GC38-0229-1
File No. S370-40

Systems

Operator's Library: OS/VS2 MVS System Commands

VS2 Release 3.8

Includes Selectable Units:

Supervisor Performance #2	VS2.03.807
TSO/VTAM Level 1	VS2.03.813
Service Data Improvements	VS2.03.817
Dumping Improvements	5752-833
Attached Processor System for Models 158/168	5752-847
MVS Processor Support	5752-851
Hardware Recovery Enhancements	5752-855
TSO/VTAM Level 2	5752-858

IBM

Second Edition (July 1978)

This is a major revision of, and obsoletes, GC38-0229-0 and Technical Newsletter GN28-2861 and incorporates changes released in the following Selectable Unit Newsletters and System Library Supplements:

Supervisor Performance 2	VS2.03.807	GN25-0131	(dated May 28, 1976)
TSO/VTAM Level 1	VS2.03.813	GN25-0126	(dated May 28, 1976)
Service Data Improvements	VS2.03.817	GN25-0137	(dated July 30, 1976)
Dumping Improvements	5752-833	GC38-1026	(dated March 30, 1977)
Attached Processor System for Models 158/168	5752-847	GC38-1033	(dated April 29, 1977)
MVS Processor Support	5752-851	GD23-0032	(dated November 30, 1977)
Hardware Recovery Enhancements	5752-855	GC38-1038	(dated May 31, 1977)
TSO/VTAM Level 2	5752-858	GD23-0051	(dated September 30, 1977)

This revision also incorporates the information previously described in two other publications - *Operator's Library: OS/VS2 Console Configurations*, GC38-0120, and *Operator's Library: OS/VS2 Display Consoles*, GC38-0260, which are now obsolete.

This edition with Technical Newsletter GN28-2997 applies to release 3.8 of OS/VS2 and to all subsequent releases until otherwise indicated in new editions or Technical Newsletters. Changes are continually made to the information herein; before using this publication in connection with the operation of IBM systems, consult the latest IBM **System/370 Bibliography**, GC20-0001, for the editions that are applicable and current.

Preface

This publication describes the basic system functions and commands used by a system console operator in running a System/370 under OS/VS2 MVS. Subsystem (JES2 or JES3) commands - which can also be used to perform many of the same functions - are documented separately in specific subsystem publications; refer to *Operator's Library: OS/VS2 MVS JES2 Commands*, GC23-0007, *Operator's Library: Network Job Entry Facility For JES2 Commands*, GC23-0011, or *Operator's Library: OS/VS2 MVS JES3 Commands*, GC23-0008. In some cases, the specific use of the system commands varies depending upon which subsystem (JES2 or JES3) is in operation. These variations are described in this publication. Throughout this book, references to system programming library publications are provided for the system programmer seeking more information about specific operations.

This book is intended for computer operators and system programmers who have the responsibility for controlling and operating an OS/VS2 MVS system. It assumes that a user of this book is familiar with the hardware controls and features of his specific hardware configuration. It also assumes that the user has a basic understanding of OS/VS2 MVS and its functions. Additional OS/VS2 MVS information can be obtained from the publications listed below.

Note: The system activity measurement facility (MF/1) and the dynamic support system (DSS) are not supported with MVS/System Extensions (program number 5740-XE1).

This publication is organized as follows:

- Chapter 1, "Introduction," provides an overview of multiprocessing characteristics, initializing the system, controlling the system, recording system information, and using dynamic device reconfiguration.
- Chapter 2, "Console Configurations" describes the characteristics of consoles used in the multiple console environment of OS/VS2 MVS and presents guidelines for an operator in a multiple console environment.
- Chapter 3, "Display Consoles" provides characteristics and functional information necessary to use and control display consoles.
- Chapter 4, "The shared DASD Option" describes the shared DASD option and its use in a multiple processor environment.
- Chapter 5, "System Commands" lists all system commands in alphabetical order with their syntax and describes the functions that a command performs. The command syntax and format are presented in this chapter.
- Appendix A, "System Command Summary" is provided for quick reference and retrieval of the functions performed by each of the system commands.
- Appendix B, "Model 168 Console" describes the physical characteristics of the 3066 display console on the Model 168.
- Appendix C, "2250 Display Console" describes the physical characteristics of the 2250 display console.
- Appendix D, "2260 Display Console" describes the physical characteristics of the 2260 display console.
- Appendix E, "3036, 3056, 3158, and 3277 Display Console" describes the physical characteristics of the 3036, 3056, 3158, and 3277 display consoles.
- Appendix F, "Printer-Keyboard Consoles" describes the physical characteristics of printer-keyboard consoles.
- Appendix G, "2740 Communications Terminal" describes the physical characteristics of the 2740 communications terminal.

The glossary at the end of this book defines the technical terms used in this publication.

When a command is mentioned in any part of the text, its syntax and functions are described in Chapter 5, "System Commands"

Associated Publications

The following publications should be used in conjunction with this manual:

Operator's Library: OS/VS2 MVS JES2 Commands, GC23-0007
Operator's Library: Network Job Entry Facility for JES2 Commands, GC23-0011
Operator's Library: OS/VS2 MVS JES3 Commands, GC23-0008
OS/VS Message Library: VS2 System Messages, GC38-1002
OS/VS Message Library: OS/VS2 System Codes, GC38-1008
OS/VS Message Library: VS2 Routing and Descriptor Codes, GC38-1102
OS/VS2 System Programming Library: Initialization and Tuning Guide, GC28-0681
OS/VS2 System Programming Library: JES2, GC23-0002
OS/VS2 System Programming Library: JES3, GC28-0608
OS/VS2 System Programming Library: Job Management, GC28-0627
OS/VS2 System Programming Library: Service Aids, GC28-0674
OS/VS2 System Programming Library: Supervisor, GC28-0628
OS/VS2 System Programming Library: SYS1.LOGREC Error Recording, GC28-0677
OS/VS2 System Programming Library: System Generation Reference, GC26-3792
OS/VS2 System Programming Library: TSO, GC28-0729
OS/VS2 System Programming Library: MVS Diagnostic Techniques, GC28-0725
OS/VS2 System Programming Library: System Management Facilities, GC28-0706
OS/VS2 MVS Attached Processor System for Models 158/168 System Information, GC28-0840
OS/VS2 JCL, GC28-0692
Operator's Library: OS/VS2 TCAM, GC30-3037

In addition, the operator's library contains publications that should be selected based upon your installation's specific system configuration:

OS/VS2 Multiprocessing: An Introduction and Guide to Writing Operating and Recovery Procedures, GC28-0952
Operator's Library: System/370 Model 155 Operating Procedures, GA22-6966
Operator's Library: System/370 Model 165 Operating Procedures, GA22-6969
Operator's Library: VTAM Network Operating Procedures, GC27-0027
Operator's Library: OS/VS2 TCAM, GC30-2046
Operator's Library: IBM 3850 Mass Storage System (MSS) under OS/VS, GC35-0014
Operator's Library: System/370 Model 145 Operating Procedures, GC38-0015
Operator's Library: System/370 Model 168 Operating Procedures, GC38-0030
Operator's Library: IBM 3031 Operator's Guide, GC38-0033
Operator's Library: IBM 3032 Operator's Guide, GC38-0032
Operator's Library: IBM 3033 Operator's Guide, GC38-0031
IBM 2740/2741 Communications Terminal Operator's Guide, GA27-3001

Contents

Illustrations

Summary of Amendments for GC38-0229-1 OS/VS2 Release 3.7 and listed Selectable Units

The information in this edition of *Operator's Library: OS/VS2 MVS System Commands* is presented in a different format from the previous edition. The material has been completely reorganized. Refer to the preface for a description of the new content and organization. In addition to the reorganization, this edition includes:

- The incorporation of these Selectable Unit Newsletters and System Library Supplements:
  - Supervisor Performance #2, SU7 GN25-0131
  - TSO/VTAM Level 1, SU13 GN25-0126
  - Service Data Improvements, SU17 GN25-0137
  - Dumping Improvements, SU33 GC38-1033
  - Attached Processor System for Models 158/168, SU47 GC38-1033
  - MVS Processor Support, SU51 GD23-0032
  - Hardware Recovery Enhancements, SU55 GC38-1038
  - TSO/VTAM Level 2, SU58 GD23-0051
- The incorporation of maintenance and service changes
- The incorporation of *Operator's Library: Display Consoles* and *Operator's Library: OS/VS2 Console Configurations* which are obsoleted with this edition of *Operator's Library: OS/VS2 MVS System Commands*

Chapter 1. Introduction

OS/VS2 MVS is an operating system designed to make effective use of system resources. In conjunction with either the JES2 or JES3 job entry subsystems, MVS provides the means of controlling jobs, teleprocessing networks, and system monitoring and accounting functions to ensure maximum system throughput.

Using system commands and either JES2 or JES3 commands, the system operator can control, modify, and monitor system and job functions. Some functions require the use of a JES command, other functions require the use of a system command, and still other functions can be performed with either a JES command or a system command. Chapter 5, "System Commands," describes how to use the system commands. The JES subsystem commands are described in the following publications:

- *Operator's Library: OS/VS2 MVS JES2 Commands*
- *Operator's Library: OS/VS2 MVS JES3 Commands*
- *Operator's Library: Network Job Entry Facility for JES2 Commands*

To acquaint the operator with the use of system commands, this chapter presents high-level information about the MVS functions that the operator can control. In addition to a brief description of multiprocessing characteristics and guidelines, the system functions described here include:

- Starting, quiescing, and stopping the system
- Allocating and controlling configurations and devices
- Controlling the system
- Controlling jobs
- Controlling the external writer
- Controlling time sharing
- Recording system information
- Dynamic device reconfiguration

Multiprocessing Characteristics and Guidelines

OS/VS2 MVS is capable of running in uniprocessor (single processor) or multiprocessor (multiple processor) mode. The MVS system commands are the same in either mode. Some commands, such as VARY command, have parameters or keywords that apply only to a multiprocessing environment. Chapter 5 describes these parameters and keywords under the appropriate command and points out their use in a multiprocessing system.

Multiprocessing (MP) provides for the parallel processing of jobs by two or more processors. Multiprocessing systems can be either loosely-coupled or tightly-coupled.

Loosely-coupled MP systems pass control information and data between two or more processors via the channel-to-channel adapter. Each processor runs under its own control program.

Tightly-coupled MP systems have two processors that are physically connected to each other, that share main storage and I/O resources, and that run under a single control program.

MVS with JES2 supports uniprocessing and tightly-coupled multiprocessing.

MVS with JES3 supports uniprocessing and both loosely-coupled and tightly-coupled multiprocessing.

Loosely-Coupled Multiprocessing Guidelines

An MVS system with JES3 supports up to eight processors connected by channel-to-channel adapters. In a loosely-coupled MP system, additional ASP (asymetric multiprocessing system) processors can be added. MVS with JES3 can support a combined total of thirty-two (32) processors in an MP system.

JES3 has its own address space in each processor. The processor that actively controls the entire MP system is called the global processor. The remaining processors are called local processors, if physically attached, and ASP processors, if ASP is used in a loosely-coupled system.

