = 24576 bytes (8470)

Then:

#Segments = ((700 * 1024) - (8000 + (2 * 24576))) / (48 * 256)
= (716800 - 57152) / 12288
= 659648 / 12288
= 53

In this example there are 53 segments in the cache buffer. Each segment can hold 48 sectors (12K bytes) of disk data.

When disk cache is initialized manually (via the CM command) the cache buffer cannot be placed in high main storage because that area is reserved for the job region. The cache buffer is obtained from an available buffer in middle main storage.

8.7.3. Cached I/O Commands

The cached I/O commands are read, write, and search/read high/equal commands. Search/read high/equal commands are primarily issued by data management, while searching through the index of a MIRAM file, as a fast method to locate the index record(s) that point(s) to the user data record. The search is based on a key argument that is supplied by the initiator of the search.

Search/read equal commands are not cached due to the infrequency of use. If data on the disk is accessed infrequently and is read into the cache, it displaces other data and increases disk cache overhead. Search/read equal commands are used mainly in VTOC searches by disk space management and in $Y$SHR searches by data management during open and close processing.

8.7.4. I/Os Not Cached

Disk cache does not cache all read and search/read high/equal commands. It is important to know which I/Os are not cached because they may cause the cache statistics to reflect lower hit rates.

The following I/Os are not cached:

- Search/read or read multitrack,
- Read with no data transfer
- Read larger than half of the segment size

A MIRAM file with IRAM characteristics or an IRAM file has an index that is more than one track. Data management issues multitrack searches and reads to this index.

A read with no data transfer is commonly used by data management when issuing a write command with a read verification option.

A read larger than half of the segment size is not cached, so that a program issuing large reads bypasses the caching of data, reduces disk cache overhead, and avoids the displacement of other disk data. If the disk data is accessed infrequently, then a large I/O buffer or the use of the selective caching options (see 8.3) is recommended. If the data is accessed frequently, it is recommended that the I/O buffer be less than or equal to the cache segment size, so that the data is cached.

8.7.5. Monitoring DCF Effectiveness

Generally, the larger the cache buffer, the greater the performance. However, there is an exception. When the production mix is random, there is a significantly large amount of cache overhead with a very low hit rate. In this situation, it is better to run without DCF.

The most effective way to determine the optimal cache buffer and segment sizes is to use the cache statistics to monitor DCF operations, while experimenting with different cache and segment sizes. See Section 8.6 for information on DCF statistics commands.

8.8. Fine-Tuning DCF

8.8.1. Specifying Segment Size

Disk cache segment size is specified by the following methods:

- System generation (SYSGEN)
- Unsolicited cache command (CM RESEG)
- Manual cache loading (CM SIZ and CM SEG)
- Default value

For information on specifying segment size at system generation, see the *Installation Guide*, 7004 5505. See 8.1 for information on the CM#RESEG, SIZ, and SEG commands and default values.

The default segment sizes may not be optimal for every user. You may not have enough main storage to run with a large cache buffer to get large segments. Specifying segment size allows you to run with large segments even with a small cache buffer and, conversely, small segments with a large cache buffer.

If a segment size is not specified, the default values specified in the following table are used:

Cache Buffer Size	Number of 256-Byte Sectors
160 - 255K	12 (3K)
256 - 511K	24 (6K)
512 - 1,024K	48 (12K)

Note that 8417 and 8419 disks have 60 and 50 sectors per track respectively. The maximum default segment size is 48 sectors. Thus, with a 48-sector segment, an 8419 track would use one segment of 48 sectors and a second segment with 2 sectors (46 sectors of the second segment are unused). An 8417 track would use two segments of 48 and 12 sectors (36 sectors of the second segment are unused).

It is recommended that system with predominantly 8419 and/or 8417 disks run with a segment size that minimizes unused segment space (i.e., full track segment size).

The following lists the System 80 disks and the corresponding number of segments per track:

8430/8433	33 sectors (256-byte sectors)
8416/8418	40 sectors
8419	50 sectors
8417	60 sectors
8470/8494/M9720	96 sectors

Note: *8430/33 disks are variable-sectored disks. When you choose a segment size, consider the fact that smaller sector sizes increase track overhead.*

8.8.2. Optimizing Disk Cache Performance

Generally, sequential processing performs better with a small number of larger segments, and random processing performs better with a larger number of small segments. By experimenting with different sizes, you can find the combination best suited to your production environment. Figure 8-1 shows the effect of segment size on run time.

You may need to remove a disk from disk cache for performance reasons. If data accessed infrequently is read into the cache buffer, then other disk data is displaced, a lower hit rate is obtained, and disk cache overhead is increased. For information on the remove and activate commands, see 8.3.1 and 8.4, respectively.

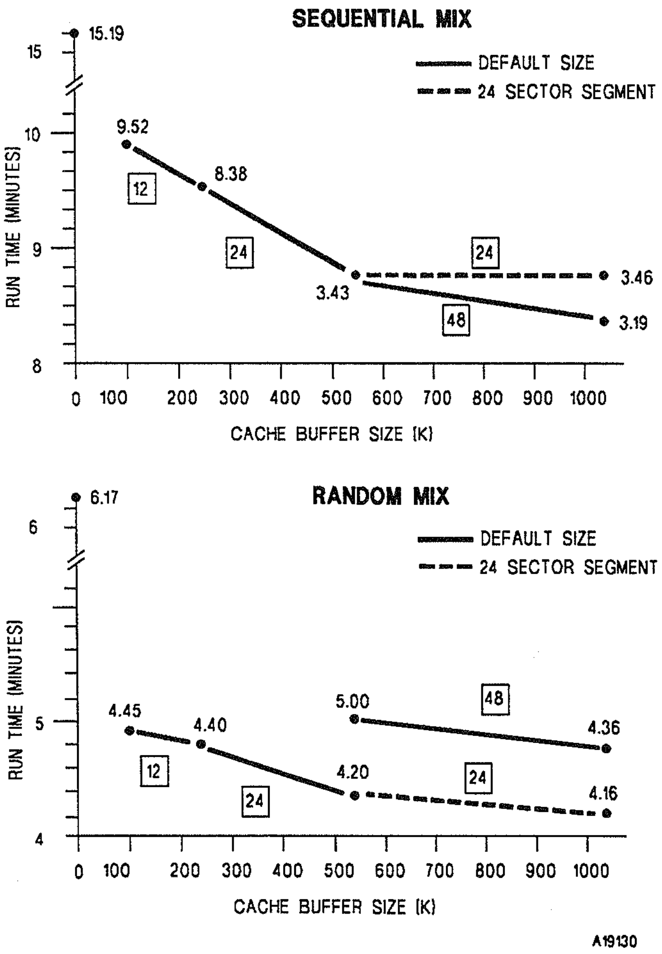


Figure 8-2. Effect of Segment Size on Run Time

Section 9
ONUERL Program

ONUERL is a utility program that edits the contents of the system error log file ($Y$ELOG) and prints a formatted report containing both detailed and summary information. ONUERL also maintains a summary of the error data in the system summary file ($Y$ESUM), which is available for transmission to a Unisys support center when it is written to the system journal.

You have several options when running ONUERL. You can:

- Choose to hold the output in a spool file for later printing
- Select where processing will start in the ELOG file
- Suspend error logging
- Spool error summary data to a diskette
- Obtain a printout that documents what ONUERL does
- Request graphic and tabular displays of accumulated error data (ONELAN program, described later in this appendix)

Note: *It is recommended that you run ONUERL on a planned schedule to help diagnose system problems. You must run the ONUERL program whenever the message "Error log is nearly full" is displayed. The run time for ONUERL varies according to the amount and type of data being processed.*

9.1. Executing ONUERL

You have the option of executing the ONUERL program in two ways:

- Manually, on an as-needed basis, with the ability to change the default (preset) program options each time you run the program
- Automatically, on a fixed, 7-day cycle, with the ability to change the preset options only for the initial running of the program

9.1.1. Manual Program Execution

Enter the following command to run the ONUERL program.

```
RV ONUERL,[,,OPT=Y]
```

- Omit the OPT=Y keyword if you want to run the program by using the preset program options currently in effect. (See 9.2.)

- Include the OPT=Y keyword if you want to modify or view the program options before running ONUERL.

ONUERL displays the following message after loading:

```
nn OS/3 ONUERL-ERROR LOG EDITOR VERSION xx/xx.
```

If you omitted the OPT=Y keyword from your RV command, ONUERL:

- Executes using the preset program options in effect (See 9.2.)

- Prepares its output per the preset program options in effect

- Displays the following message upon normal termination of the program:

```
JCnn  JOB ONUERL  TERMINATED NORMALLY    hh:mm:ss
```

If you included the OPT=Y keyword in your RV command, ONUERL displays the following message:

```
nn MODIFY PROGRAM OPTIONS (Y,N,?)  ?
```

The message requests you to indicate whether you want to make changes to the program options (Y), run the program by using the preset option values as currently defined (N), or display the preset option values for your review (?)

Respond to this message by entering the message identification number (nn) followed by the appropriate character displayed in the message. The program reacts to your response as follows:

? Response

The program displays the current preset value for each option. Then it redisplays the modify program option message to let you run the program by using the preset options shown or to let you redefine the options.

Example:

```
     nn ?
       PRESET OPTIONS :
     OPEN=CURRENT,  SUMMARY=THRESHOLD,  ELOG=YES,  DOC=NO,  THRESHOLDS=NO,
     ELOG=YES,  DUMP=NO,  ESUMFIL=U,  DMPSUM=NO
nn   MODIFY PROGRAM OPTIONS (Y,N,?)  ?
```

N Response

ONUERL executes using the preset program options in effect (shown). The output is prepared as directed by the preset program options and the "Terminated Normally" message is displayed at the successful completion of the program.

Y Response

The program displays a message instructing you to key in the numbers corresponding to the options you want to change.

Example:

```
nn   -    MODIFY PROGRAM OPTIONS (Y,N,?)  ?
nn Y
nn -      ENTER OPTIONS, SEPARATED BY COMMAS (1,2,3,4,5,6,7,8,?)
```

Each number displayed in the message corresponds to a specific program option. To display the correlation between numbers and options, enter the message number followed by a question mark (?). After displaying the numbers and options, the program automatically redisplays the "Enter options" message to let you select the options you want to change.

Example:

```
nn ?
          1=OPEN, 2=SUMMARY, 3=ELOG, 4=DOC, 5=THRESHOLD,
        6=DUMP, 7=ESUMFIL, 8=DMPSUM
nn -  ENTER OPTIONS, SEPARATED BY COMMAS (1,2,3,4,5,6,7,8,?)
```

Select the options you want to change by entering their respective numbers (1 through 8). The program displays (one at a time) each option you selected, along with the choices for modifying that option.

Example:

```
nn        1,2,3,5,6,8
nn -  1: OPEN= (C,P,B,?)
```

Specify your choice by entering the appropriate character displayed.

Example:

```
nn -      1: OPEN= (C,P,B,?)
nn C
nn   -    2: SUMMARY= (T,Y,R,N,D,?)
```

If you want the program to display a brief definition of each choice before making your selection, key in the ?. Otherwise, refer to Table A-1 for an in-depth description of the options and their functions, and then enter your selection.

Example:

```
    nn        2: SUMMARY= (T,Y,R,N,D,?)
    nn ?
         -  T=SPOOL SUMMARY IF THRESHOLD EXCEEDED
            Y=SPOOL SUMMARY TO SYSTEM JOURNAL
            R=REPEAT LAST SUMMARY SPOOLING
            N=NO SUMMARY SPOOLING THIS RUN
            D=INHIBIT SUMMARY SPOOLING PERMANENTLY
    nn?       2: SUMMARY= (T,Y,R,N,D,?)
    nn    T
  nn    3: ELOG= (Y,N,?)
```

ONUERL executes after you complete your last option modification. Figure 9-1 shows a step-by-step example of the messages and response related to ONUERL execution.

9.1.2. Automatic Program Execution

Enter the following command to schedule the ONUERL program for automatic execution cycle:

```
RV ONUERLAS,Ntime+x[,OPT=Y]
```

where:

`ONUERLAS`
The program JCS uses to initiate the automatic execution cycle. Subsequent jobs are identified as ONUERLxx, where xx is the last two digits of the job number.

`N`
Specifies a normal scheduling priority.

`time`
Specifies the time of day (military format) to run the program.

`+x`
Specifies a value (0-6) indicating the number of days between the day that this RV command is entered and the day that the program is executed. For example, you enter the RV command on a Monday but you want the program to run every Friday. There is a 4-day delay between the day the command is entered and the day the program is scheduled to execute. Therefore, the value of x is specified as 4.

`OPY=Y`
Specifies that you want to view and/or change the program options under which ONUERL executes. With the OPT=Y keyword omitted, ONUERL runs using the preset default options currently in effect. (See Table 9-1.)

Be aware that when you execute ONUERL on an automatic schedule, the option changes you make apply only to the initial execution of the program. All subsequent executions of the program are run with the preset default.

If the OPT=Y keyword is omitted from the RV command you enter, ONUERL is scheduled for execution using the preset program options in effect.

If the OPT=Y keyword is included in your RV command, the program displays the following message asking you to indicate whether you want to make changes to the program options (Y), run the program by using the preset option values as currently defined (N), or display the preset option values for your review (?):

```
nn MODIFY PROGRAM OPTIONS (Y,N,?)  ?
```

From this point on, the process of reviewing the preset options and making changes is the same as that described for manually running the ONUERL program. (See 9.1.1.)

9.2. ONUERL Program Options

The ONUERL program options are listed with their associated message display numbers and functional descriptions in Table 9-1. The shaded entries in the table identify the preset (default) value for each option. When executed, ONUERL uses the preset option values unless you elected to make changes to these values (by including the OPT=Y parameter in the RV command). Modified option values (except for SUMMARY=D) remain in effect only for the current execution of the program. Therefore, you must define the changes each time you run the program.

Table 9-1. ONUERL Program Options

1	OPEN= { C P B } C Current. ONUERL opens the error log file $Y$ELOG at the record following the final record of the last ONUERL run. ONUERL processes to the most recent record. P Prior. $Y$ELOG is opened at the same record where it was opened the last time ONUERL was run. The data displayed the last time is displayed again, along with any new data. ONUERL processes up to the most recent record. B Begin. $Y$ELOG is opened at the beginning of the file and displays the oldest record to the most recent record in the file.

continued

Table 9-1. ONUERL Program Options (cont.)

2 SUMMARY= $\left\{\begin{array}{c}Y\\T\\N\\R\\D\end{array}\right\}$

Y

Yes. Summary data gathered during execution of ONUERL is written to the system journal file for tramnsmission later.

T

Threshold. Summary data gathered during the execution of ONUERL is examined for over-threshold occurrence. When total error count exceeds 5 percent of the total I/O count for any device, the program displays a message indicating that summary data is being written to the system journal file; the operator can later transfer this information to the support center.

If over-threshold condition did not occur, summary data gathered is merged into the summary data file $Y$ESUM for future transmission.

N

No. ONUERL operates in local mode and does not spool error summary data to the system journal file even if over-threshold conditions are detected. However, summary data is still saved in the summary data file ($Y$ESUM) for future transmission. Note that summary data present in $Y$ESUM before ONUERL is run is destroyed unless SUMMARY=REPEAT.

R

Repeat. Indicates that summary data previously saved in $Y$ESUM is to be written to the system journal. In this case, only $Y$ESUM is opened, not the error log file ($Y$ELOG). If there are no records in the $Y$ESUM file, an error message is displayed and the program is terminated.

D

Disable. This option permanently disables the writing of error summary data to the system journal file. It is intended for those sites where transmission of summary data is not allowed.

If writing of summary data is disabled, selection of the YES or THRESHOLD option prompts a console message to determine if you want writing to the system journal restored.

If writing was disabled, selection of the REPEAT option prompts a console message indicating that the request cannot be honored.

3 ELOG= $\left\{\begin{array}{c}Y\\N\end{array}\right\}$

Y

Yes. ONUERL has no effect on error logging.

N

No. Error logging is suspended during ONUERL execution.

continued

Table 9-1. **ONUERL Program Options** (cont.)

4	DOC= Y N Y Yes. Operating instructions are printed at the local printer, and the program terminates. N No. Operating instructions are not printed.
5	THRESHOLDS= (Y) (N) (D) Y Yes. Modify recoverable error thresholds. N No. Run with present recoverable error thresholds. D Display. Display current recoverable error thresholds. This option will broadcast the following message: - DO YOU WISH TO ALTER THRESHOLDS (Y,N)
6	DUMP= (Y) (N) Y Yes. Causes a hexadecimal dump of the entire contents of $Y$ELOG. (This is a debug option.) N No. Do not dump $Y$ELOG.
7	ESUMFIL= (I) (U) I Initialize $Y$ESUM file U Update $Y$ESUM file.
8	DUMPSUM= (Y) (N) Y Yes. Causes a hexadecimal dump of the entire contents of $Y$ESUM. (This is a debug option.) N No. Do not dump $Y$ESUM.

The step-by-step sample session shown in Figure 9-1 illustrates the progam messages and operator responses that are used when making changes to the ONUERL program options.

```
     RV ONUERL,,OPT=Y                                             13:28
     R03 RUN PROCESSOR SUCCESSFULLY PROCESSED ONUERL              13:28 RV7939
     JC01  JOB ONUERL   EXECUTING JOB STEP ONUERL00  #001  13:28:30   13:28 ONUERL
     OS/3 ONUERL-ERROR LOG EDITOR VERSION xx/xx.                  13:28 ONUERL
11   - MODIFY PROGRAM OPTIONS (Y,N,?)                             13:28 ONUERL
11 ?                                                              13:28
     - PRESET OPTIONS :                        .                  13:28 ONUERL
     OPEN=CURRENT, SUMMARY=THRESHOLD, ELOG=YES, DOC=NO,           13:28 ONUERL
     THRESHOLD=NO, DUMP=NO, ESUMFIL=U, DMPSUM=NO                  13:28 ONUERL
14   - MODIFY PROGRAM OPTIONS (Y,N,?)    ?                        13:28 ONUERL
14 Y                                                              13:29
15   - ENTER OPTIONS, SEPARATED BY COMMAS (1,2,3,4,5,6,7,8,?)     13:29 ONUERL
15 ?                                                              13:29
     - 1=OPEN, 2=SUMMARY, 3=ELOG, 4=DOC, 5=THRESHOLDS,            13:29 ONUERL
     6=DUMP, 7=ESUMFIL, 8=DMPSUM                                  13:29 ONUERL
17   - ENTER OPTIONS, SEPARATED BY COMMAS (1,2,3,4,5,6,7,8,?)     13:29 ONUERL
17 1,2,3,4,5,6,7,8                                                13:29 ONUERL
18   - 1: OPEN= (C,P,B,?)                                         13:29 ONUERL
18?                                                               13:29
     - C=CURRENT.(ERRORS LOGGED SINCE LAST EDIT).                 13:29 ONUERL
       P=PRIOR.(REPEAT LAST EDIT).                                13:29 ONUERL
       B=BEGIN.(ENTIRE CONTENTS OF ERROR LOG).                    13:29 ONUERL
22   - 1: OPEN= (C,P,B,?)                                         13:29 ONUERL
22C                                                               13:29
23   - 2: SUMMARY= (T,Y,R,N,D,?)                                  13:29 ONUERL
23?                                                               13:29
     - T=SPOOL SUMMARY IF THRESHOLD EXCEEDED.                     13:29 ONUERL
       Y=SPOOL SUMMARY TO SYSTEM JOURNAL                          13:29 ONUERL
       R=REPEAT LAST SUMMARY SPOOLING                             13:29 ONUERL
       N=NO SUMMARY SPOOLING SUMMARY THIS RUN                     13:29 ONUERL
       D=INHIBIT SUMMARY SPOOLING SUMMARY PERMANENTLY.            13:29 ONUERL
29   - 2: SUMMARY= (T,Y,R,N,D,?)                                  13:29 ONUERL
29T                                                               13:29
30   - 3: ELOG=(Y,N,?)                                            13:29 ONUERL
30?                                                               13:29
     - Y=HAS NO EFFECT ON ERROR LOGGING.                          13:29 ONUERL
       N=SUSPEND ERROR LOGGING DURING EDIT.                       13:29 ONUERL
33   - 3: ELOG=(N,Y,?)                                            13:29 ONUERL
33Y                                                               13:29
34   - 4: DOC= (N,Y,?)                                            13:30 ONUERL
34?                                                               13:30
     - N=INHIBIT PRINTING OF OPERATING INSTRUCTIONS               13:30 ONUERL
       Y=PRINT OPERATING INSTRUCTIONS AND TERMINATE               13:30 ONUERL
37   - 4: DOC= (N,Y,?)                                            13:30 ONUERL
37N                                                               13:30
38   - 5: THRESHOLDS= (N,D,Y,?)                                   13:30 ONUERL
38?                                                               13:30
     - N=RUN WITH CURRENT THRESHOLDS                              13:30 ONUERL
       D=DISPLAY CURRENT THRESHOLDS                               13:30 ONUERL
       Y=MODIFY RECOVERABLE ERROR THRESHOLDS                      13:30 ONUERL
```

Figure 9-1. Sample ONUERL Program Execution (Part 1 of 2)

```
42   - 5: THRESHOLDS= (N,D,Y,?)                                  13:30 ONUERL
42N                                                              13:30
43   - 6: DUMP= (N,Y,?)                                          13:30 ONUERL
43?                                                              13:30
     - N=INHIBIT DUMP.                                           13:30 ONUERL
       Y=HEX. DUMP OF ENTIRE $Y$ELOG.                            13:30 ONUERL
     NOTE: THIS OPTION WILL NOT BE ACCEPTED IF SUMMARY ON.       13:30 ONUERL
47   - 6: DUMP= (N,Y,?)                                          13:30 ONUERL
48N                                                              13:31
49   - 7: ESUMFIL= (U,I,?)                                       13:31 ONUERL
49?                                                              13:31
     - U=UPDATE ESUM                                             13:31 ONUERL
       I=INITIALIZE ESUM.                                        13:31 ONUERL
52   - 7: ESUMFIL= (U,I,?)                                       13:31 ONUERL
52U                                                              13:31
53   - 8: DMPSUM= (N,Y,?)                                        13:31 ONUERL
53?                                                              13:31
     - N=INHIBIT DUMP                                            13:31 ONUERL
       Y=HEX. DUMP OF $Y$ESUM FILE.                              13:31 ONUERL
56   - 8: DMPSUM= (N,Y,?)                                        13:31 ONUERL
56N                                                              13:31 ONUERL
     - LAST RUN ON    yy/mm/dd  AT    hh:mm:ss                   13:32 ONUERL
     SORT MI00 END OF SORT                                       13:32 ONUERL
     SORT AI86 RECORD COUNT          175 IN,          0 DELETED  13:32 ONUERL
      - THRESHOLD EXCEEDED                                       13:33 ONUERL
      - LOGICAL UNIT= 12 VOL= SJ0001                             13:33 ONUERL
      - CONTACT CUSTOMER SUPPORT CENTER FOR TRANSMISSION         13:33 ONUERL
     JC02  JOB ONUERL   TERMINATED NORMALLY          13:34:06    13:34 ONUERL
```

Figure 9-1. Sample ONUERL Program Execution (Part 2 of 2)

9.3. SDF$DSP System Program

When you run ONUERL, the content of the system definition file spools to the system journal diskette after all summary data is spooled. It also spools simultaneously to the local printer.