Any processor that is capable of executing jobs is called a main processor. In any JES3 MP system there is one global main processor. Additionally, there can be one or more local main processors and one or more ASP main processors. If the global processor fails, JES3 designates any properly-configured processor in the system as the new global processor. This facility is called the dynamic system interchange (DSI).

Tightly Coupled Multiprocessing Guidelines

A tightly-coupled multiprocessing system is composed of two Model 158 or 168 processors operating simultaneously under one control program while sharing system resources. As the term "tightly coupled" implies, these processors are physically attached to one another. (Tightly coupled multiprocessing is not to be confused with the multi-access spool feature of JES2 or with JES3 loosely coupled multiprocessing systems.)

The ability to reconfigure the system as required is a major characteristic of a tightly-coupled multiprocessing system. Reconfiguration is the changing of the type or quantity of resources available to a system. You can reconfigure a multiprocessing system either logically or physically.

Physical reconfiguration for processor, real storage, and, usually, paths to shared I/O, is accomplished through the configuration control panel. The procedures for entering a valid configuration appear in the functional characteristics manual for your model processor. When you physically reconfigure storage, you must assign contiguous storage from location 0 up to at least 512K, because the nucleus must be loaded in this section of storage.

Logical reconfiguration is accomplished by the VARY command. You can dynamically add or delete a processor, storage, a channel, a path to a device, or a device. Refer to "Placing Resources Online and Offline," in Chapter 5, under the VARY command. While logical reconfiguration can be done without physical reconfiguration, logical reconfiguration must accompany physical reconfiguration.

When you enter a VARY CPU,OFFLINE command, the system waits up to three minutes for all I/O activity to complete so the processor can be put offline. If the I/O activity has not completed after five minutes, you receive the following message:

```
* id IEE717D ddd BUSY, REPLY EXTEND OR CANCEL
```

Reply EXTEND if you want the command processor to wait up to three minutes and reissue the message if the I/O still has not completed. Reply CANCEL if you want the system to cancel the VARY CPU command and continue processing.

Operating a multiprocessing system is essentially the same as operating a uniprocessing system. However, the time-of-day (TOD) clocks must be synchronous. When you initialize a multiprocessing system in which the clocks are not synchronized, you receive a message listing the TOD clock setting for each processor with a set clock. You must reply with either the address of the processor with the desired setting or the time to be set for both processors with a set clock. Refer to "Setting the Time-of-Day Clock and Specifying the Installation Performance Specification," under the REPLY command in Chapter 5, for further information on setting the TOD clock.

The alternate CPU recovery (ACR) feature, available with multiprocessing, attempts to maintain the availability of the system when one of the processors is unable to continue. The operable processor attempts to perform the work that was in progress on the failing processor until the system can continue normal processing with the failing processor logically offline. You are informed of the loss of I/O devices, channel paths, or other processor functions. Tasks requiring the lost functions or devices are terminated or not allowed to run. However, on a 168 MP channel reconfiguration hardware (CRH) maintains all the I/O devices on the failing processor so that no device or channel paths are lost. If the system is unable to continue processing because of an unrecoverable error, and an error occurs during ACR processing, the system enters a wait state. Refer to *OS/VS Message Library: VS2 System Codes*, for a discussion of the action to be taken.

Starting, Quiescing, and Stopping the System

The following briefly describes the IPL and initialization process and several ways MVS allows the operator to initialize the system quickly and efficiently.

Starting the System

To start the system:

- Set the LOAD UNIT dials to the unit address of the IPL volume, and press the LOAD key on the control panel.
- Respond to the system parameter messages that appear on the screen.
- Set the time and date.
- Start the system input readers and output writers.
- Vary devices offline as appropriate.

Some processors, such as the 3158 do not have LOAD UNIT dials or a LOAD key. On these, first type in the load unit address or point to it with the light pen. Next, press the keys that the display prompts you to press, or point the light pen, if present, to the LOAD and EXECUTE functions, in that order. This accomplishes the load.

During nucleus initialization (NIP), the first messages appear on the top line of the screen, and subsequent messages are displayed on succeeding lines. When the number of lines displayed reaches eight, as each new line is displayed, the oldest (top-most) message is removed from the screen to provide room for the new message. You do not have to delete any messages during NIP.

MVS Initialization Process

The OS/VS2 MVS initialization process eliminates unnecessary informational messages and allows your installation to:

- Specify default parameters during system generation
- Use parameter lists in SYS1.PARMLIB
- Specify quick start to take advantage of a previously initialized link pack area (LPA)
- Eliminate the time-of-day (TOD) clock messages; these messages appear only when the clock is not set
- Eliminate system management facilites (SMF) parameter specification by the operator
- Let the system control program issue certain operator commands
- Allow the subsystem to automatically start readers, writers, and initiators
- Control the processing rates of individual jobs

The nucleus initialization program (NIP) attempts to initialize the system regardless of system parameter specification errors. If a system parameter is in error, the system prompts you for a correct one. If the incorrect parameter came from SYS1.PARMLIB, you can press EOB or the equivalent key (such as END or ENTER, depending on your console model) to select the system defaults, if available, or cancel the option.

Limiting Apparent Machine Size (Uniprocessor Only)

There are times when it is necessary to limit the amount of real storage available to the operating system (for example, during system testing or when storage hardware errors occur). The following procedure, which should be done only at the direction of your system programmer, describes how to limit the size of real storage.

1. Place an address compare stop at location hexadecimal 80 before the initial program load.

2. After the system stops, alter the contents of location hexadecimal 09 to one of the hexadecimal values listed below with their corresponding machine sizes:

   X'D0' — 768k
   X'D1' — 1024k
   X'D2' — 2048k
   X'AB' — 3072k
   X'D3' — 4096k
   X'D4' — 8192k

Initial Program Loading

The initial program loading procedure is performed after a power-on following an electrical shutoff, after malfunctions that require reloading the control program into storage, after scheduled maintenance, and as part of switching from one system to another. The operating procedures manual for your model describes the detailed IPL procedure.

Loading the system causes the system MANUAL light to turn off, turns on the system LOAD light, and starts reading the IPL program from the system input device. After the IPL program is read into real storage, control is passed to it, and the LOAD light turns off. If either the reading operation or the passing of control is unsuccessful, the processor pauses and the LOAD light stays on. When the IPL program gets control, it loads the nucleus of the control program into real storage. After the nucleus is loaded, control is given to a nucleus initialization program (NIP).

If the IPL program does not finish successfully, or an I/O error occurs during NIP processing, the system is placed in a disabled wait state and an error code is placed in the low-order 12 bits of the program status word (PSW). The operating procedures publication for your processor model provides instructions for displaying the PSW. If the WAIT indicator turns on without a message, display the PSW, note the error code, and follow the instructions for that code given in *OS/VS Message Library: OS/VS2 System Codes*. In a multiprocessing configuration, ignore the WAIT indicator if it turns on because the processor is waiting for another processor to finish IPL.

All printers designated as alternate multiple console support (MCS) consoles must be ready at IPL time or the IPL sequence does not complete. If IPL is suspended because the alternate consoles are not ready, making them ready allows IPL to complete normally.

Loading a Secondary (Alternate) Nucleus

The IPL program loads a standard, or primary, nucleus unless you cause it to load a secondary nucleus. Loading a secondary nucleus might be required for system testing or for initializing the system using a different set of parameters. Use the following procedure, which should be performed only at the direction of your system programmer, to load a secondary nucleus.

1. Place an address stop at location hexadecimal 80 or put the processor in instruction step (single cycle) mode before the initial program load.
2. Display the contents of location hexadecimal 08 after the system stops (it should contain hexadecimal F1).
3. Alter the contents of location hexadecimal 08 to the EBCDIC character specified by your system programmer. This character specifies the IEANUC0x member of the SYS1.NUCLEUS data set to be loaded.

If you are uncertain of the procedures for address stopping, displaying storage, or altering storage, refer to the operating procedures manual for your model processor.

Quiescing the System

Issuing the QUIESCE command causes the system to suspend the processing of all active jobs and prevent the starting of any new ones. The system enters the MANUAL state, the MANUAL light is on, and no processing is being done. Quiescing the system does not affect any job step timings (for accounting purposes). Processing can be continued by pressing the RESTART key (or performing the RESTART function).

Stopping the System

When all processing (including subsystem processing) has completed, use the HALT command to ensure that all system statistics and data records in storage are collected for system recording facilities.

Allocating and Controlling Configuration and Devices

Device allocation is the assignment of input/output devices to job steps. Device allocation includes the assignment of volumes and data sets to the job step. In performing these resource assignments (devices, volumes, and data sets), the system might interact with you to:

- Mount or dismount volumes
- Make decisions. For example, to bring a device online or wait
- Describe wait situations, such as waiting for data sets or waiting for volumes

Requests for device allocation originate from two sources: JCL data definition (DD) statements and dynamic device allocation requests.

The JCL data definition statements can be entered to the system by:

- Normal job input to the job entry subsystem reader
- Jobs submitted through the TSO SUBMIT command
- Started cataloged procedures
- The MOUNT command
- TSO logons

Each form of input can be controlled by the operator (for example, holding the reader, or setting the maximum logon count).

Dynamic device allocation/unallocation requests originate from within executing programs and therefore are not subject to the same controls as JCL DD allocations.

Operationally, the assignment of devices is influenced by:

- The online/offline status of the device. Except for special exceptions such as the OLTEP process, only online devices can be allocated to job steps. Offline devices can be brought online by the VARY command or in response to allocation recovery message, IEF238D.
- The MOUNT attribute. The MOUNT attribute, which applies only to tape or DASD devices, is influenced by the MOUNT and UNLOAD system commands. Allocation requests that can be satisfied by mounted devices are processed quickly and without operator intervention (for example, mount messages).
- The USE attribute. A parameter of the MOUNT command, the USE attribute affects the type of data sets that can be allocated on a tape or DASD volume. Having a proper mix of volumes with the various USE attributes can reduce the amount of volume mounting required of the operator.

Device Assignment

The following brief overview of device allocation suggests some general guidelines that are useful.

- The job scheduler of the operating system, using information from data definition (DD) statements, determines the input/output resources to be assigned to a job and the volumes that are required. If these volumes are not mounted, the job scheduler issues a message stating that a specific volume or a scratch volume must be mounted. When both scratch volumes and specific volumes are requested by the same job, mount the scratch volumes first to achieve maximum system efficiency.
- Occasionally you receive two mount messages for the same volume — one starting with IEF and the other with IEC. Treat the two messages as though they were one. The second request is merely a reminder that you have not yet mounted the volume.
- Never mount a blank tape volume unless specifically directed to do so, because the system scans the entire volume for a tape label. If an unlabeled tape is required, a tapemark must be written to avoid unnecessary scanning. After you mount the tape volume and ready the drive, the system reads the volume label. If an incorrect volume is mounted, the system unloads the incorrect volume and repeats the mounting message.
- When you know that several jobs are going to need a volume, use the MOUNT command to reserve that volume on a device. Volumes reserved by a MOUNT command are not demounted by the system until an UNLOAD command is issued. Allocation processing is faster when the required directed access volume is reserved rather than removable. Note that the MOUNT command *must not* be used for devices managed by JES3; see *Operator's Library: OS/VS2 MVS JES3 Commands*.