9.4. ONUERL Console Messages

Table 9-2 lists the console messages that may be displayed during execution of the ONUERL program. An explanation and operator action, where required, also are provided.

Table 9-2. ONUERL Console Messages

Message/Explanation/Operator Action

`ERROR READING PARAM, R0=e-code. TERMINATE? (Y,N)`

An error was detected in reading a // PARAM statement from the job control stream, where *e-code* specifies the error encountered. Respond Y to terminate program. Respond N to ignore.

`INVALID ENTRY nnnnn. IGNORE? (Y,N)`

A // PARAM statement has the wrong format, where nnnnn is a display of the PARAM statement. Respond Y to ignore statement and continue. Respond N to terminate program.

`$Y$ELOG IS EMPTY`

$Y$ELOG contains no new error data since the last time ONUERL was run. ONUERL is terminated.

`ERROR ACCESSING  $Y$ELOG. R0=e-code`

An error was detected in attempting to access $Y$ELOG, where *e-code* is the error encountered. The program makes on attempt at recovering. If unsuccessful, ONUERL terminates.

`INVALID DATA IN $Y$ELOG. JOB TERMINATED`

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

`FIRST RECORD NOT HISTORICAL RECORD. JOB CANCELED`

The beginning of $Y$ELOG cannot be found. ONUERL terminates.

`COTABL FULL. JOB CANCELED`

The capacity of the internal device list was exceeded. ONUERL terminates.

`MEDIA TABLE FULL`

The capacity of the media error table was exceeded. Further media errors are not saved.

`NO SUMMARY DATA AVAILABLE. JOB CANCELED`

In running ONUERL in the SUMMARY, REPEAT mode, no summary data is found in $Y$ESUM. ONUERL terminates.

`$Y$ESUM nnnn error. CD$FNMC: sense-data`

An error was detected in accessing $Y$ESUM, where nnnn is OPEN, CLOSE, INPUT, or OUTPUT. *Sense-data* is the contents of the four sense bytes returned by the system. ONUERL terminates.

continued

Table 9-2. ONUERL Console Messages (cont.)

Message/Explanation/Operator Action

```
$Y$ESUM ERROR: EOF. FILE CAPACITY EXCEEDED
```

An attempt to create a new record failed for lack of space. ONUERL terminates. $Y$ESUM must be erased and reallocated. Then ONUERL can be re-executed. File $Y$ESUM is reinitialized with the latest device list.

```
NEW SUMRCD CREATED FOR DEVICE nnn
```

This channel/device number was assigned to a new or different device type. A new record must be created and stored in $Y$ESUM.

```
INVALID DATE/TIME: xxx. JOB CANCELED.
```

A test of system date and time shows invalid values that are displayed. Error records cannot be summarized reliably. ONUERL is canceled. Rerun ONUERL when date and time have been properly entered.

```
$Y$ESUM NOT INITIALIZED.
```

This message appears if $Y$ESUM file was not initialized. It is followed by:

```
    ENTER CUSTOMER-ID. (_________,?)
```

This message is displayed during $Y$ESUM file initialization. Enter your 9-character customer ID. The following message then appears:

```
    CUSTOMER-ID IS your-customer-id. CONFIRM (Y,N)
```

This message echoes the customer ID you entered and requests your confirmation.

If ? is entered as a response to the customer ID request, the following message appears:

```
    DEPRESS EOT IF ID. NOT AVAILABLE.
```

```
SUMMARY PERMANENTLY DISABLED. RESTORE? (Y,N)
```

Option SUMMARY=YES or THRESHOLD was selected, but SUMMARY was previously disabled. Respond Y to restore SUMMARY capabilities. Respond N to ignore this request.

```
CANNOT HONOR SUMMARY REQUEST. JOB CANCELED.
```

Option SUMMARY=REPEAT was selected, but SUMMARY was previously disabled. No SUMMARY data is available for transmission.

```
SUMMARY NOT TRANSMITTED. SEND? (Y,N)
```

Over-threshold occurred in the last execution of ONUERL, and SUMMARY data was not transmitted. Transmission must be made before the program can proceed with collection of new data. Respond Y to permit transmission. Respond N to terminate program.

```
INVALID DISKETTE (R,C?)
```

The diskette in FDD1 is not a system journal diskette. Respond R to retry after loading the correct diskette. Respond C? to cancel program.

continued

Table 9-2. ONUERL Console Messages (cont.)

Message/Explanation/Operator Action
`THRESHOLDS EXCEEDED` Error thresholds were exceeded for the processing period.
`LOGICAL UNIT= xx VOL=xxxxxxxxxxx` Identifies the logical unit and volume being referenced.
`OVER-WRITE, NEXT LU, END JOB? (O,N,E,?)` Request for operator permission to write to the LU and volume previously displayed. The choices available are: 1. Overwrite the volume on the current logical unit. 2. Skip this logical unit and proceed to the next available logical unit. 3. Terminate the program without writing any data.
`OVER-WRITE, ANOTHER DISKETTE, END JOB? (O,D,E,?)` Request to operator when LU00 (FDD) is being accessed and the volume inserted is not SJ0001. The choices available are: 1. Overwrite the volume on logical unit 00. 2. Allow operator to insert another diskette. 3. Terminate the program without writing any data.
`INSERT 1.44MB FORMATTED DISKETTE IN FDD1 AND 60` Message displayed when logical unit 00 (FDD) is used as system journal. Insert diskette into FDD and then enter "60 *jobname*"
`LOGICAL UNITS/MEDIA FOR SYSTEM JOURNAL NOT USABLE. PROGRAM TERMINATING` The permitted logical units were not usable.

9.5. ONUERL Reports

The reports obtained as a result of running the ONUERL program provide a comprehensive listing of all errors contained in the $Y$ELOG file, as defined by the program parameters. The main body of the reports consists of a single line for each error log entry. The entries are sorted chronologically by channel device number. The reports also include a summary of sense bits in error for the device, and a summary of total error count versus total count of valid I/Os by device address.

Note: *ONUERL reports do not include your customer-id unless you add it to the control stream. To do so, use the general editor (EDT). (The ONUERL job control stream resides in the $Y$JCS file located on SYSRES.) When you've read the file into the workspace, look for the // PARAM LOG-ID=*'log id data' *statement, where* 'log id data' *is 29 characters reserved for your installation name followed by your 9-character customer-id. This information must be enclosed in single quotes. When you've saved the file, your customer-id is permanently added to the control stream.*

The description of each summary report heading follows:

`device name`
Names the device in error.

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

`date and time`
The date and time the error was written to the error log.

`VSN`
The volume serial number of the media in use at the time of error, if present in the error log.

`J-NME`
The name of the job (if present in the error log) encountering the error.

`J-ADD`
The starting address in main storage for the job running at time of error report.

`J-SIZE`
Main storage assigned to user job.

`CEIQ/CAIQ`
Identifies the queue reporting the error: channel attention interrupt queue (CAIQ) or channel error interrupt queue (CEIQ).

`CCW`
Identifies the channel command word.

`I/O`
The number of valid I/Os prior to the error.

`RTRY`
The number of times the I/O was attempted before reporting the error.

`STST`
The device and subchannel status of the device in error.

`DSX`
The device status extension byte.

`DSSC`
The device state sequence code.

`DSW`
The directive status word.

`SS0-SS23`
The sense bytes transmitted by the device to the channel at the time of the error. Up to 24 sense bytes are printed.

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

`REC`
Indicates whether the I/O recovered (Y) or not (N).

`SENSE BYTE X'80'-X'01'`
Summary counts of sense bits set for the device. These counts are taken from the error log file and will vary depending upon the OPEN option that was selected.

`DEVICE #`
The device address (channel, subchannel, and device number) of the device in error.

`DEVICE (did) number of ERROR/S OCCURRED FOR number of VALID I/O'S.`
By device address, a summary of total errors versus total I/Os. These counts are taken from the error log file and will vary depending upon the OPEN program option that was selected.

`VSN/ERROR TABLE FOR DVC (did)`
By device address and VSN, a summary of all errors taken from the error log file. The count will vary depending upon the OPEN program option that was selected.