- When referring to I/O devices in the unitaddr parameters of operator commands, you must use the unique unit address assigned to each device.
- Your installation can define symbolic group names of one to eight characters to be used by programmers in (DD) statements. The number of devices associated with a symbolic name can range from one to the total number of devices in your installation. The symbolic name allows the devices to be grouped according to the attributes your installation considers significant. Do not use these symbolic names in operator commands, except in the optional device name parameter of the MOUNT and START commands.
- Make sure there are sufficient work volumes available to satisfy requests for temporary data sets at peak loads. Failure to do this can cause the system to request additional scratch volumes. Work volumes should be balanced across channels to increase system efficiency.

Automatic Volume Recognition

Automatic volume recognition (AVR), an integral part of the operating system, allows you to mount labeled volumes on unused drives not managed by JES3. The the system recognizes and remembers these volumes and assigns the drives to later job steps as required.

Tape device allocation under MVS has been changed from previous systems such as MVT or SVS, and these changes have affected the practice of premounting tape volumes. In MVT and SVS, premounting a tape volume on an eligible device caused a request for that volume to be allocated to the drive on which it was premounted. This happened because MVT and SVS read the volume serial numbers of volumes on all the eligible tape drives before device selection took place. In MVS, this is not the case, and, therefore, it can be difficult to know where to premount a tape volume.

In MVS, if a specific volume could be allocated to multiple generic device types, AVR reads the volume serial numbers of the premounted volumes one generic device type at a time, according to the order of generic device types in the device preference table (PREFTAB). The installation can either specify the order of device preference at system generation with the DEVPREF parameter of the SCHEDULR macro or take the default values supplied by the system.

Displaying Configuration and Device Information

System commands enable you to display the status (such as online or offline) of a single device or all devices of specific type, such as DASD or tape. By using the DISPLAY and MONITOR commands, you can determine what devices are available for job processing, as well as the configuration and status of the system consoles.

The DISPLAY command requests the status at the time the command is issued. The MONITOR command requests a continual display of status information.

Controlling The System

To control the operating system effectively, you must be aware of the current status of the system. Status information, such as the number of jobs and teleprocessing functions, enables you to take the appropriate actions to operate the system efficiently and correct potential problems.

MVS includes system and subsystem commands that display job and system status either at a specific time or continually at a regular interval. The DISPLAY and MONITOR system commands are used to obtain this job, system, or time sharing status on both display and non-display consoles. For display consoles only, the TRACK system command can also be used to display this information continually at regular intervals.

In a multiple console environment, you can use the MSGRT system command to route status information to one or more consoles. Communication between operators in a multiple console environment, as well as communication with time sharing users, is possible through the use of the SEND command.

The SET command provides the option of setting the local time and date and of changing the system's job scheduling parameters. You should, however, consult your system programmer before changing parameters.

Use of the System Restart Key

The system RESTART key has several functions. It can be used to:

- Restart the system after a QUIESCE command has been entered. Refer to Chapter 5, "Quiescing the System," for a discussion of this procedure.
- Restart the system from a wait state that is specified in *OS/VS Message Library: VS2 System Codes*.
- Invoke recovery for a routine that has entered an invalid wait state or disabled loop.

  An invalid wait state is characterized by:

  - The processor system light remaining OFF
  - The processor wait light remaining ON
  - The wait state code in the PSW (IC) is not listed in the *OS/VS Message Library: VS2 System Codes*.

  A disabled loop is characterized by:

  - The processor system light is ON
  - The processor wait light is OFF
  - Repetitive I/O or nonproductive processing occurs and the PSW (IC) frequently displays the same addresses
  - All interrupts are masked off for the system

When you have determined that the system is in a disabled loop or an invalid wait state, press the RESTART key to initiate recovery processing. This procedure, which loads a new PSW from storage location 0, allows the user or system program to ABEND and invokes the necessary recovery routines.

In a multiprocessing environment, press the RESTART key on the processor in the wait state or loop; the other processor continues normally.

If the system does not recover, follow your installation's procedures for recording system problems. You may want to refer to *OS/VS2 System Programming Library: SYS1.LOGREC Error Recording* for the procedures to follow to print the LOGREC data set. Once you have recorded the system information, you must reinitialize the system.

Controlling Jobs

OS/VS2 MVS provides system commands that can be used to control the flow of a job through the system. Job-related commands enable you to start, stop, or terminate a job as well as to modify the job's parameters and performance group. You can also restart a job that has failed.

Updating System Data Sets

Running jobs that update system data sets (SYS1.LINKLIB, SYS1.LPALIB, and SYS1.PROCLIB) during normal production is not recommended. For example, don't run jobs that might try to use a cataloged procedure concurrently with a job that is updating that procedure.

To run a job that updates system data sets, perform the following procedures:

1. Make sure no other jobs are active in the system; use the appropriate subsystem command to prevent the initiation of jobs on the job queue.
2. Make sure no other jobs are entered into the system; use the appropriate subsystem command to stop all input readers.
3. Assign the highest priority to the update job.
4. Run the update job.
5. Resume normal processing — use the appropriate subsystem command to free the jobs placed on the hold queue earlier.

In JES3 systems, you can use the //*MAIN card UPDATE= parameter to update the procedure library.

Note: An update to SYS1.LPALIB is not effective until an IPL is performed. This data set is not reloaded for the duration of the current IPL.

Restarting a Job

Once a job is executing, it might end abnormally because of a hardware, programming, or system error. Such an ending can occur at any time during program execution. Valuable machine time would be lost if an abnormal end occurred during one of the last job steps of a multistep program or in the middle of a long job step, and execution had to start again at the first job step. There are two ways of avoiding this problem; automatic restart and deferred restart.

For JES2 jobs and JES3 jobs executing on MVS processors, the checkpoint/restart feature of the system is provided to allow a restart of an abnormally ended job either at the beginning of a job step or at a checkpoint within the current step. The programmer submitting the job provides for an automatic restart or a deferred restart.

Note: In a JES3 environment, failure options for jobs executing on an ASP main processor are determined by options specified on the //*MAIN statement in the job stream.

Automatic Restart

If the programmer submitting the job has provided for an automatic restart and the job ends abnormally, you receive the following system message:

```
* id IEF225D SHOULD jobname.stepname.procedure [checkid] RESTART
```

This message allows you to prevent repeated restarts at the same checkpoint or job step.

When you are requested to authorize an automatic restart, you can reply YES, HOLD, or NO as follows:

- Reply YES if the restart is to be performed at a specific checkpoint or job step for the first time. If it is a job step restart and the step to be restarted used a card input data set that was not part of the SYSIN stream, you must return, to the appropriate hoppers, all cards read by the job step before it ended abnormally. If it is a checkpoint restart follow the programmer's instructions for replacing the input cards.
- Reply HOLD if you want to defer the restart: for example, to permit another job to run first. You must issue the appropriate subsystem command when you are ready to restart the job. Also, if desired, you can cancel the job. However, canceling the job can result in unrecoverable paging space or the failure of certain data sets to be deleted if virtual I/O is being used.
- Reply NO if a restart at a specific checkpoint or job step has been requested repeatedly. When your reply is NO, and the programmer wants a restart to be performed, he must resubmit the job for a deferred restart.

If the programmer specifies VIRTUAL=REAL (V=R), the job is processed entirely in real storage; it is not paged out. For a V=R job, the restart may be delayed while the system waits for the allocation of storage. If another job is using the required storage, you get no message — only a delay. Enter DISPLAY A,L to see if a system task or other job is using the storage required by the job with a V=R region. You can then stop or cancel the conflicting task or job. The system might request that you mount data volumes other than those required at the beginning of the job.

Note: Any operator commands in the input stream of the job step being restarted are not executed.

Deferred Restart

If the programmer submitting the job has provided for a deferred restart and the job ends abnormally, he must resubmit the job for the deferred restart. To restart the job, the programmer must provide a restart deck for submission to the system through the system input reader. The JCL statements to be included in the restart deck are described in detail in the publication *OS/VS2 JCL*.

The device configuration of your system at the time of restart need not be the same as it was when the job ended abnormally. However, enough devices must be available to satisfy the needs of the job step being restarted. The system under which a step restart is run need not be the same as it was for the job's original execution. However, a checkpoint restart should be run under the original system unless the alternate system can meet the following restrictions:

- The job entry subsystem is the same.
- The release number is the same.
- The link pack area modules in use at the checkpoint reside in the same storage locations.
- An area of storage identical to the original area is available to a V=R job.

If the required storage is not available, the system cancels the restart and you receive the following message:

```
IEF209I VIRTUAL STORAGE UNAVAILABLE FOR jobname.stepname.procedure
```

If the required storage is not available, it is for one of the following reasons:

- The link pack area expands into the required storage. This can occur if an initial program loading (IPL) has been performed after the original execution of the job and before the restart. If it does occur, contact your system programmer for a respecification of the system parameters and repeat initial program loading using the new values.
- The system queue (SQA) area expands into the required storage. When this occurs, contact your system programmer for a respecification of the SQA system parameter and repeat initial program loading using the new SQA value.

When a job restarts correctly, you receive two messages: IHJ006I JOB RESTARTING and IHJ008I JOB RESTARTED. If, for V=R jobs, these messages do not appear, enter DISPLAY A,L to see if a system task or other job is using the required storage. You can then stop or cancel the conflicting job. The system might request that you mount data volumes other than those required at the beginning of the job. In addition, any card input data sets that have been used by the failing job step must again be made available to the system.

Controlling the External Writer

The external writer can write SYSOUT data in ways and to devices not supported by job entry subsystem (JES2 or JES3) output writers. You must use an external writer to write SYSOUT data to a tape or a DASD, or to write a SYSOUT data set with a SYSOUT JCL parameter that specifies a special installation-supplied writer routine. You can also use the external writer to write output to any devices attached to the control processor; these devices are called local devices. You can not use the external writer to write SYSOUT data to remotely attached devices.

Your installation can reserve special output classes for SYSOUT data to be written by an external writer, but this is not necessary. If both a subsystem writer and an external writer are assigned to the same output class, the SYSOUT data is selected on a contention basis; a subsystem writer does not select SYSOUT data sets that are to be written by an installation-supplied writer routine. To determine whether any SYSOUT data sets requiring installation-supplied writer routines are waiting to be written, use the appropriate subsystem command (refer to the appropriate operator's library publication, JES2 or JES3 commands).

To start an external writer, enter a START command that specifies either the IBM-supplied procedure (XWTR) or an installation-supplied procedure. You can specify the device on the START command and optionally specify whether the external writer should begin immediately to process all output directed to local devices and enqueued in the output classes assigned to the external writer. If one or more output classes are specified in the START command or on the EXEC statement in the cataloged procedure used to start the external writer, the external writer begins to process immediately. If no classes are specified, the external writer waits for you to assign data selection criteria with the MODIFY command.

By using the MODIFY command, you can cause the external writer to select data sets according to output class, job-id, special forms name, and destination (remote or local). You need specify only those parameters that are to be changed. Omitted parameters on the MODIFY command remain set to their current value as set by defaults or a previous MODIFY command.

The defaults are the specified classes, local destination, any job, any forms, and any writer routine. If the immediate-processing option is used, classes are specified on the EXEC statement or the START command. The absence of an installation writer name in the

SYSOUT parameter of the JCL statement indicates that an installation-written writer routine or the standard writer is being used. If the immediate-processing option is not selected (class is not specified), the defaults are any class, any destination, any job, any forms, and any writer routine. However, these options are not used until a MODIFY command is entered.

When no particular forms are specified, forms-mounting is accomplished on demand as directed by the external writer. When no particular writer routine is specified (any writer), writers are invoked as required by the external writer, subject to your authorization. You are asked to authorize the use of any routine not previously authorized for this external writer. If you do not wish to authorize the routine, you may delete the data set or cause it to be written by the standard writer (STDWTR).

You cannot specify print trains and forms control buffers as data set selection criteria. These two parameters, plus the 3525 interpret feature, are specified on the DD statement in the cataloged procedure used to start the external writer (and possibly override the START command) and apply to all data sets written by this writer. Any separation of output by print train, forms control, and punch card interpretation must be controlled by directing the SYSOUT to separate output classes.

Controlling Time Sharing

Time sharing allows programmers at remote terminals to develop, test, and execute programs without the turnaround delays that occur when they submit jobs to a computer center. With time sharing, a large number of jobs can share the resources of the system concurrently, and the execution of each job is controlled primarily by the remote terminal user. Thus, time sharing can be defined as the shared, conversational, and concurrent use of a computing system by a number of users at remote terminals.

Time sharing in MVS is provided by TSO, formerly an option of the control program (hence, the name time sharing option) and now a standard feature. A time sharing environment in which communication between TSO and the terminal is controlled by the virtual telecommunications access method (VTAM) is called TSO/VTAM. A time sharing environment in which communication between TSO and the terminal is controlled by the telecommunications access method (TCAM) is called TSO/TCAM.

You start and stop TSO/VTAM time sharing by using the START and STOP commands; you start and stop TSO/TCAM time sharing by using the MODIFY command. You can modify TSO/VTAM time sharing by using the MODIFY command; you cannot modify TSO/TCAM time sharing. The rest of the commands that control time sharing are the same for both environments.

For further information about VTAM, see *Operator's Library: VTAM Network Operating Procedures*. For further information about TCAM, see *Operator's Library: OS/VS TCAM*.

Note: Subsystem (JES2 and JES3) commands are not accepted from TSO terminals.

Recording System Information

To monitor the performance of the system and to gather system statistics for accounting and billing purposes, some method of recording system activity must be used. The operating system has several facilities for recording facilities that include:

- System log
- System management facilities (SMF)
- System activity measurement facility (MF/1)
- Generalized trace facility (GTF)

The following text describes each system recording function.

The System Log

The system log is a data set residing in the primary job entry subsystem's spool space. It can be used by problem programs, operators, and, optionally, as the hard-copy log (see "System Log" in Chapter 2).

In JES3 systems, the system log can record console activity. When used to record console activity, the system log is referred to in JES3 messages as DLOG.

The system log is queued for printing when the number of messages recorded reaches a threshold specified at system initialization. You can force the current system log data set to be queued for printing by issuing the WRITELOG command.

Several kinds of information can appear in the system log:

- Job time, step time, and data from the JOB and EXEC statements of completed jobs entered by user-written routines
- Operating data entered by problem programs using a write to log (WTL) macro instruction
- Descriptions of unusual events entered by the operator through the LOG command
- WTO and WTOR messages including their routing codes and a time stamp
- Accepted replies to WTOR messages
- Commands (system and subsystem)

System Management Facilities

System management facilities (SMF), a standard feature of MVS, consists of system routines and optional user-written exit routines that collect, format, and record system and job-related information. An installation can use this information to evaluate system workload, resource utilization, and performance. System-related SMF routines record information on workload, configuration, and paging activity. Job-related SMF routines record information on processor time, SYSOUT activity, and data set activity for each job, job step, and TSO user.

The information gathered by SMF and user written exit routines is recorded on direct access volumes in the SYS1.MANX and SYS1.MANY data sets. These data sets must be online at system initialization. When one data set is full, you receive a message telling you to dump the filled data set. Data recording is automatically switched to the alternate data set.

Use the IFASMFDP program to dump an SMF data set. To speed execution, it is a good practice to submit the dump job before it is needed, hold it in the job queue, and release it when necessary. Sample JCL is as follows:

```
//DUMPX     JOB     ...
//STEP1     EXEC    PGM=IFASMFDP
//DUMPIN    DD      DSN=SYS1.MAN {X}
                                 {Y}
//                  DISP=OLD
//DUMPOUT   DD      UNIT=tapeaddr,
//                  VOL=SER=serial,
//                  DISP=(NEW,KEEP),
//                  DSN=smfdata
//SYSPRINT  DD      SYSOUT=A
```

Error Recovery. If an I/O error occurs while writing to SYS1.MANX or SYS1.MANY, you receive a message and SMF switches to the alternate data set. Allow the system to become inactive, allocate a new data set to replace the failing one, and reinitialize the system.

Manually Switching SMF Data Sets. Use the SWITCH command to manually switch the recording of SMF data from one data set to the other.

System Activity Measurement Facility (MF/1)

MF/1 is an analysis tool that can be used to monitor selected areas of system activity and obtain feedback in the form of SMF records and/or formatted reports. An installation can use MF/1 to monitor the utilization of individual processors, channels, and devices. MF/1 can also be used to identify periods of time when resource utilization is exceptional.

MF/1 permits the gathering of information on the following activities either individually or in combination:

- Processor activity
- Channel activity and channel-processor overlap activity
- I/O device activity
- Paging activity
- Workload activity

System Tracing

System tracing is an integral part of the operating system that records, for diagnostic purposes, events that occur during system initialization or operation. The system trace is the only trace that can be used between subsystem initialization and the start of the generalized trace facility (GTF). Refer to the TRACE command in Chapter 5.

The Generalized Trace Facility

The generalized trace facility (GTF) is used as an aid to determine and diagnose problems that occur while using the operating system. GTF records system and user-defined program events and edits the data to the desired format. The system events that can be recorded (traced) are:

- I/O interrupts — IO
- SIO interrupts — SIO
- SVC interrupts — SVC
- Program interrupts — PI
- External interrupts — EXT
- System dispatcher events — DSP
- System recovery routine activity — RR
- System Resources Manager activity — SRM
- Remote network I/O activity — RNIO
- Program-controlled interrupts (PCIs) for I/O interrupts
- TRC events associated with the trace task itself

User-defined events can be recorded by specifying USR data to be passed to GTF when an executing program issues the GTRACE macro.

Note: JES3 has an additional trace facility — JES3 Event Trace Facility — that is described in *Operator's Library: JES3 Commands*.

Dynamic Device Reconfiguration

There are times when a device hardware problem or a need to improve system performance makes it necessary to move a demountable tape or DASD volume from one device to another. Dynamic device reconfiguration (DDR), invoked by the SWAP system command, allows you to move or swap a demountable volume.

The SWAP command enables the operator to turn on or off system-initiated swapping requests. When DDR is on, the system dynamically performs the swapping function whenever the originally-allocated device is unavailable for use. DDR tells the operator to mount the volume on another available device. When the swapping function is turned off, you can invoke operator-initiated DDR by issuing the SWAP command and specifying the "from" and "to" device addresses (See the SWAP command in Chapter 5 of this book).

On JES3 systems, DDR interfaces with JES3 to ensure that the "to" device has not been assigned to another job or function. When the swap is complete, DDR notifies JES3.

The following devices are supported by DDR:

- 2400 and 3400 series tape drives. Since the "to" device must support the density of the "from" device, you can swap a volume from a 2400 series drive to a 3400 series drive but not from a 3400 series drive to a 2400 series drive. Note that if you are using a 7-track tape drive at 200 BPI, be certain that the "to" device also has the 200 BPI feature.
- 2501, 2520, 2540, 3505, 3525, 1403, 1443, and 3211 unit record devices. These devices are not swapped by system-initiated DDR. The SWAP command must be issued to swap these devices.
- 2319/2314, 3330/3333, and 3340/3344, and 3350 direct access devices. If you are using a 3348 Model 70F Data Module, make sure that the "to" 3340 device has the fixed-head feature installed. The 3344 and 3350 are supported only by the operator-requested DDR facility. When swapping a 3340/3344 device with the fixed-head feature, care must be taken to ensure that the "to" device also has the fixed-head feature installed.

The following devices are not supported by DDR:

- Graphic or teleprocessing devices
- Shared DASD devices, unless the device is swapped to itself
- 3330V device type, see *Operator's Library: IBM 3850 Mass Storage System (MSS) Under OS/VS*, for information about the 3330V
- System-requested DDR for 3344 and 3350 fixed-head DASD devices
- Any permanently-resident volume (such as a system residence or page data set volume).

Chapter 2. Console Configurations

This chapter describes the support for the various console combinations that can be included in a System/370 supported by MVS. A System/370 console can be one or more of the devices listed in Figure 2.1. The devices assigned as consoles are specified at system generation. Those specified are the only devices that can be used.

Multiple-console support (MCS) is standard for MVS. The multiple-console configuration has a master console and one or more secondary consoles that perform installation-assigned functions. The master console and each secondary console has an alternate console. In addition, MVS supports a major variation of the multiple console configuration: multiple console support with JES3 console support. An extensive discussion of JES3 console support can be found in the JES3 reference manuals listed in the preface of this publication.

The operating procedures publication for your processor contains information about the physical device or devices that make up the console combination for an installation.

I/O (input/output) devices that MVS supports have three logical conditions that determine how or if the device functions. The devices can be:

1. Online: The system assigns functions with these two limitations:
   - The device must be capable of performing the function (a card reader cannot be used as an output device).
   - The device cannot be assigned as a console.
2. Offline: The system cannot use the device at all.
3. Console: The system can use the device to send messages to you, and you can use the device to issue commands (if the device has input capability), but you cannot use the device for other input/output purposes.

You can use the I/O devices listed in Figure 2.1 either online or offline or as a console if specified during system generation. This chapter discusses the devices only when used as consoles.

Consoles

Figure 2.1 shows the devices that can be used as System/370 consoles while you are using MVS. There are four categories:

1. Console printer-keyboards (a 3215, for example)

   The console printer-keyboard is a full-capability console. That is, it is a combination input (operator-to-system) and output (system-to-operator) device. This device provides a printed record of system communications.

2. Display consoles (a 2250, for example)

   The display console can also be a full-capability device, but it does not provide a permanent record of system communications.

3. Output-only consoles, which can be used for the MCS (multiple-console support) hard-copy log (a 1403 or a 3284, for example)
4. Composite consoles made up of separate input and output devices (a 2501 and 1403, for example)

A composite console is made up of two devices, each having a unique device address. You must specify both addresses when you assign a composite console. Console operation is the same as with a combination input and output device. Before IPL, both devices of a composite console must be ready and must remain so until IPL is complete. This restriction applies to MCS output-only consoles as well.