Note: *The total of all VSN errors and the device error count will typically be equal only if all device errors are associated with a VSN.*

Samples of the reports produced by the ONUERL program are shown in Figure 9-2. The samples provided may differ slightly from the actual reports produced for your system. Supervisor characteristics and ELOG flags displayed in the system description lines of the ONUERL report are defined in Tables 9-3 and 9-4. MSE display and CPU retry stack log-out display of the ONUERL report are defined in Tables 9-5 and 9-6. Exigent machine check displays are defined in Table 9-7.

```
OS/3 ONUERL-ERROR LOG EDITOR VERSION xx/xx,    CUST.ID:----------------------          SERIAL# ____ yy/mm/dd  hh.mm  PAGE XX

*****device name*********************    *****device name*********************    *****device name*********************
*    DVC TYPE CODE:                 *    *    DVC TYPE CODE:                 *    *    DVC TYPE CODE:                 *
*    CHAN./DVC ADDR.: 00 00         *    *    CHAN./DVC ADDR.: 00 00         *    *    CHAN./DVC ADDR.: 00 00         *
*    FEATURES:                      *    *    FEATURES:                      *    *    FEATURES:                      *
*    24x80 SCREEN                   *    *    24x80 SCREEN                   *    *    24x80 SCREEN                   *
*    LGERMSK: xxxx, LGEMSK: xxxx    *    *    LGERMSK: xxxx, LGEMSK: xxxx    *    *    LGERMSK: xxxx, LGEMSK: xxxx    *
*************************************    *************************************    *************************************

OS/3 ONUERL-ERROR LOG EDITOR VERSION 48/29          CUST. ID.: XXXXXXXXX                             yy/mm/dd   hh.mm PAGE XX
CUSTOMER ID.: C42722  . OS/3 VER.   15,    REV.  00,  DATE   yy/mm/dd, TIME  07.27.30, FLAGS 0080, CHARACTERISTICS 33BE0080
SYSRES: 401, PRINTER: 330, READER: 332, $Y$ELOG: 401 MAIN STORAGE SIZE; 00100000, USER MEMORY SIZE:000C2400
 BIT MAP ADDRESS: 01F478, BITMAP WORD COUNT:    1. BITMAP DATA: (IN BITS. EACH BIT=64K. O=ON LINE, 1=OFF LINE)
00000000 00000000 11111111 1111111


OS/3 ONUERL-ERROR LOG EDITOR VERSION 48/29      CUST. ID.: XXXXXXXXX                           yy/mm/dd   hh.mm PAGE XX

* EMCIC *   yy/mm/dd 20.40.28  JOB:           .   ADDR: 000000.  SIZE: 000000.
MCIC: 0800005D 20000000.   PSW: E0010000 00000000.    TOD: 9786DAC6 33CE8000.
SPV RGS:00: 00000000, 01: 0000D7C0, 02: 000261B8,  03: 600261D2, 04: 000030D2, 05: 00008860, 06:046FAD42, 07: 0000E240,
        08: 00001720, 09: 0000E240, 10: 400034FA, 11: 000000400, 12: 0000DE94, 13: 40026IBA, 14: 000034F4, 15: 0000320A,
MMU LOG #0   0: 00000000 00000000,  1: 00000000 00000000,   2: 00000000 00000000,
             3: 00000000 00000000,  4: 00000000 00000000,   5: 00000000 00000000,
ED MCIC: 0800005D 30000000.  ED ER0: 00000390.  ED ER1: 110201F0.
INST. + CACHE LOGOUT: 0: 00000000 00000000,  1: 00000000 00000000,
OPERAND CACHE LOGOUT: 0: 00000000 00000000,  1: 00000000 00000000,
FLOATING POINT LOGOUT:0: 00000000 00000000,  1: 00000000 00000000,
```

Figure 9-2. Sample Error Log Report Format (Part 1 of 10)

```
OS/3 ONUERL-ERROR LOG EDITOR VERSION 48/29      CUST. ID.: XXXXXXXXX                              yy/mm/dd    hh.mm PAGE XX

BPU LOGOUT: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
            00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
            00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
            00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 9786DAC6 33CE8000
LOW MEMORY (X'310'-'313'): 00000000
MACHINE CHECK INTERRUPT CODES:
SUBCLASS: EXTERNAL DAMAGE
AREA CODE: I/O EXTERNAL DAMAGE: CHC AND IOP ABNORMALITY
VALIDITY:    8

OS/3 ONUERL-ERROR LOG EDITOR VERSION 48/29      CUST. ID.: XXXXXXXXX                              yy/mm/dd    hh.mm PAGE XX

device name. CHDV: xxxx, ADDR: xxxxxx. SZE: xxxxxx

             ICW/CCW                                   I/O RTRY STST DSX DSSC DSW SSO-1  2-3 4-5 6-7  8-9  10-11 OPR  REC
yy/mm/dd hh.mm.ss    VSN:______    J-NME:________    J-ADD:______    J-SIZE:______
xxxxxxxxxxxxxxxxxxxxxxxx      xxxxxxxxxxxxxxxxxxxxxxx  xxxxx xx xxxxx xxxx xx  xx xxxxx xxxx xxxx xxxx xxxx xxxx  x    x

OS/3 ONUERL-ERROR LOG EDITOR VERSION 48/29      CUST. ID.: XXXXXXXXX                              yy/mm/dd    hh.mm PAGE XX

U-14 TAPE DR., CHDV: 180
             CCW/ICW                                   I/O  RTRY STST DSX DSSC DSW SSO-1  2-3 4-5 6-7  8-9  10-11 OPR  REC
yy/mm/dd hh.mm.ss    VSN:TAPE1      J-NME:DUMPTAPE      J-ADD:______     J-SIZE:______
C308A80C60002900 0108A80C20022900                        2   2 0E00 800 35   00  0840  0084 0000 0000 0000 0000        Y
CMD: WRITE              XPCT BLK:   0001: RCVD BLK:   0001                                  VRC SPEEC CHK
```

Figure 9-2. Sample Error Log Report Format (Part 2 of 10)

```
SENSE BYTE   X'80'    X'40'    X'20'    X'10'    X'08'    X'04'    X'02'    X'01'    DEVICE # xxx
     0        xxx      xxx      xxx      xxx      xxx      xxx      xxx      xxx
     1        xxx      xxx      xxx      xxx      xxx      xxx      xxx      xxx
     2        xxx      xxx      xxx      xxx      xxx      xxx      xxx      xxx
     3        xxx      xxx      xxx      xxx      xxx      xxx      xxx      xxx
     4        xxx      xxx      xxx      xxx      xxx      xxx      xxx      xxx
     5        xxx      xxx      xxx      xxx      xxx      xxx      xxx      xxx
     6        xxx      xxx      xxx      xxx      xxx      xxx      xxx      xxx
     7        xxx      xxx      xxx      xxx      xxx      xxx      xxx      xxx
     8        xxx      xxx      xxx      xxx      xxx      xxx      xxx      xxx
     9        xxx      xxx      xxx      xxx      xxx      xxx      xxx      xxx
     A        xxx      xxx      xxx      xxx      xxx      xxx      xxx      xxx
     B        xxx      xxx      xxx      xxx      xxx      xxx      xxx      xxx
   xxx ERROR/S OCCURRED FOR   xxxxxxx VALID I/O's

VSN/ERROR TABLE FOR DVC 1B1.
VSN:TAPE2  :       24 ERRORS FOR   7475 I/O'S
VSN:SCRT   :       33 ERRORS FOR  12612 I/O'S
VSN:TEST01 :        6 ERRORS FOR   6283 I/O'S
VSN:OLMTAP :        1 ERRORS FOR      5 I/O'S
VSN:R512Q1 :       22 ERRORS FOR     25 I/O'S

OS/3 ONUERL-ERROR LOG EDITOR VERSION 48/29      CUST. ID.: XXXXXXXXX                                  yy/mm/dd   hh.mm PAGE XX

8470 DISC DR., CHDV: 290, ADDR: 000000,  SZE: 000000
               CCW                 I/O     RTRY STST DSSC   0-3   4-7    8-11   12-15    16-19    20-23   OPR REC
yy/mm/dd hh.mm.ss    VSN:REL08S     J-NME:........     J-ADD:......     J-SIZE:......
0000000000000000  1300516860000008       1  1  0200  35 00000000 00000038 00000000 00000000 00000000 00000000          Y
CMD: RECALIB.        CLY: 768,   HD:N/A .
```

Figure 9-2. Sample Error Log Report Format (Part 3 of 10)

```
SENSE BYTE   X'80'    X'40'    X'20'    X'10'    X'08'    X'04'    X'02'    X'01'    DEVICE # xxx
     0        xxx      xxx      xxx      xxx      xxx      xxx      xxx      xxx
     1        xxx      xxx      xxx      xxx      xxx      xxx      xxx      xxx
     2        xxx      xxx      xxx      xxx      xxx      xxx      xxx      xxx
     3        xxx      xxx      xxx      xxx      xxx      xxx      xxx      xxx
     4        xxx      xxx      xxx      xxx      xxx      xxx      xxx      xxx
     5        xxx      xxx      xxx      xxx      xxx      xxx      xxx      xxx
     6        xxx      xxx      xxx      xxx      xxx      xxx      xxx      xxx
     7        xxx      xxx      xxx      xxx      xxx      xxx      xxx      xxx
     8        xxx      xxx      xxx      xxx      xxx      xxx      xxx      xxx
     9        xxx      xxx      xxx      xxx      xxx      xxx      xxx      xxx
     A        xxx      xxx      xxx      xxx      xxx      xxx      xxx      xxx
     B        xxx      xxx      xxx      xxx      xxx      xxx      xxx      xxx
   xxx ERROR/S OCCURRED FOR   xxxxxxx VALID I/O's

VSN/ERROR TABLE FOR DVC xxx.
VSN:REL08S :        x ERRORS FOR   xxxx I/O'S  CYL   HEAD     ERR
                                                80    N/A       04
                                               768    N/A       01
                                                20    N/A       03

OS/3 ONUERL-ERROR LOG EDITOR VERSION 48/29      CUST. ID.: XXXXXXXXX                                 yy/mm/dd   hh.mm PAGE XX

8418 DISC DR., CHDV: 2A1
               CCW                                     I/O RTRY STST DSX DSSC DSW SS0-1  2-3 4-5 6-7  8-9  10-11 OPR  REC
CMD: SEEK-READ      CYL: 711,  HD:N/A .                                                 DTA FLD ECC
yy/mm/dd hh.mm.ss    VSN:SCRT        J-NME:PREP18       J-ADD:______     J-SIZE:______
0000000000000000 17005B0060000008                         1  1  0E00 0000 35  00  0811  0220  0C06 8000 0000  0000      Y
CMD: SEEK-READ       CYL: 711,   HD:N/A.                                                DTA FLD ECC
```

Figure 9-2. Sample Error Log Report Format (Part 4 of 10)

```
SENSE BYTE  X'80'    X'40'    X'20'    X'10'    X'08'    X'04'    X'02'    X'01'    DEVICE # xxx
     0       xxx      xxx      xxx      xxx      xxx      xxx      xxx      xxx
     1       xxx      xxx      xxx      xxx      xxx      xxx      xxx      xxx
     2       xxx      xxx      xxx      xxx      xxx      xxx      xxx      xxx
     3       xxx      xxx      xxx      xxx      xxx      xxx      xxx      xxx
     4       xxx      xxx      xxx      xxx      xxx      xxx      xxx      xxx
     5       xxx      xxx      xxx      xxx      xxx      xxx      xxx      xxx
     6       xxx      xxx      xxx      xxx      xxx      xxx      xxx      xxx
     7       xxx      xxx      xxx      xxx      xxx      xxx      xxx      xxx
     8       xxx      xxx      xxx      xxx      xxx      xxx      xxx      xxx
     9       xxx      xxx      xxx      xxx      xxx      xxx      xxx      xxx
     A       xxx      xxx      xxx      xxx      xxx      xxx      xxx      xxx
     B       xxx      xxx      xxx      xxx      xxx      xxx      xxx      xxx
     x ERROR/S OCCURRED FOR   xxxxxxx VALID I/O's

VSN/ERROR TABLE FOR DVC xxx.
VSN:REL08S :       x ERRORS FOR   xxxx I/O'S  CYL   HEAD     ERR
                                               80    N/A       04
                                              768    N/A       01
                                               20    N/A       03

OS/3 ONUERL-ERROR LOG EDITOR VERSION 48/29     CUST. ID.: XXXXXXXXX                   yy/mm/dd   hh.mm PAGE XX

8420/22 DSKTE, CHDV: C20, ADDR: 000000, SZE: 000000
               ICW                                     I/O RTRY STST DSX DSSC DSW SSO-1  2-3 4-5 6-7  8-9  10-11 OPR  REC
yy/mm/dd hh.mm.ss    VSN:......    J-NME:.........    J-ADD:......    J-SIZE:......
5600886A2000008000000070 000000000000000000000000       1  1  0000 0000 35   03  0000   0000   0000 0000 0000   0000      N
CMD: READ VOL. ID
                                                                  I/O TERMINATION UNKNOWN
```

Figure 9-2. Sample Error Log Report Format (Part 5 of 10)

```
SENSE BYTE   X'80'    X'40'    X'20'    X'10'    X'08'    X'04'    X'02'    X'01'    DEVICE # xxx
     0        xxx      xxx      xxx      xxx      xxx      xxx      xxx      xxx
     1        xxx      xxx      xxx      xxx      xxx      xxx      xxx      xxx
     2        xxx      xxx      xxx      xxx      xxx      xxx      xxx      xxx
     3        xxx      xxx      xxx      xxx      xxx      xxx      xxx      xxx
     4        xxx      xxx      xxx      xxx      xxx      xxx      xxx      xxx
     5        xxx      xxx      xxx      xxx      xxx      xxx      xxx      xxx
     6        xxx      xxx      xxx      xxx      xxx      xxx      xxx      xxx
     7        xxx      xxx      xxx      xxx      xxx      xxx      xxx      xxx
     8        xxx      xxx      xxx      xxx      xxx      xxx      xxx      xxx
     9        xxx      xxx      xxx      xxx      xxx      xxx      xxx      xxx
     A        xxx      xxx      xxx      xxx      xxx      xxx      xxx      xxx
     B        xxx      xxx      xxx      xxx      xxx      xxx      xxx      xxx
     x ERROR/S OCCURRED FOR   xxxxxxx VALID I/O's

VSN/ERROR TABLE FOR DVC xxx.
VSN:ID16J0 :        x ERRORS FOR      x I/O'S  TRK   SCTR     ERR
                                                01     01        05
VSN:JHPDD1 :        x ERRORS FOR   xxxx I/O'S  TRK   SCTR     ERR
                                                36     06        05
                                                36     07        05
VSN:JHPDD2 :        x ERRORS FOR      x I/O'S  TRK   SCTR     ERR
                                                36     08        05

DEVICE xxx: xxxx ERRORS OCCURRED FOR  xxxxxxx VALID I/O'S
DEVICE xxx: xxxx ERRORS OCCURRED FOR  xxxxxxx VALID I/O'S
DEVICE xxx: xxxx ERRORS OCCURRED FOR  xxxxxxx VALID I/O'S
DEVICE xxx: xxxx ERRORS OCCURRED FOR  xxxxxxx VALID I/O'S
DEVICE xxx: xxxx ERRORS OCCURRED FOR  xxxxxxx VALID I/O'S
DEVICE xxx: xxxx ERRORS OCCURRED FOR  xxxxxxx VALID I/O'S
DEVICE xxx: xxxx ERRORS OCCURRED FOR  xxxxxxx VALID I/O'S
DEVICE xxx: xxxx ERRORS OCCURRED FOR  xxxxxxx VALID I/O'S
DEVICE xxx: xxxx ERRORS OCCURRED FOR  xxxxxxx VALID I/O'S
DEVICE xxx: xxxx ERRORS OCCURRED FOR  xxxxxxx VALID I/O'S
DEVICE xxx: xxxx ERRORS OCCURRED FOR  xxxxxxx VALID I/O'S
```

Figure 9-2. Sample Error Log Report Format (Part 6 of 10)

```
OS/3 ONUERL-ERROR LOG EDITOR VERSION 48/29      CUST. ID.: XXXXXXXXX                              yy/mm/dd   hh.mm PAGE XX

OS/3 VER. 14/00   CUST. ID.: XXXXXXXXX   yy/mm/dd     hh.mm.ss
FLAGS 0080, CHAR. 73B30080,   MAIN STOR. 00600000
SYSRES: 2A0, PRNTR: 005, READER:   C20, $Y$ELOG: 2A0
BIT MAP DATA (IN BITS. ONE BIT=64K.   0=ON LINE, 1=OFF LINE) :
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000

OS/3 ONUERL-ERROR LOG EDITOR VERSION 48/29      CUST. ID.: XXXXXXXXX                              yy/mm/dd   hh.mm PAGE XX

VER. 48/29 - SENSE SUMMARY DATA AT:  hh.mm
CUSTOMER ID.:XXXXXXXXX
DVC LIST:C12,C14,C18,. . . . . . . .
OVER-THRESHOLD DVC/S:XXX,XXX,XXX.......
SUMMARY SPAN:  yy/mm/dd. TO yy/mm/dd.

U-RCD ADDR         390
DEVICE            0770
THRESH. VALUE     .050
TOT. I/O COUNT      15
RECOV. COUNT         4
UNRECOV. COUNT       5
OPR RTRY             3
BUF LOAD CHECK       9

TAPE ADDR           5D3    480    481
DEVICE            U26/28 BT32   BT32
THRESH. VALUE      .050   .050   .050
TOT I/O COUNT     58582  68231  81323
NOISE                      114     42
SKEW CHCK             6
END CHK                      1
RD/WR VRC CHK.             174    211
MTE/LRC CHK.                 3      6
ENV/EDD CHK.         29    101     46
```

Figure 9-2. Sample Error Log Report Format (Part 7 of 10)

```
DISK ADDR          1A1    1A2    1A3
DEVICE            8494   8494   8494
THRESH. VALUE     .050   .050   .050
TOT I/O COUNT     6267   8746  12089

DISK ADDR          1B6
DEVICE            8419
THRESH. VALUE     .050
TOT I/O COUNT      142
RECOV. COUNT        12
UNRECOV. COUNT      68
OPR RTRY             2
DVC CHCK            74
ID FLD DATA CHK      5
STOP STATE           2

DSKT ADDR          C21
DEVICE            8420/22
THRESH. VALUE     .050
TOT I/O COUNT       19
RECOV. COUNT         7
UNRECOV. COUNT       4
DVC NOT RDY          5
NO DTA SEP           6
```

Figure 9-2. Sample Error Log Report Format (Part 8 of 10)

```
BETWEEN yy/mm/dd AND yy/mm/dd, OVER-THRESHOLD DEVICES:

* EMCIC *

DATE: yy/mm/dd TIME: hh.mm.ss   MCIC: 2020000D 000BA9D0
    ED MCIC: 00000000 00000000   PSW: E0040000 9006825C
    ED R0: 00000000  ED R1: 00000000
DATE: yy/mm/dd TIME: hh.mm.ss   MCIC: 2020000D 000BBD60
    ED MCIC: 00000000 00000000   PSW: E0040000 9006825C
    ED R0: 00000000  ED R1: 00000000

ERROR ACTIVITY BY VSN.
 DVC  VSN       TIC      TOC     TIC/TOC

2A0   REL08X       24    67921  0.000353
CYL,ER#: 155,   2; 536,   1; 466,   4;  28,  2;
 162,   10; 132,   2; 140,   1; 465,   1; 776,  1;

      REL082       22    68105  0.000323
CYL,ER#: 462,   3; 466,   4; 464,   2; 163,  1;
 162,    1; 543,   2; 542,   1; 140,   2; 781,  1;
  28,    2; 155,   3;

2A1   DATAPK       89     39856  0.002233
CYL,ER#:  54,   2;  69,   12;  81,   7; 145, 12;

OS/3 ONUERL-ERROR LOG EDITOR VERSION 48/29     CUST. ID.: XXXXXXXXX                                                 yy/mm/dd   hh.mm PAGE XX

  70,  18;  71,  11;  00,   8;  74,   9;    55,     9;
   53,  1;
      SCRT         49    51091  0.000959
CYL,ER#:  11,   4; 424,   8; 453,  11; 711,    12;
 764,  14;

290   REL08S       8     3095  0.002585
CYL,ER#:  80,   4; 768,  1;  20,   3;
C20   ID16J0       5        6  0.833333
TK,SCTR,ER#:  01, 01,  5;
      JHPDD2       5      674  0.007418
```

Figure 9-2. Sample Error Log Report Format (Part 9 of 10)

```
TK,SCTR,ER#:  36, 08,    5;
      JHPDD1       10     7277  0.001374
TK,SCTR,ER#:  36, 06,    5; 36, 07,   5;

C21   OS3PCB        2       10  0.2
TK,SCTR,ER#:  16, 01,   2;

- END OF DATA
NOTES: + SIGNS IN TOTAL I/O COUNT INDICATE THAT THE I/O COUNTER HAS WRAPPED.
       NUMBER SHOWN IS MINIMUM AND MAY BE GREATER BY 32768 BYTE INCREMENTS

       ** IN OTHER COUNTS INDICATE THAT ERROR THRESHOLD HAS BEEN EXCEEDED.

       ONLY COUNTERS WITH NON-ZERO VALUES ARE DISPLAYED.
        I/O COUNT INCLUDES THE ON ON WHICH ERROR HAS INCURRED.

OS/3 ONUERL-ERROR LOG EDITOR VERSION 48/29      CUST. ID.: XXXXXXXXX                   yy/mm/dd   hh.mm PAGE XX

SYSTEM DEFINITION FILE

DEVICE ADDRESS        C10
TYPE/FEATURE          WSC
MICROCODE NAME        WSC2-400

DEVICE ADDRESS        C13
TYPE/FEATURE          3560
MICROCODE NAME        WS1DJ000

Note:

Each device in the device table is printed here in this report.

BASE OS/3 RELEASE   -- 14.0.0
```

Figure 9-2. Sample Error Log Report Format (Part 10 of 10)

Table 9-3. CHARACTERISTICS Field Displayed in ONUERL Report

Byte No.	Bit No.	Definition
0	0	Minimum supervisor
	1	Software performance monitor
	2	Online diagnostics
	3	Timer
	4	N/A
	5	Suppress JC58 message display
	6	Suppress R06 message display
	7	Memory consolidation
1	0	Error logging
	1	User files locked
	2	Job accounting
	3	Spooling
	4	Roll out/roll in
	5	Input spooling
	6	Shared code
	7	Tape block numbers
2	0	Console log
	1	Retain console log
	2	File lock = share
	3	File lock = granule
	4	Interactive OS/3
	5	Maximum CPU time configuration
	6	Maximum wall time configuration
	7	Workstation support
3	0	RC symbiont available
	1	CDI supported
	2	DTFs supported
	3	Ignore DM SFTs
	4	Dynamic buffer management
	5	Resident loader
	6	Setime support
	7	Workstation RU can use card RDR

Table 9-4. FLAGS Field Displayed in ONUERL Report

Byte No.	Bit No.	Definition
0	N/A	N/A
1	0	Error log effective
	1	Error log active
	2	Logging machine check HPR
	3	ELOG not recovered at IPL
	4	Exigent machine check to be logged
	5	Retry stack machine check to be logged
	6	Main storage machine check to be logged
	7	Resident ELOG in control

Table 9-5. MSE Display of ONUERL Report

Item	Definition
ADR	Address range in hexadecimal
CNTRL bits	0: ENC (ENable Count) When 1, permits the incrementation of the near-valie count field in this MSE log entry. 1: ENI (ENable Incrementation) When 1, permits the incrementation of the total near-valid count field in control register 13. 2: SR (Source Report) When 1, indicates the near-valid count field in this MSE log entry has been incremented. 3: MS (Multiple Syndrome) When 1, indicates that one or more syndrome changes have been detected while logging into this MSE log entry. 4-6: Reserved 7: CO (Carry Out) When 1, indicates the near-valid count in this MSE log entry has wrapped around ($FFFF_{16}$ to 0000_{16}). Multiple occurrences of wraparound will be merged.
SYND (SYNDrome)	Contains the most recent syndrome recorded, if any, in this MSE log entry. This identifies the byte and bit position (X-coordinate) at which the single-bit failure (corrected) occurred.
COUNT	Contains a count of the occurrences of logging into the particular MSE log entry.