The two most common composite consoles are made up the the IBM 3504 or 3505 Card Read Punch as the input console and the IBM 3211 Printer as the output console. The standard line length is 80 characters for input messages and 120 characters for output messages.

Entering Commands Through a Composite Console

- Press the STOP key on the reader.
- Place the command card in the reader; no double slash, //, is used on a command *card*, as opposed to a command *statement* entered in the input stream.
- Press the START and EOF keys on the reader.

1052-7	Printer-Keyboard (Note 1)
1403	Printer (output only)
1443-N1	Printer (output only)
2250-1 & 3	Display Unit
2260-1	Display Station
2501-B1 & B2	Card Reader (input only)
2520-B1	Card Read Punch (input only)
2540	Card Read Punch (input only)
2740-1	Communication Terminal (via BTAM)
3036	Display Console
3066	System Console
3158	Display Console
3210-1 & 2	Console Printer-Keyboard
3211	Printer (output only)
3213	Console Printer (output only)
3215	Console Printer-Keyboard
3277-1 & 2	Display Station (Note 2)
3284-1 & 2	Printer (output only)
3286-1 & 2	Printer (output only)
3505	Card Reader (input only)
3525	Card Punch with read feature (input only)

Notes:

1. This device is supported only when attached through a 2150 adapter.
2. MVS supports the 3277-1 as output only.

Figure 2.1. Devices Supported as MVS Consoles

Multiple-Console Configuration

The multiple-console configuration consists of from one to ninety-nine consoles. One console is the master console, and the rest are secondary consoles. The master console is always active. The secondary consoles can be active (being used) or inactive. Composite consoles can be specified at system generation. For every composite console that is specified, the total number of consoles allowed decreases by one. (A composite console represents two *logical* consoles.)

The master console is your principal means of communicating with the system. It is the only console that you can use to:

- Enter all MVS operator commands
- Change the status of the hard-copy log
- Switch to a new master console

The master console also receives all messages not specifically assigned to a secondary console. Only one master console can be active at any one time.

A secondary console is used for specific types of operator to system communication when it is more convenient to have a console located away from the processor. A secondary console might, for example, be located close to tape or disk drives or remote teleprocessing devices to make it easier for the operator in that area to see which magnetic tapes or disk packs to mount. A secondary console cannot enter all MVS commands (see "Operator Command Groups" later in this chapter), and can receive only those messages that are specified to be routed to that console.

Operating with Multiple-Console Support

Each console in an MCS configuration is defined as master or secondary during system generation. The following additional console attributes can also be defined then or later during system operation:

- The alternate consoles (backup consoles) assigned in case of a malfunction
- The operator commands (operator command groups) that MVS accepts from that console
- The message groups (routing codes) that the console receives

Alternate-Console Selection

Each console, whether secondary or master, is assigned an alternate console for backup in the event of a malfunction. The alternate console is usually another functioning console in the system configuration, but it can be any device with the same or greater capability (that is, a full-capability console can be the alternate for a status display console, but an output-only console cannot back up a full-capability console). A system detected failure of one of the consoles causes automatic switching to an active console in the failing console's alternate chain. If an active alternate cannot be found, the master console is assigned. You can determine the current alternates for each console by issuing the DISPLAY CONSOLES command.

If a console fails, MCS selects the alternate. However, depending on the arrangement defined for your installation, the method of selection differs. Figure 2.2 shows one possible arrangement of alternate consoles. An arrow points to the console that assumes the functions of the failing console.

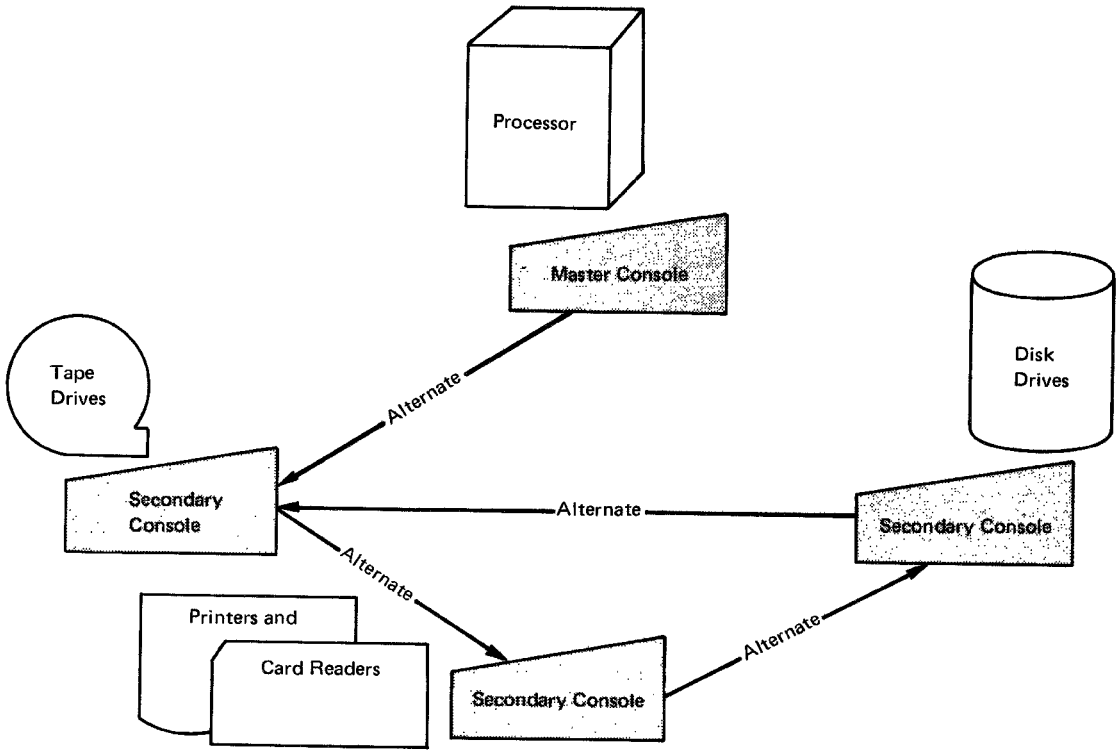


Figure 2.2. Alternate Chain for Multiple-Console Configuration

Automatic Console Switching

When a console failure occurs, the communication routines in MVS attempt to switch to the failing console's alternate. The first active alternate console MVS finds is assigned the functions of the failing console. If no alternate console is active, the master console assumes the failing console's function.

MVS transfers to the alternate console all unanswered and unissued messages from the failing console. However multiple-line messages being written when the failure occurs are not transferred. The alternate console operator receives a message stating the functions the alternate console has assumed from the failing console.

Note: If the master console is attached through a failing processor in an MP environment, MVS switches consoles so that a console attached through the functioning processor becomes the master. In order for the switch to happen on a 168 MP, the master console's alternate must be active and attached to the functioning processor. If MVS cannot switch consoles, the master console is still accessible through channel reconfiguration hardware. The console switch occurs whether or not channel reconfiguration hardware is active.

Reassigning the Master Console

If the master console is operating properly, you can switch to another master without disrupting normal operations. Enter the VARY command with the MSTCONS keyword through the current master console. If you try to enter this command from a secondary console when the master console is operating normally, the command is rejected and a message appears at the secondary console to indicate that the switch did not take place.

Occasionally, the system does not detect a failure. In this case, the master console operator should press the external interrupt key on the operator's control panel to switch to the alternate.

When no alternate consoles are active, console switching cannot take place. If an attempt is made (either by MVS or the operator) to switch the master console, all active secondary consoles receive a message asking that the VARY MSTCONS command be entered. The first VARY MSTCONS command entered with the correct syntax from any secondary console is accepted and processed. A message is issued that informs the new master console operator of the change and provides the unit address of the previous master console.

When no secondary consoles are active when a master console failure occurs, a no-consoles condition occurs. See "Responding to a No-Consoles Condition" later in this chapter.

Operator Command Groups

MVS operator commands are assigned to one of five command groups according to command function. The command groups and their functions are:

- Informational commands (INFO)
- System control commands (SYS)
- I/O control commands (IO)
- Console control commands (CONS)
- Master console-only commands (MC)

The commands that make up each group are shown in Figure 2.3. (For information about JES commands, refer to *Operator's Library: OS/VS2 MVS JES2 Commands* or *Operator's Library: OS/VS2 MVS JES3 Commands*.

MVS allows you to enter informational commands from any console. However, for a secondary console to enter system control, I/O control, or console control commands, the particular command group must be assigned to that console. If a secondary console enters a command that it is not authorized to enter, MVS rejects the command and sends an error message to the issuing console. The master console can enter all possible operator commands. In addition, only the master console can enter certain commands (MC commands) and the special operands of the VARY command listed in Figure 2.3.

To determine the commands that each secondary console is authorized to enter, use the DISPLAY CONSOLES command to display the console configuration.

To assign command groups, use the AUTH operand of the VARY command. This form of the VARY command is restricted to the master console and can be found in Chapter 5.

Group Number	Command Group	Commands
N/A	INFO (Informational)	CONTROL (See Note 1) DISPLAY (See Notes 1 and 4) LOG MONITOR MSGRT (See Note 1) REPLY SEND STOPMN STOPTR (See Note 1) TRACK (See Note 1)
1	SYS (System Control)	CANCEL CHNGDUMP DUMP HALT (See Note 5) HOLD (See Note 2) MODE PAGEADD RELEASE (See Note 2) RESET SET SETDMN SLIP START STOP SWITCH TRACE WRITELOG
2	IO (I/O Control)	ASSIGN (See Note 4) MOUNT PURGE (See Note 4) SWAP UNLOAD VARY { OFFLINE, ONLINE, PATH, unitaddr } (See Note 3) VARY { NET } (See Note 5)
3	CONS (Console Control)	VARY { ALTCONS, CONSOLE } VARY { OFFLINE, ONLINE, PATH, unitaddr } (See Note 3)
N/A	MC (Master Console only)	All of the above commands, plus DUMP, FORCE, QUIESCE, and VARY { AUTH, CH, CN, CPU, HARDCPY, MSTCONS, STOR }

Notes:

1. CONS command group when message routing is specified.
2. HOLD and RELEASE are explained in *Operator's Library: OS/VS TCAM.*
3. IO command group when specifying a non-console device; CONS when specifying a console.
4. ASSIGN, PURGE, and DISPLAY 3850 are explained in *Operator's Library: IBM 3850 Mass Storage System (MSS) Under OS/VS.*
5. HALT NET and VARY NET are explained in *Operator's Library: VTAM Network Operating Procedures.*

Figure 2.3. System Command Groups

Message Routing Codes

Each MVS message is associated with a routing code. This code, a decimal number from 1 to 15, is used by the system to determine which console or consoles should receive a message.

The system programmer assigns routing codes, as specified by your installation, to the consoles attached to your system so that a specific message type is routed to the proper console.

Routing codes do not appear with a message. To determine the routing codes each console receives, use the DISPLAY CONSOLES command to display the console configuration.

To assign these routing codes, use the ROUT operand of the VARY command. Figure 2.4 lists the routing codes.