Table 9-6. CPU Retry Stack Log-Out Display of ONUERL Report

Item	Definition
error CLASS	00: Not used 01: Internal CPU 02: COS 03: Main storage 04: M-bus 05: D-bus 06: MSP detected unconnectable ECC error 07: MSP detected error (non-ECC related) 08: I/O structure fault 09: Channel check 0A: Machine check hardware mask (MC bit) set OB-FF: Reserved
RETRY COUNT	Indicates the number of retries that occurred in the process of recovering from the failure described. Exceeding 255 retries results in an exigent machine check.
HyPerViSoR State indicator (currently active slot)	01: DBUS extension 02: DMA 03: MLCM 04: SDMA 05: QUER 06: Maintenance panel 07: S-2
WD2 and WD3	The format of words 2 and 3 depends upon the error class: Error Class 01, 02 Word 3, bits 16-31 indicate the failing microinstruction or COS address Error Class 03, 06, 07 Word 3, bits 0-7 indicate the absolute main storage address. Error Class 04 Words 2 and 3 contain the error report from failing unit.

Table 9-7. Exigent Machine Check Displays

Item	Definition
MCIC	Machine check interrupt code
PSW	Program status word
TOD	Time of day
SPV RGS	Supervisor registers
ED MCIC	External damage MCIC
ED ERO, ER1	External damage registers 0, 1

9.6. Error Log Analysis (ONELAN Program)

The ONELAN program provides error log analysis and functions as part of the ONUERL program. You use ONELAN to display accumulated error data in the form of graphs and tables. Analyzing errors in this way provides you with:

- Equipment deviation
- Equipment trends and performance
- Significant error counts
- Serious equipment error problems

You can obtain this information for individual devices (device error count) or for all devices of a particular type (subsystem error count). The error counts are based on monthly totals. Each graph displays up to 5 months of accumulated errors.

Note: *Before running ONELAN for the first time, ONUERL must be run to properly initialize the $Y$ESUM file used by ONELAN. It is also recommended that ONUERL be run at least twice between executions of ONELAN so that ONELAN produces a more meaningful output.*

Enter the following command to execute the ONELAN program:

```
RVΔONELAN
```

After ONELAN begins to execute, it displays the following message:

```
ONELAN - ERROR LOG ANALYSIS VER. 01/06
- DO YOU NEED HELP RUNNING THIS PROGRAM ? (Y,N)
```

Your answer to this message determines whether you use standard or tutorial mode to communicate your requests to the ONELAN program. You cannot change modes after making a selection unless you reload the entire program.

Enter N for standard mode, Y for tutorial.

9.6.1. Standard Mode Procedure

When standard mode is selected, the program displays the following message:

```
nn   -    ENTER REQUEST (LST,DOC,ALL,DVC,SUB,END,?)
```

The message lists the keywords that identify the options for running ONELAN. It requests you to enter the keywords related to the functions you want performed.

Enter a '?' if you want the program to display definitions for the keywords. The program redisplays the 'ENTER REQUEST' message after listing the definitions.

```
nn?
      -    AVAILABLE REQUESTS:
            LST:  DISPLAY LIST OF AVAILABLE GRAPHS.
            DOC:  PRINT OPERATING INSTRUCTIONS.
            ALL:  PRINT ALL GRAPHS.
            END:  TERMINATE PROGRAM.
            DVC:  PRINT DEVICE GRAPH.
            SUB:  PRINT SUB-SYSTEM GRAPH.
nn    -    ENTER REQUESTS (LST,DOC,ALL,DVC,SUB,END,?)
```

Enter the keywords in any order (freeform) but make certain that they are preceded by the message-id number and separated by commas.

```
nn DVC,SUB
```

Entering the DVC (device graph) keyword displays the following message requesting you to identify the device by channel (C), subchannel (S), and unit (U) address:

```
- ENTER DEVICE ADDRESS (CSU(,CSU)...,?)
```

Enter the addresses. Multiple addresses must be separated by commas. To display a sample address, enter a question mark (?).

Entering the SUB (subsystem graph) keyword displays the following message requesting you to enter the channel (C) and subchannel (S) address for the subsystem:

```
- ENTER SUB-SYSTEM ADDRESS (CS(,CS)...,?)
```

Enter the addresses. Multiple addresses must be separated by commas. To display a sample address, enter a question mark.

After validating the device and subsystem addresses, ONELAN directs the appropriate graphs to the system printer.

When more than one keyword is entered on a line, the program honors them one at a time until there are no outstanding keywords. If the END keyword is omitted, the program redisplays the 'ENTER REQUEST' soliciting you to enter additional keywords. If any keywords are specified after "END", they are not processed.

To terminate ONELAN, enter the END request when the "ENTER REQUEST" message is redisplayed.

A sample ONELAN dialog in standard mode is shown in Figure 9-3.

```
    RV ONELAN
    R03 RUN PROCESSOR SUCCESSFULLY PROCESSED ONELAN                 9:20 RV7962
    JC01  JOB ONELAN   EXECUTING JOB STEP ONUERL00 #001 09:20:17    9:20 ONELAN
    ONELAN-ERROR LOG ANALYSIS VER. xx/xx                            9:20 ONELAN
34   - DO YOU NEED HELP RUNNING THIS PROGRAM ?(Y,N)                 9:20 ONELAN
34   N                                                              9:20
     - LAST RUN ON   yy/mm/dd AT   hh:mm:ss                         9:20 ONELAN
36   - ENTER REQUEST (LST,DOC,ALL,DVC,SUB,END,?)                    9:20 ONELAN
36   ?                                                              9:20
     - AVAILABLE REQUESTS:                                          9:21 ONELAN
       LST: DISPLAY LIST OF AVAILABLE GRAPHS.                       9:21 ONELAN
       DOC: PRINT OPERATING INSTRUCTIONS.                           9:21 ONELAN
       ALL: PRINT ALL GRAPHS.                                       9:21 ONELAN
       END: TERMINATE PROGRAM.                                      9:21 ONELAN
       DVC: PRINT DEVICE GRAPH.                                     9:21 ONELAN
       SUB: PRINT SUB-SYSTEM GRAPH.                                 9:21 ONELAN
11?  - ENTER REQUEST  (LST,DOC,ALL,DVC,SUB,END,?)                   9:21 ONELAN
11   LST,ALL,SUB                                                    9:21
     GRAPHS AVAILABLE: 311,312,313,314,330,332,370,100,101,103      9:21 ONELAN
     104,105,320,321                                                9:21 ONELAN
14   - ENTER SUBSYSTEM ADDRESS (CS(,CS)...,?)                       9:22 ONELAN
14   ?                                                              9:22
     EXAMPLE: 10,37                                                 9:22 ONELAN
16   - ENTER SUBSYSTEM ADDRESS (CS(,CS)...,?)                       9:22 ONELAN
16   31,32,33,10,37                                                 9:22
17   - ENTER REQUEST  (LST,DOC,ALL,DVC,SUB,END,?)                   9:22 ONELAN
17   DVC,END                                                        9:23
18   - ENTER DEVICE ADDRESS (CSU(,CSU)...,?)                        9:23 ONELAN
18   ?                                                              9:23
     EXAMPLE: 100,101,370                                           9:23 ONELAN
20   - ENTER DEVICE ADDRESS (CSU(,CSU)...,?)                        9:23 ONELAN
20   103,104,321,END                                                9:24
     JC02   ONELAN   TERMINATED NORMALLY                09:24:34    9:24 ONELAN
```

Figure 9-3. Sample ONELAN Dialog, Standard Mode

9.6.2. Tutorial Mode Procedure

When tutorial mode is selected, ONELAN displays a series of prompts asking you to define the operations you want it to perform. The prompts are displayed one at a time with each prompt requiring a response before the next one can be displayed. A response of N to any prompt causes the program to display the next prompt in the sequence. A response of Y causes the appropriate graph/display to occur before ONELAN displays the next prompt.

The order in which ONELAN displays the prompts is as follows:

```
DO YOU NEED OPERATING INSTRUCTIONS ?  (Y OR N)
```

Enter a Y response if you want ONELAN to print the operating instructions on your system printer. Enter N to terminate this request and display the next prompt.

```
DO YOU WANT A LIST OF AVAILABLE GRAPHS ? (Y OR N)
```

Enter a Y response if you want ONELAN to display, at the console, a list of all device address for which there is a graph. Enter N to terminate this request and display the next prompt.

```
DO YOU WANT ALL DEVICES GRAPHED ? (Y OR N)
```

Enter a Y response if you want ONELAN to print individual graphs for all devices on your system printer. Enter N to terminate this request and display the next prompt.

```
DO YOU WANT SUBSYSTEM GRAPH(S) ? (Y OR N)
```

Enter a Y response if you want ONELAN to print graphs for one or more subsystems.

ONELAN now displays the following message requesting you to specify the required channel and subchannel address for each subsystem.

```
ENTER SUBSYSTEM ADDRESS (CS(,CS)...,?)
```

Enter the channel and subchannel addresses of the devices you want graphed. Multiple addresses must be separated by commas.

If you want to see a sample address, enter a question mark (?). ONELAN displays the sample address and then redisplays the sub-system address request.

```
?
     EXAMPLE: 10,37
ENTER SUBSYSTEM ADDRESS (CS(,CS)...,?)
```

After you respond with the addresses, ONELAN asks you if you want device graphs.

```
DO YOU WANT DEVICE(S) GRAPH(S) ? (Y OR N)
```

Enter a Y response if you want ONELAN to print individual graphs for specific devices.

ONELAN now displays the following message requesting you to specify the required channel, subchannel, and unit address for each device.

```
ENTER DEVICE ADDRESS (CSU(,CSU)...,?)
```

Enter the channel, subchannel, and unit address of the device you want graphed. Multiple addresses must be separated by commas.

If you want to see an address example, enter a question mark (?). ONELAN displays the sample address and then redisplays the device address request.

```
?
     EXAMPLE: 100,101,370
ENTER DEVICE ADDRESS (CSU(,CSU)...,?)
```

After you enter the addresses (or if you answered N to the device graph request), ONELAN displays the following request:

```
TERMINATE PROGRAM ? (Y OR N)
```

Enter a N response of you want to print more graphs. (ONELAN redisplays the "Enter subsystem request.)

Enter a Y response to terminate ONELAN.

A sample ONELAN dialog in tutorial mode is shown in Figure 9-4.

```
      RV ONELAN
      R03 RUN PROCESSOR SUCCESSFULLY PROCESSED ONELAN          9:20 RV7962
      JC02  JOB ONELAN   EXECUTING JOB STEP ONUERL00 #001 09:20:17   9:20 ONELAN
      ONELAN-ERROR LOG ANALYSIS VER. 01/06                     9:20 ONELAN
 11?  - DO YOU NEED HELP RUNNING THIS PROGRAM ?(Y,N)           9:20 ONELAN
 11   Y                                                        9:20
 13?  - DO YOU NEED OPERATING INSTRUCTIONS ?(Y OR N)           9:20 ONELAN
 13   N                                                        9:21
 14?  - DO YOU WANT A LIST OF AVAILABLE GRAPHS ?(Y OR N)       9:21 ONELAN
 14   Y                                                        9:21
 15   - GRAPHS AVAILABLE: 330,370,100                          9:21 ONELAN
 16?  - DO YOU WANT ALL DEVICES GRAPHED ?(Y OR N)              9:21 ONELAN
 16   Y                                                        9:21
 17?  - DO YOU WANT SUBSYSTEM GRAPH(S) ?(Y OR N)               9:22 ONELAN
 17   Y                                                        9:22
 18?  - ENTER SUBSYSTEM ADDRESS (CS(,CS)...,?)                 9:22 ONELAN
 18   ?                                                        9:22
 19   EXAMPLE: 10,37                                           9:22 ONELAN
```

Figure 9-4. Sample ONELAN Dialog, Tutorial Mode (Part 1 of 2)

```
20?  - ENTER SUBSYSTEM ADDRESS (CS(,CS)...,?)                   9:22 ONELAN
20   37,33,10                                                   9:22
21?  - DO YOU WANT DEVICE(S) GRAPH(S) ?(Y OR N)                 9:23 ONELAN
21   Y                                                          9:23
24?  - ENTER DEVICE ADDRESS (CSU(,CSU)...,?)                    9:23 ONELAN
24   ?                                                          9:23
25   - EXAMPLE: 100,101,370                                     9:23 ONELAN
26?  - ENTER DEVICE ADDRESS (CSU(,CSU)...,?)                    9:23 ONELAN
26   100                                                        9:23
27?  - TERMINATE PROGRAM ?(Y OR N)                              9:23 ONELAN
27   N                                                          9:24
28?  - DO YOU WANT SUBSYSTEM GRAPH(S) ?(Y OR N)                 9:24 ONELAN
28   N                                                          9:24
29?  - DO YOU WANT DEVICE(S) GRAPH(S)  ?(Y OR N)                9:24 ONELAN
29   Y                                                          9:24
30?  - ENTER DEVICE ADDRESS (CSU(,CSU)...,?)                    9:24 ONELAN
30   370                                                        9:24
31?  - TERMINATE PROGRAM ?(Y OR N)                              9:24 ONELAN
31    Y                                                         9:24
32   JC02  JOB ONELAN    TERMINATED NORMALLY          09:24:47  9:24 ONELAN
```

Figure 9-4. Sample ONELAN Dialog, Tutorial Mode (Part 2 of 2)

ONELAN Messages

The following console workstation or console messages may be displayed during execution of the ONELAN program:

`INVALID ENTRY`

The request entered is invalid, probably due to incorrect spelling. The 'ENTER REQUEST' message will be redisplayed. Reenter the request.

`TOO MANY ENTRIES. REQUEST TRUNCATED.`

The request buffer cannot accommodate the number of requests entered. The leftover requests must be reentered when the accepted requests have been honored.

`NO RECORD AVAILABLE FOR DVC xxx.`

A request was made for a graph for the specified device, but there is no summary data for that device. The request is ignored.

`$Y$ESUM nnnn error. CD$FNMC: sense data`

An error was detected during accessing of $Y$ESUM, where nnnn is OPEN, CLOSE, INPUT, or OUTPUT. Sense data is returned by the system. ONELAN terminates.

Figures 9-5 and 9-6 show sample device error log analysis graphs for a 3561 workstation (address 312) and a 8420 diskette (address 321), respectively. The graph on the left is the device error count for the months of November and December. The graph on the right shows the ratio of the total incident count (device error count) to the total output count (TIC/TOC ratio). The exact error counts and ratios are given in the table at the bottom of each display.

Note: *As a new month's counts become available, the graphs will expand to the right until 5 months are shown. Thereafter, the leftmost month is truncated automatically.*

The following dialog was used to obtain the displays shown in Figures 9-5 and 9-6:

```
ENTER REQUEST (LST,DOC,ALL,DVC,SUB,END,?)
DVC,END
ENTER DEVICE ADDRESS (CSU(,CSU)...,?)
312,321
```

A sample subsystem error log analysis for disk is provided in Figure 9-7. The device error count and TIC/TOC ratio graphs show the device addresses for all diskettes present on the system and charts their errors for the months of November and December, with each month appearing on a separate page. The exact totals and TIC/TOC ratios are given in the table at the bottom of each display. The third part of the subsystem error log analysis charts the total subsystem errors by month. The fourth part of the subsystem error log analysis summarizes the disk error totals by function for each month.

The following dialog was used to obtain the subsystem error log analysis graphs for disk:

```
ONELAN - ENTER REQUEST (LST,DOC,ALL,DVC,SUB,END,?)
SUB
ONELAN - ENTER SUBSYSTEM ADDRESS (CS,(,CS)...,?)
10
```

When the ENTER REQUEST message reappears, you may request additional displays or END.

```
 ONELAN-ERROR LOG ANALYSIS. VER. xx/xx.                 yy/mm/dd  hh.mm  PAGE n
                    SYSTEM-80 - 3561 WK-STA    ADDRESS: 0312
       DEVICE ERROR COUNT                         DEVICE TIC/TOC RATIO
       250 -                                            1.0 -
           I                                                I
           I                                                I
           I                                                I
           I                                                I X
       200 -                                             .1 - X
           I                                                I X
           I                                                I X
           I                                                I X
           I                                                I X
       150 -                                           .01 - X
ERROR      I X                                  TIC/TOC     I X
           I X                                              I X
COUNT      I X                                   RATIO      I X
           I X                                              I X
       100 - X                                         .001 - X
           I X                                              I X
           I X                                              I X
           I X                                              I X
           I X                                              I X
        50 - X                                        .0001 - X
           I X                                              I X
           I X                                              I X
           I X                                              I X
           I X                                              I X
         0 - X    X                                  .00001 - X    X
           --I----I--                                       --I----I--
            NOV  DEC                                         NOV  DEC
                     MONTH       NOV        DEC
                     TIC           140         0
                     TOC           506         3
                     TIC/TOC:  0.27668    0.
```

Figure 9-5. Device Error Log Analysis for 3561 Workstation

```
 ONELAN-ERROR LOG ANALYSIS. VER. xx/xx.                 yy/mm/dd  hh.mm  PAGE n
                    SYSTEM-80 - 8420           ADDRESS: 0312
       DEVICE ERROR COUNT                          DEVICE TIC/TOC RATIO
       250 -                                            1.0 -
           I                                                I
           I                                                I
           I                                                I
           I                                                I
       200 -                                             .1 -
           I                                                I
           I                                                I
           I                                                I
           I                                                I
       150 -                                            .01 - X    X
ERROR      I X                                  TIC/TOC     I X    X
           I X                                              I X    X
COUNT      I X                                   RATIO      I X    X
           I X                                              I X    X
       100 - X                                         .001 - X    X
           I X                                              I X    X
           I X                                              I X    X
           I X                                              I X    X
           I X                                              I X    X
        50 - X                                        .0001 - X    X
           I X                                              I X    X
           I X                                              I X    X
           I X                                              I X    X
           I X                                              I X    X
         0 - X    X                                  .00001 - X    X
           --I----I--                                       --I----I--
            NOV  DEC                                         NOV  DEC
                     MONTH       NOV        DEC
                     TIC           150          43
                     TOC         15049        4357
                     TIC/TOC:  0.009967   0.009869
```

Figure 9-6. Device Error Log Analysis for 8420 Diskette

```
 ONELAN-ERROR LOG ANALYSIS. VER. xx/xx.                 yy/mm/dd  hh.mm  PAGE n
                        SYSTEM 80 -  DISK SUB-SYSTEM.
       DEVICE ERROR COUNT            MONTH:  DEC     DEVICE TIC/TOC RATIO
       250 -                                       1.0 -
           I                                           I
           I                                           I
           I                                           I
           I                                           I
       200 -                                        .1 -
           I                                           I
           I                                           I
           I                                           I
           I                                           I
       150 -                                       .01 -
ERROR      I                               TIC/TOC     I
           I                                           I
COUNT      I                                RATIO      I                 X
           I                                           I                 X
       100 -                                      .001 -                 X
           I                                           I                 X
           I                                           I                 X
           I                                           I                 X
           I                                           I             X   X
        50 -                                     .0001 -             X   X
           I                                           I             X   X
           I                 X                         I             X   X
           I                 X                         I             X   X
           I             X   X                         I             X   X
         0 -             X   X       X          .00001 -             X   X       X
           --I---I---I---I---I---I---I---I-            --I---I---I---I---I---I---I---I-
            105 106 107 100 101 102 103 104             105 106 107 100 101 102 103 104

DVC ID.    105        106      107        100           101       102     103
TIC             0         0         0        10          26          0       0
TOC             0         0         0     86236        5339          0       0
TIC/TOC: 0.         0.        0.        0.000116    0.00487    0.        0.       0.
```

Figure 9-7. Disk Subsystem Error Log Analysis (Part 1 of 3)

```
 ONELAN-ERROR LOG ANALYSIS. VER. xx/xx.                 yy/mm/dd  hh.mm  PAGE n
                        SYSTEM 80 -  DISK SUB-SYSTEM.
       DEVICE ERROR COUNT           MONTH:  NOV              DEVICE TIC/TOC RATIO
       250 -                         X              1.0 -                 X
           I                         X                  I                 X
           I                         X                  I                 X
           I                         X                  I                 X
           I                         X                  I                 X
       200 -                         X               .1 -                 X           X
           I                         X                  I                 X           X
           I                         X                  I                 X           X
           I                         X                  I                 X           X
           I                         X                  I                 X           X
       150 -                         X              .01 -                 X           X
ERROR      I                         X      TIC/TOC     I                 X           X
           I                         X                  I                 X           X
COUNT      I                         X       RATIO      I                 X       X   X
           I             X           X                  I                 X       X   X
       100 -             X           X             .001 -                 X       X   X
           I             X           X                  I                 X       X   X
           I             X           X                  I                 X       X   X
           I             X           X                  I                 X       X   X
           I             X           X                  I             X   X       X   X
        50 -             X           X            .0001 -             X   X       X   X
           I             X           X                  I             X   X       X   X
           I             X           X                  I             X   X       X   X
           I             X           X                  I             X   X       X   X
           I             X           X                  I             X   X       X   X
         0 -             X   X       X           .00001 -             X   X       X   X
           --I---I---I---I---I---I---I---I-            --I---I---I---I---I---I---I---I-
            105 106 107 100 101 102 103 104              105 106 107 100 101 102 103 104

DVC ID.    105        106      107        100         101    102       103
TIC             0         0         0       105         3      0      366
TOC             0         0         0    413005         2       0   103354
TIC/TOC: 0.         0.        0.         0.000254   0.      0.         0.003541
```