System Routing Code	Definition
1	Master console action
2	Master console information
3	Tape pool
4	Direct access pool
5	Tape library
6	Disk library
7	Unit record pool
8	Teleprocessing control
9	System security
10	System error/maintenance
11	Programmer information
12	Emulators
13	Reserved for customer use
14	Reserved for customer use
15	Reserved for customer use
16	Reserved for future expansion

Figure 2.4. Message Routing Codes

Hard-Copy Log

The hard-copy log provides a permanent record of system activity. It is required in systems with one active display console or more than one active console. It records all commands, all in-line command responses, messages with specified routing codes, and, optionally, out-of-line command responses.

The hard-copy log can be assigned to a device, such as a printer, or to the system log by using the VARY command with the HARDCPY operand.

Messages and commands can be buffered before printing (using SYSLOG), or can be printed when issued on the hard-copy console. With either method, a permanent record is produced that can be retained by your installation to obtain information about operating conditions and maintenance. You can specify SYSLOG (the system log), which is the only buffered hard-copy device supported, or another nongraphic console as the hard-copy log.

You can use the hard-copy log to collect all messages and commands or to record selected messages. Unless otherwise specified, the hard-copy log records those messages with routing codes 1, 2, 3, 4, 7, 8, and 10.

Interpreting Hardcopy Log Messages

The hardcopy log provides a permanent record of system activities including:

- Operator commands.
- Command responses.
- System requests.
- Operator responses to system requests.
- Most messages occurring on active consoles.

Figures 2.5 and 2.6 give the format for the single-line and multiline messages, respectively.

Note that once JES3 has become active, the JES3 log becomes the system log and can be either a master hardcopy log (HLOG) or a master disk log (DLOG); message formats for JES3 logs are described in *Operator's Library: OS/VS2 MVS JES3 Commands*.

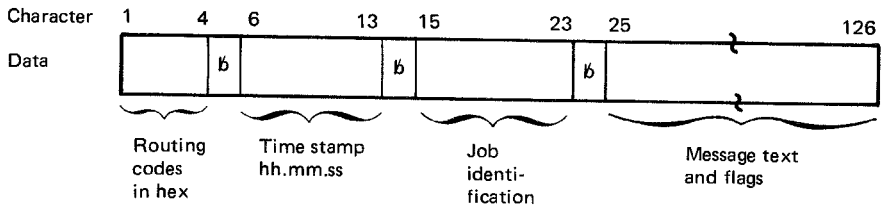


Figure 2.5. Single-Line Hardcopy Log Message Format

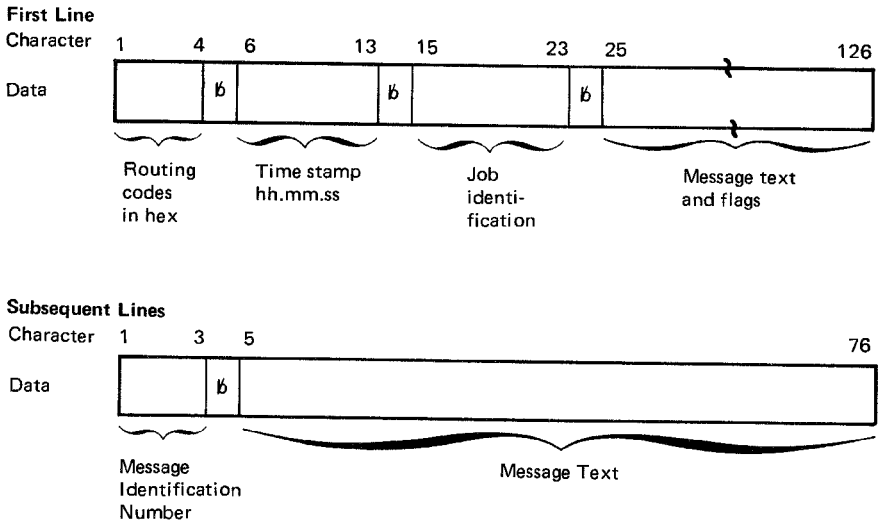


Figure 2.6. Multiline Hardcopy Log Message Format

System Log

The system log (SYSLOG) can be used as the hard-copy log. The only difference is that the system log temporarily stores messages and commands on a direct-access data set, rather than directly writing the messages and commands, as a printing console does. A system writer is used to write the system log data set when the data set is full. The WRITELOG command can also be used to ready the system log for printing.

If the system log (SYSLOG) is designated as the hard-copy device, the system attempts to switch hard-copy to an appropriate printing console if the SYSLOG device fails. If a suitable console is not active at the time of failure, hard-copy is suspended and the master console is notified.

System Assigned Hard-Copy Log

When hard-copy is not present, an attempt to add a second active console to MVS causes MVS to assign a hard-copy log. The type of hard-copy log is indicated in Figure 2.7.

Console Type	Nongraphic Master	Graphic Master
Nongraphic Secondary	Hard Copy = Master Console	Hard Copy = Secondary Console
Graphic Secondary without SYSLOG	Hard Copy = Master Console	Hard Copy = *suspended
Graphic Secondary with SYSLOG	Hard Copy = Master Console	Hard Copy = SYSLOG
*The Master Console is notified and hard-copy is suspended.		

Figure 2.7. Hard-Copy Devices Assigned by MVS

Changing the Status of the Hard-Copy Log

The VARY command with the HARDCPY operand is used to assign or modify the hard-copy log. VARY HARDCPY can be entered only through the master console. If a display console is active, or if there is more than one active console, an attempt to remove the hard-copy assignment from the console configuration is rejected.

If SYSLOG is specified, start a writer that includes the system log message class (A for MVS, unless otherwise specified in your installation).

How to Bypass a Console Malfunction

If MVS detects a console malfunction on any console in a system using multiple-console support, it tries to bypass the inoperative console using automatic console switching. To bypass a failing master console, secondary console operators can use command-initiated console switching. If a failure that the system cannot detect occurs on the master or a secondary console, you must use manual console switching (see "Reassigning the Master Console" in this chapter for details on master console switching).

Preserving the Hard-Copy Log

When the failing console is the hard-copy log, MVS searches the alternate chain of the failing console to find an active non-display console. The first one that MVS finds is assigned the hard-copy log function. If the search is not successful, MVS searches the master console and its alternate chain. If this search is not successful, MVS selects any active non-display console by searching consoles in the order specified at system generation. If the system log (SYSLOG) becomes inactive, MVS searches for an active non-display console. MVS first searches the master console's alternate chain. If unsuccessful, MVS then searches for any non-display console in the order that consoles were specified at system generation. The hard-copy function is assigned to the first non-display console found. If the search is unsuccessful, hard-copy is suspended.

Recovering Lost Messages

Some console malfunctions prevent waiting messages from being displayed on the failing console. If the failing console is full-capability console, use the following procedures to avoid losing the waiting messages:

1. Use the VARY command to make the current master console the alternate of the failing console.
2. Use the VARY command to make the failing console the master console.
3. Press the interrupt key on the processor. This action causes the inoperative console's functions and waiting messages to be automatically switched to the alternate console.

Note: Multiple-line messages in progress are lost.

Changing Console Status

Use the VARY command to change the assignment or the status of any MCS console. These three keywords tell MVS which console status to change:

1. MSTCONS

   MSTCONS is a master console-only operand. It refers to the master console. Use the MSTCONS operand only when you want to switch the master console assignment to another active console.

2. HARDCPY

   HARDCPY is a master console-only operand. It refers to any change in the status of the hard-copy log or the system log.

3. CONSOLE

   CONSOLE refers to any change in console status except the special cases where the MSTCONS or HARDCPY keyword applies.

All forms of the VARY command are described in Chapter 5 of this book.

The VARY command is also used to:

- Place one or more secondary consoles in an online, offline, or console status.
- Change a secondary console's routing code assignment, command-entering capability, and alternate console.
- Change the master console's routing code assignment and alternate console.

The VARY command for these operations must be issued from the master console or a secondary console that can issue the CONS command group.

When you use the VARY command to place an active console in an online or offline status, the command-entering capability and message routing assignments of the console are not assumed by any other secondary console. Therefore, if you wish to continue using these functions, you must assign them to another console.

An unauthorized entry or improperly specified command causes an error message to be returned to the console entering the command.

Placing a Console in Offline Status

When a secondary console must be bypassed for any reason, you must enter a VARY command to place the secondary console offline. Command activity from the console is immediately suspended, but messages can continue to be displayed until all waiting messages have been issued.

The VARY command does not cause the functions of the bypassed console to be assigned to another console. However, any messages that would be routed exclusively to the bypassed console are now routed to the master console by default.

Before you use the VARY command to bypass the hard-copy log device, you must change the hard-copy log function to another console device or to the system log.

Responding to an Inactive Console Condition

An inactive console condition is characterized by a lack of message traffic or system response to commands. It could be caused by the absence of system activity, or it may be the result of a problem in the message handling portion of the control program.

One function of the message handling portion of the control program is to check for the end line of a status display or other multiple-line messages. If for some reason the system fails to detect the end line, your console may be put into an inactive condition awaiting completion of the display or message.

If an MCS console appears inactive, check the system response by requesting a display of the time (DISPLAY T command). The system should respond within a few seconds with the time and date. If it does not respond, cancel any status displays being presented on the inactive console by using the CONTROL C,D command. If this does not return the console to normal activity, cancel any jobs that have written multiple-line messages to the console. If neither of these procedures returns the console to normal activity, assume that there is some other problem related to the console. VARY the console offline, then vary it back to CONSOLE state. If the problem recurs, switch control to another console and report the problem to your system programmer.

Responding to a No-Consoles Condition

A no-consoles condition exists when there are no full capability consoles (devices capable of input and output functions) available to multiple console support (MCS). When MCS enters a no-consoles condition, write-to-operator queue elements (WQEs) are not freed. When the IPL-specified limit of WQEs is reached, non-privileged tasks are placed in a wait state and MVS enters an enabled wait.

Note: Output-only consoles and JES3 consoles can continue to function during a no-consoles condition.

An attempt is made to notify the operator of a no-consoles condition in one or more of the following ways:

- Message IEA420A is issued to any active output-only console or JES3 console. The message text is given below.

```
IEA420A NO FULL CAPABILITY CONSOLES
  TO RESTORE MASTER CONSOLE:
  1) PRESS ENTER, REQUEST, OR END KEY ON ANY AVAILABLE CONSOLE
  2) PRESS THE EXTERNAL INTERRUPT KEY
```

- An attempt is made to sound an alarm on a device. The following conditions are necessary for the alarm to be sounded:

  A. The device was specified as a console at system generation.

  B. The alarm feature is installed.

  C. The device is online or in console status.

  D. The device is unallocated.

  E. The device is not a JES3 console.

- An attempt is made to sound the power and cooling alarm on a 3036 console. The following conditions are necessary for the alarm to be sounded on a 3036 console:

  A. The 3036 console was specified as a console at system generation.

  B. The 3036 is online or in console status.

  C. The 3036 is unallocated.

  D. The 3036 is not a JES3 console.

- Message traffic on full capability consoles stops. This lack of traffic can be the only indication of a no-consoles condition because there might not be a console on which message IEA420A can appear and MCS might not find any devices on which to sound an alarm.

- Any VARY CONSOLE commands issued while MCS is in a no-consoles condition is rejected with the following message:

```
IEE744I VARY CONSOLE REJECTED - NO FULL CAPABILITY CONSOLES
  TO RESTORE MASTER CONSOLE:
  1) PRESS ENTER, REQUEST, OR END, ON ANY AVAILABLE CONSOLE
  2) PRESS THE EXTERNAL INTERRUPT KEY
```

  Note: The VARY CONSOLE command could have been entered from a JES3 console.

Recovery Considerations

Before an attempt is made to recover from a no-consoles condition, you should note that:

- Recovery cannot be attempted until all alarms have sounded and message IEA420A has been issued.
- The attempt to recover the failing master console should be done first.
- A device is selected to be the new master console by pressing the ENTER, REQUEST, or END key on the desired device. If a device other than the failing master console is selected for recovery, it must be a full capability device that was defined as a console at system generation.
- If a device is not selected and the external interrupt key is pressed, MCS attempts to recover the failing master console.
- If a 3036 operators console was in use, see the topic "3036 Recovery Procedure".
- If the message lines on the screen are not fully recovered, press the CANCEL key.

General Recovery Procedure

The following steps are necessary for recovery from a no-consoles condition:

A. Select a device other than a 2740, preferably the master console.

B. Power on the device if it is not already powered on.

C. Press the ENTER, REQUEST, or END key. This action causes an attention interrupt to be generated on this device.

D. Press the external interrupt key.

If the recovery attempt is successful, the device that generated the attention interrupt becomes the master console and message IEE143I appears on it.

```
IEE143I OLD=console NEW=console VALDCMD=auth
IEE143I ROUTE=routcd T=a H=b
```

This message states that the failing master console's functions are switched to this device. Messages queued to appear on the failing master console now appear on the new master console. The no-consoles condition is resolved and system operation continues.

Note 1: The first device on which an attention interrupt was generated is selected for the new master console. Any attention interrupts generated for any device after the first, are ignored.

Note 2: If the message lines on the screen are not fully recovered, press the CANCEL key.

If the recovery attempt is unsuccessful, the device that generated the attention interrupt does not become active and the alarm sounds. Repeat the recovery procedure once more. If the device does not become active after the second attempt, there could be an error on the device and it cannot be used. Select another device and repeat the recovery procedure. If a device cannot be made active with the recovery procedure, a re-IPL is necessary if the following conditions exist:

1. JES3 is not active.
2. JES3 is active but does not have two full capability JES3 consoles.

If JES3 is active and has at least two full capability JES3 consoles (CN1 and CN2), the following recovery procedure can be attempted:

A. Using JES3 commands, direct JES3 to relinquish control of a full capability console.

Example:

```
*SWITCH,CN2,CN1
*DISABLE,CN2
*MODIFY,V,CN2,OFF
```

B. Press the external interrupt key. If the old master console becomes active, the no-consoles condition has been resolved and the former JES3 console is no longer needed. If the old master console does not become active, proceed to the next step.

C. Press ENTER, REQUEST, or END on the former JES3 console.

D. Press the external interrupt key. The former JES3 console becomes the new master console.

3036 Recovery Procedure

If the master console (3036 operator console) fails to recover from a no-consoles condition, perform the following steps to make the other 3036 screen (service support console) the master console.

A. Vary the 7443 or 2955 offline if it is in use.

B. Enable the I/O interface.

C. Select the configuration frame.

D. Select the program frame (3277 emulator) if the configuration frame does not indicate that is has already been selected. This step disables the hardware interface to the 7443 or 2955.

E. Press the ENTER key on the service support console; this action generates an attention interrupt for this device.

F. Press the external interrupt key.

The service support console should become the new master console. If it does not, repeat steps E and F. If unsuccessful, use the procedure in the topic "General Recovery Procedure" for other devices (non-3036) specified as consoles at system generation. If the attempt to make one of the other devices the master console is unsuccessful, a re-IPL of the system is necessary.

If the service support console becomes the master console, but the 3036 operators console has not been recovered, the operator could notice a slower response to operator commands. This can happen if an exceptional condition, such as a channel or processor error, occurs simultaneously with the operator action. To recover the unusable 3036 operators console attempt the following:

A. Issue the following command on the active 3036 (service support console):

```
VARY xxx,OFFLINE
```

Note: xxx is the unit address of the 3036 operators console screen.

B. Re-IMPL the 3036 operators console. This action puts the console in the index frame.

C. Select the configuration frame on the service support console and issue the Normalize command (see the operating procedures publication for the machine to which the 3036 is attached). If this command fails, the 3036 operator console cannot be recovered, contact your service representative for assistance. If the command is successful, an indication appears on the service support console. The 3036 operators console is in normal mode.

D. Return to the program frame on the active 3036 (service support console) and issue:

```
VARY xxx,CONSOLE
VARY xxx,MSTCONS
```

Note: xxx is the 3036 operators console's unit address.

E. The 3036 operator console should now be the master console. The 3036 service support console can now be varied offline and returned to the mode it was in before the no-consoles condition occurred.

Console Processing During System Recovery

During system recovery, normal console functions are not available. The disabled console communications facility uses the master console or its alternate for communications between you and system recovery routines.

MCS Message Queuing

Multiple console support keeps messages in buffers in virtual storage until the messages can be presented at all eligible consoles. A shortage of buffer space or the retention of an excessive number of messages can cause the system to slow down, put jobs in a wait state, exhaust the common service area (CSA), and could require a re-IPL.

When buffer use reaches 80 percent of the limit specified at IPL, MCS issues the following message:

```
IEA405E WTO BUFFER SHORTAGE - 80% FULL
```

If the problem continues to build and MCS buffer use reaches its limit, MCS issues the following action message:

```
IEA404A WTO BUFFER SHORTAGE CRITICAL - 100% FULL
```

In addition, MCS puts all non-privileged tasks requesting a buffer (with a WTO macro instruction) into a wait state.

You should determine the reason for the buffer shortage so that you can correct the problem. Possible reasons are:

- A console is not ready and messages are accumulating in the console message buffers because
  - An intervention required condition exists
  - The console has been powered off
  - Some part of the path to the device is not operational, for example, an I/O interface is disabled
- A console is not in roll mode and messages are accumulating in the console message buffers.
- The limit specified at IPL (WTOBFRS initialization parameter) is too low for the amount of message traffic the system is producing.

Issue the DISPLAY CONSOLES command to determine the extent of the problem and the responsible console. The resulting display, Figure 2.8, includes a line indicating both the current number of buffers in use (410 in the figure) and the limit specified at IPL (500 in the figure). The display also indicates the number of buffers queued to each console. From Figure 2.8, it is clear that console 017 is the source of the problem. In the figure, the buffer limit at IPL seems adequate, therefore the console is probably experiencing a malfunction that is causing unpresented messages to accumulate in the console message buffers. Issue DISPLAY R to view unanswered requests such as intervention required.

When the number of buffers in use drops below 60% of the limit specified at IPL time, MCS issues the following message:

```
IEA406I WTO BUFFER SHORTAGE RELIEVED
```

1. All lines of an out-of-line multi-line status display that have not been presented occupy message buffers. Therefore, you should erase these displays when they are no longer needed.
2. The current buffer count can be larger than the IPL-specified limit. Even though the buffer count is greater than or equal to the limit, MCS always gives a privileged task a buffer unless CSA is exhausted. However, MCS puts all non-privileged tasks requesting a buffer into a wait state until the buffer count goes below the limit.
3. MCS does not use the WTOBFRS parameter value specified at IPL until either:
   - The hardcopy facility (SYSLOG or a console) becomes active

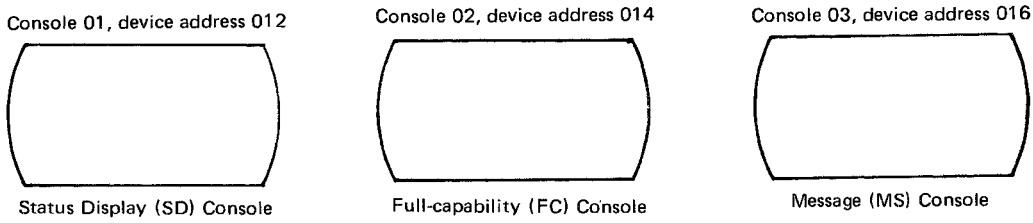
     or
   - When MCS receives control from the master scheduler after NIP processing is complete. At this point, consoles other than the master become active.

```
IEE250I 16.05.38 CONSOLE DISPLAY 508
WTO BUFFERS: CURR =   410        LIM =    500
CONSOLE/ALT  COND    AUTH    ID AREA    NBUF ROUTCD
SYSLOG       H       CMDS                    ALL
     01A/01B M       ALL     01 Z,A-B     5  ALL
     009/01F A       ALL     02           1  9
     014/016 A       ALL     03 Z,A      10  ALL
     008/00F N       INFO    04              ALL
     00F/008 N       INFO    05              ALL
     015/017 N       NONE    06              1-4, 7-8, 10
     016/014 N       ALL     07 Z,A          ALL
     017/015 A       NONE    08         400  1, 2, 10
     018/019 N       ALL     09 Z,A          ALL
     019/018 N       ALL     10 Z,A          ALL
     01B/61B N       ALL     11 Z,A-B        ALL
     01F/009 N       ALL     12              ALL
     1A0/3E0 N       ALL     13 Z,A          1-4, 8, 10
     2B0/2F1 N       IO,CONS 14 Z,A          ALL
     2F1/2F2 N       SYS,IO  15 Z,A          ALL
     2F2/3B0 N       SYS,IO  16 Z,A          ALL
```

Figure 2.8. Display Resulting From DISPLAY CONSOLES Command

Using a Console Cluster

You can control the system with a cluster of consoles. A cluster might consist of three console devices: one full-capability console, and two output-only display consoles:



The following groups of commands show how to set up and use the cluster. All of the commands are issued on console 02. The first group of commands assigns a dynamic display of system activity to console 01.

```
V O-012,CONSOLE  Make console 01 active
K V,USE=SD,L=1  Change its mode to output only for
  status displays
K A,23,L=1  Set up a 23-line display area
MR TR=A,L=01  Route TRACK output to new area
TR A,L  Periodically display job information
K T,UTME=10,L=1  Set time interval for updating periodic
  display to 10 seconds
K D,U,L=1  Cause immediate updating of the dynamic display
  on console 01 putting new time interval into effect.
```

The second group of commands assigns part of the ordinary system message traffic to console 03.

```
V O-016,CONSOLE,ROUT=(3,4,5,6)  Make console 03 active
  and route tape and disk messages to it
K V,USE=MS,L=3  Change its mode to output only
  for message stream
```

The third group of commands assigns the remainding ordinary message traffic to console 02 and status displays to area A of that console.

```
V O-014,CONSOLE,ROUT=(1,2,7,8,9,10,11,12,13,14,15)
  Make console 02 active and assign remaining messages to it
K A,8  Define an 8-line area for status display on
  console 02
```

Rather than issue all of the above commands from console 02, you could initialize the cluster automatically at system initialization by placing all of the commands (except the MSGRT (MR) command) in a COMMNDxx member of SYS1.PARMLIB. Each CONTROL (K) and TRACK (TR) command specified in a COMMNDxx member must include the L=cc operand to show which console the command is to affect.