Figure 9-7. Disk Subsystem Error Log Analysis (Part 2 of 3)

```
 ONELAN-ERROR LOG ANALYSIS. VER. xx/xx.                 yy/mm/dd  hh.mm  PAGE n
                        SYSTEM-80      DISK SUB-SYSTEM.
                        *........................*
                        **  SUB-SYSTEM TOTALS  **
                        *........................*
            SUB-SYSTEM ERROR COUNT                SUB-SYSTEM TIC/TOC RATIO
      1250 -                                        1.0 -
           I                                            I
           I                                            I
           I                                            I
           I                                            I
      1000 -                                         .1 -
ERROR      I                                            I
           I                                TIC/TOC     I
           I                                 RATIO      I
           I                                            I
       750 -                                        .01 -
COUNT      I                                            I
           I                                            I
           I                                            I
           I                                            I
       500 -                                       .001 -
           I                                            I
           I                                            I X
           I                                            I X   X
           I         X                                  I X   X
       250 - X       X                            .0001 - X   X           X
           I X       X   X                              I X   X           X
           I X   X   X   X                              I X   X   X       X
           I X   X   X   X   X                          I X   X   X   X   X
           I X   X   X   X   X                          I X   X   X   X   X
         0 - X   X   X   X   X                   .00001 - X   X   X   X   X
           --I---I---I---I---I                          --I---I---I---I---I
            NOV DEC JAN FEB MAR                          NOV DEC JAN FEB MAR

MONTH        NOV    DEC    JAN    FEB    MAR
TIC           XX     XX     XX     XX     XX
TOC          XXX    XXX    XXX    XXX    XXX
TIC/TOC:   .XXXX  .XXXX  .XXXX  .XXXX  .XXXX
```

Figure 9-7. Disk Subsystem Error Log Analysis (Part 3 of 3)

9.7. Sending Error Data to Unisys Support Center

ONUERL does not automatically transmit error summary data to a Unisys support center. Instead, if error threshold has been exceeded or you've chosen the SUMMARY=YES option, ONUERL writes the error summary data to the system journal file. You can later transfer this information to the support center.

You must establish initial contact with the support center. Instructions for transferring data to the support center as provided in Section 10.

Section 10
Error Data Handling

10.1. Contacting the Unisys Support Center

Contact the support center whenever you need software or hardware assistance from Unisys. The support center expects to receive calls from the system administrator or another authorized person.

It is important to call as soon as possible after you encounter a problem so that you can save needed information by using appropriate dump procedures.

10.1.1. How to Contact the Support Center

In the continental United States, you call the following toll-free number to contact the support center:

1-800-328-0440

International customers should contact their Unisys subsidiary.

10.1.2. What to Do before Calling the Support Center

Before calling, consult all available documentation, such as:

- Programming or operating guides and reference manuals
- System messages manual
- Software release announcements (SRAs)

These manuals may answer your questions, or provide you with a solution to your problem.

When you do call, be prepared to provide the following types of information:

- Your name and telephone number
- The name of your company
- Your software release level
- Your SMP level and SMCs beyond your SMP level
- SYSGEN, ICAM, and IMS generation information
- Console log data
- Error messages

If you have already called the center about a software problem, give the contact number, the date, or the name of the representative who handled your previous call, as well as the results of the corrective actions previously recommended.

If your hardware problem is a checkstop or hardware-related HPR, you may be requested to make a remote maintenance connection. Have the dataset, modem, or acoustic coupler powered up and ready for immediate use.

10.1.3. When You Call

When you call the support center in the continental United States, you are prompted for a call routing code and other preliminary routing information. If you use a push-button telephone, enter the requested codes by pushing the appropriate buttons on the telephone. (Calls from rotary phones are automatically intercepted by a receptionist.)

When prompted, enter the codes that identify the model of your System 80 and whether you need hardware or software support.

After this preliminary information is entered, a receptionist will verify your service entitlement and then direct your call to the appropriate support center specialist.

10.2. The User Communication Form

The User Communication Form (UCF) is the Unisys form used to record the information needed to correct a problem. UCFs are created electronically by support center representatives.

When you call the support center with a problem that has not already been reported, the support representative creates a UCF by using the information you provide on the telephone. The completed UCF is routed to the OS/3 development center for analysis and resolution.

When the UCF is created, it is assigned an authorization number. Refer to this number whenever you call to ask about that problem or to check on the current status of your UCF.

During the UCF process, you may be asked to send additional documentation describing the suspected problem. Label any documentation you send to Unisys with:

- UCF authorization number
- Unisys customer number (assigned by Unisys)
- Company name
- Host system (processor)
- Operating system
- O.S. level
- Materials description identifying the kind of documentation you are sending (tape label, device type, etc.)
- Detailed description of the documentation you are sending and any new information that was not apparent when you reported the problem

This additional documentation should be sent to:

Unisys Corporation
OS/3 Systems Development
2476 Swedesford Road
Paoli, PA. USA 19301-0203
Attention: UCF Coordinator

10.3. Sending Error Log Data

Use the following procedure to send error data to the Unisys support center.

1. Power on modem, if not already powered on.
2. Press the **ESC** and **C** keys to display the configuration frame.
3. Press the **E** and then the **XMIT** keys to display the configuration select frame.
4. Key in **SCP** and then press the XMIT key to enter the first part of the user ID.
5. Key in **J3280404** and press **XMIT** key to enter second part of the user ID.
6. Key in **U0** and press **XMIT** key to establish modem as remote facility.
7. Key in **Z1** and press **XMIT** key to set baud rate at 1200 bps. (**Z0** for 300 bps.)
8. Press the **N** and then **XMIT** keys to turn the remote console on.
9. Key in **M1** and press **Enter** key to place console in master-master mode.
10. Press the **HS** and **TLK** switches on modem (MC indicator lights).
11. Call support center personnel (**1-800-328-0440**).
12. Enter **1** to complete the call at the prompt.
13. Enter your 3-digit prompt (system) code **050**.
14. Enter **1** for hardware or **2** for software at the prompt.
15. Support center personnel will guide you through the remote session.
16. Support center personnel will terminate the session.
17. After session is terminated, return the console to system mode and turn off the modem as follows:

    - Press the **ESC** and **C** keys to display the configuration frame.
    - Press **E** and then the **XMIT** key to display the configuration select frame.
    - Enter **M0** and then press the **XMIT** key to return to system mode.
    - Press **O** and then press the **XMIT** key to turn the moden off.

Appendix A
Supervisor Modification Procedure

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

Note: *For a description of system output messages and how to respond to them (via solicited input messages), see Section 4.*

Table A-1. How to Modify List of Resident Shared Code Modules

IPL Output Message	Operator Response	
IS THE SHARED CODE DIRECTORY INDEX TO BE BUILT? (Y,N)	Y	Specifies an index is to be built for the shared code library. The system will do 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.
	N	No shared code directory index is to be built.
ANY RESIDENT SHARED MODULES TO ADD OR DELETE? (Y,N)	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 A-2.
	Y	Specifies SYSGEN list of shared code modules requires modification. Output message requesting a function is displayed.
FUNCTION? (ADD,DEL,LIST,END,HELP)	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 descried in Table A-2.
	HELP	Displays brief description of the ADD, DEL, LIST, and END functions, followed by an output message requesting another function.
	LIST	Displays a list of the individual shared code modules currently specified to be made resident, followed by an output message requesting another function.

continued

Table A-1. How to Modify List of Resident Shared Code Modules (cont.)

IPL Output Message	Operator Response

	{ADD
	DEL}
	Specifies an individual module or group of modules is to be added to or deleted 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   - DIALOG PROCESSOR`
	`RPG  - RPG EDITOR`
	`DDPR - REMOTE DISTRIBUTED DATA`
	`PROCESSING`
	`DDPL - LOCAL DISTRIBUTED DATA`
	`PROCESSING`
	`ESC  - ESCORT`
	`ESCF - ESCORT FULL`
	`BAS  - BASIC`

`WHICH GROUP TO` {`ADD?`
`DELETE?`}
	{group-name
	module-name} [,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 job stream SCLIST to get a list of all system shared code modules filed in $Y$CLOD.

Table A-2. How to Modify Buffer Sizes Used by Dynamic Buffer Management

IPL Output Message		Operator Response
ANY CHANGES TO DYNAMIC BUFFER MANAGEMENT PARAMETERS? (Y,N)	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 the interactive IPL procedure in Section 2.
	Y	Specifies SYSGEN=specified expansion region and resident buffer sizes require modification. Output message requesting new expansion region size is displayed.
ENTER NEW EXPANSION REGION SIZE? (CURRENTLY decimal-byte-size)	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 the interactive IPL procedure in Section 2.
	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.
ENTER NEW RESIDENT BUFFER SIZE? (CURRENTLY decimal-byte-size)	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 the interactive IPL procedure in Section 2.

*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
200 k	is 200 x 1024 = 204,800 decimal bytes

Index

User Reply Form

Help Us To Help You

Document Title

Document Number Date

We are interested in your comments and suggestions regarding this document. We will use them to improve the quality of this product information. Please check the type of suggestion.

□ Addition □ Deletion □ Revision □ General Comment

Recommended Change (Please identify each affected item, such as page, paragraph, figure, or table.)

Name Telephone Number ( )

Title Company

Address (Street, City, State, ZIP)

User Reply Form

Help Us To Help You

Document Title

Document Number Date

We are interested in your comments and suggestions regarding this document. We will use them to improve the quality of this product information. Please check the type of suggestion.

□ Addition □ Deletion □ Revision □ General Comment

Recommended Change (Please identify each affected item, such as page, paragraph, figure, or table.)

Name Telephone Number ( )

Title Company

Address (Street, City, State, ZIP)

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS MAIL PERMIT NO. 72 PAOLI, PA

POSTAGE WILL BE PAID BY ADDRESSEE

Unisys Corporation
SPG East Coast
Systems Documentation Development
Tredyffrin Plant
2476 Swedesford Road
PO BOX 203
PAOLI, PA 19301-9898

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS MAIL PERMIT NO. 72 PAOLI, PA

POSTAGE WILL BE PAID BY ADDRESSEE

Unisys Corporation
SPG East Coast
Systems Documentation Development
Tredyffrin Plant
2476 Swedesford Road
PO BOX 203
PAOLI, PA 19301-9898