You can change the functions assigned to the devices in the cluster. A convenient way to do this is to set up program function keys (PFKs) to automate the changes. You might set up PFKs 10, 11, and 12 so that:

- Pressing PFK 10 causes console 01 to present a display of outstanding requests.
- Pressing PFK 11 causes a display of the status of JES2 unit record devices and outstanding requests for JES2-device related operator action.
- Pressing PFK 12 restores the dynamic display produced by the TRACK command.

The following CONTROL N, PFK commands, entered on console 02, set up PFKs 10, 11, and 12:

The CONTROL command used to set up the assignment for PFK 10 is:

```
K N,PFK=(10,CMD='K V,USE=SD,L=1;K A,23,L=1;D R,L,L=1)
```

When you press PFK 10, the following commands are entered into the system:

```
K V,USE=SD,L=1  Delete current display and change console 01
  to status display mode
K A,23,L=1  Set up a 23-line display area on console 01
D R,L,L=1  Display ids and texts of all outstanding requests
```

The CONTROL command used to set up the assignment for PFK 11 is:

```
K N,PFK=(11,CMD='K V,USE=SD,L=1;
  K A,11,6,6,L=1;$DU,L=1A;D R,L=1B;$DO,L=1C')
```

When you press PFK 11, the following commands are entered into the system:

```
K V,USE=SD,L=1  Delete current display and change console 01
  to status display console
K A,11,6,6,L=1  Set up three display areas of 11, 6, and 6
  lines respectively
$DU,L=1A  Display JES2 unit record device status in area A
D R,L=1B  Display outstanding requests in area B
$DO,L=1C  Display JES2 device-related operator action in
  area C
```

The CONTROL command used to set up the assignment for PFK=12 is:

```
K N,PFK=(12,CMD='K V,USE=SD,L=1;
  K A,23,L=1;TR A,L,L=1;K T,UTME=10,L=1;K D,U,L=1')
```

When you press PFK 12, the following commands are entered into the system:

```
K V,USE=SD,L=1  Delete current display and change console
  01 to status display
K A,23,L1=  Set up 23-line display area
TR A,L,L=1  Periodically display job information on
  console 01
K T,UTME=10,L=1  Set time interval for updating periodic
  display to 10 seconds
K D,U,L=1  Cause immediate updating of the dynamic
  display on console 01 putting new time
  interval into effect
```

Chapter 3: Display Consoles

A display console is an operator's console that displays messages on a cathode-ray tube (CRT) instead of printing them on paper. Operating a system using a display console is similar to operating a system using a printer-keyboard console. You type commands and replies to system messages on an alphameric keyboard. With the exception of the TRACK command and some forms of the CONTROL command, the commands used are the same commands used on a printer-keyboard console.

Display consoles enable you to control the system more efficiently because:

- Messages are displayed faster, allowing you to respond more quickly to the system's needs.
- Special indicators highlight different types of messages; the system marks messages requiring action and keeps them on the screen until you respond to them.
- The light pen and the program function keyboard that are optional on certain consoles enable you to enter many commands at one time.

This chapter describes the physical characteristics common to all display consoles and presents a detailed description of the operation and use of display consoles.

The operational procedures in this chapter include all display console functions, such as program function key and selector (or light) pen capability. The text of the procedures is based on the 3277 type display console, which includes the 3036, 3056, and 3158 display consoles. The 3277 type console has, either as standard or optional features, all functions available to a display console. If you are using display console other than the 3277 type, refer to the appropriate appendix in this book that describes the physical characteristics and functional capabilities of your specific display console.

The following is a list of the display consoles and the appendix that describes their physical characteristics and functions:

- 3066 (Model 168) display console – Appendix B
- 2250 display console – Appendix C
- 2260 display console – Appendix D
- 3036, 3056, 3158, and 3277 display consoles – Appendix E

Display consoles can be used with the MVS system configuration. Through MCS more than one type of console device can be used with each system. For example, a 2250 display unit and several 3277 display stations can be used as operator consoles in the same system with a model 168 display console (3066).

Display Console Characteristics

The characteristic common to all display consoles is the cathode-ray tube (CRT). Most display consoles also include a device such as a typewriter keyboard that enables you to communicate with the system. Some display consoles have special devices, such as light pens and program function keyboards, that help the operator control the system. Figure 3.1 summarizes the features available on each display console.

	Full-Capability Mode	*Output-Only Mode*	*Light Pen*	*Audible Alarm*	*Program Function Keyboard (PFK)*	*Visual Alarm*
2250	YES	NO	YES*	YES*	YES*	NO
2260	YES	YES	NO	NO	NO	NO
3036	YES	YES	NO	YES	YES**	YES
3056	YES	YES	NO	YES	YES**	YES
3066	YES	NO	NO	YES*	NO	YES
3158	YES	YES	YES	YES	YES**	YES
3277(1)	NO	YES	NO	YES*	NO	NO
3277(2)	YES	YES	YES*	YES*	YES*	NO

*Optional feature
**Simulated on the top row of keyboard

Figure 3.1. Summary of Display Console Features

Display consoles are either full-capability consoles or output-only consoles. A full-capability console has both input and output capability; it can be used both to enter commands and to display messages. An output-only display console can display either messages or status displays but not both at the same time; it cannot be used to enter commands.

Except for the console keyboard, the physical characteristics of a display console are different than those of a printer-keyboard console. The following list describes the physical characteristics of a display console:

The Display Screen: The display screen of a full-capability console is divided into five functional areas (see Figure 3.2):

- Message Area: This area contains system and problem program messages and copies of certain operator commands. The size of the message area depends on the console.
- PFK Display Line: This line contains a display of program function key (PFK) numbers that are used when entering commands with the light pen.
- Instruction Line: This line is used to display system messages pertaining to control of the console. For example, if you make an error entering a CONTROL command, an error message appears in the instruction line.
- Entry Area: These two lines are used by the operator for entering commands and replying to messages.
- Warning Line: This line is used to warn the operator of conditions that could require action. For example, a warning message appears when the message area is full and one or more messages are waiting to be displayed.

The display screen of an output-only console consists of a message area and a warning line, as shown in Figure 3.2. The message area and warning line of an output-only console are the same as on a full-capability console.

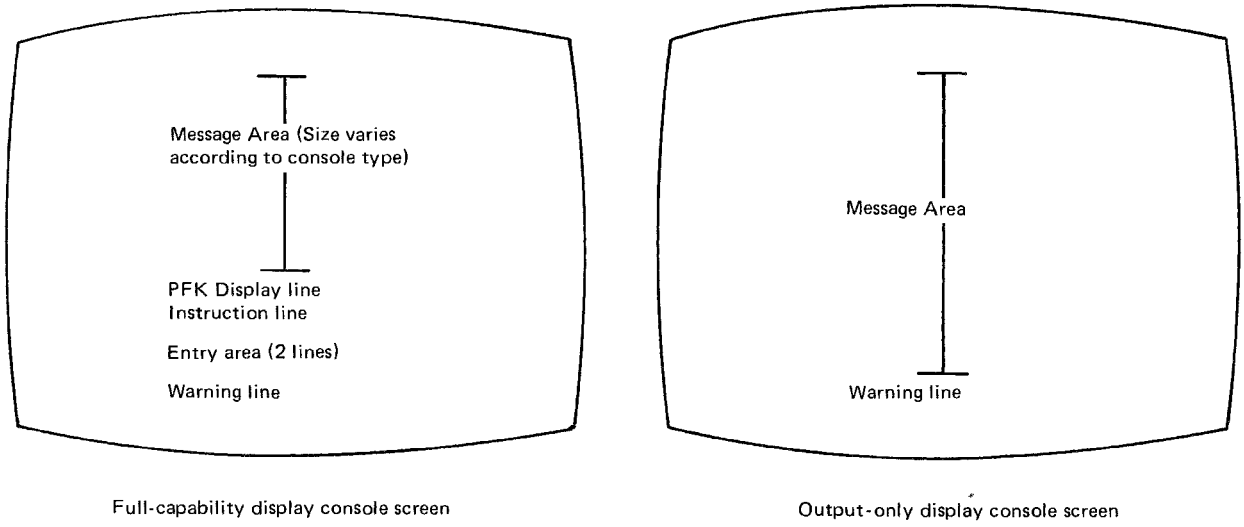


Figure 3.2. Comparison of the Display Screens of Full Capability Display Consoles and Output-only Consoles

The Keyboard: Each full-capability display console has a typewriter keyboard. The keyboard is used to type commands and responses to messages and to signal the system that information is being entered.

The Cursor: Each full-capability display console has a cursor. The cursor appears on the screen as a moveable point of light (it may be an underscore or a horizontal or vertical bar). The cursor designates the position on the screen that the system will examine for the next operator action. The action may involve positioning a typed character, entering a commands, requesting message deletion, or requesting a display. Special keys located on the console keyboard control cursor movement.

The Light Pen: The light pen (or selector pen) is a light-sensitive device that is available with certain display consoles. When the pen is placed over specific areas of the display console screen, it senses the light from the screen and signals the system. The system then determines the screen location over which the pen has been positioned and takes appropriate action. The action the system takes depends on how it has been programmed. In the operator console mode, this action may involve entering operator commands, deleting messages from the screen, canceling processes, or presenting displays.

The Program Function Keyboard: The program function keyboard (PFK) is an input device that is available on certain display consoles. Each PFK key can be associated with one or more operator commands; the operator can enter these commands by pressing the key.

Alarms: Visual and audible alarms are available on certain display consoles. The system activates an alarm when certain changes in conditions occur, such as an invalid CONTROL command.

Special Screen Characters

The system uses five special screen characters to indicate the status of certain screen messages. These special indicators appear in position three, four, or five of the lines in the message area:

- A vertical line (|) in position three indicates that required action has been taken for the message or that the message can be deleted.
- A horizontal bar (-) in position three indicates that the messages is for information only and requires no action by the operator.
- An asterisk (*) in position four indicates that the message is a system message that requires action by the operator.
- An at sign (@) in position four indicates that the message is a problem program message that requires action by the operator.
- A plus sign (+) in position five indicates that the message is a problem program WTO message.

Operator Action Differences

While the basic operating procedures are similar for all types of display consoles, the physical characteristics of each console require you to perform certain actions, such as the ENTER action, in different ways. The following text describes how to perform the ENTER, CANCEL, cursor detect, and selector (or light) pen detect actions. The descriptions of operating procedures later in this chapter refer to these actions.

Performing the ENTER Action

On a 3036, 3056, 3158, or 3277 Display Console: Information in the entry area is entered to the system when you press the ENTER key or select the ENTER indicator on the screen with the selector pen or cursor.

On a 2250 Display Console: Information is entered to the system when you press the END key. The cursor must be located in the entry area but need not be located at the end of the command.

On a 2260 Display Console: Information in the entry area is entered to the system when you position the cursor after the last character in the command and hold down the SHIFT key and press the ENTER key. The appearance of the EOM indicator (■) indicates that the system is receiving the command.

On a 3066 Display Console: Information in the entry area is entered to the system when you press the END key. The cursor must be located in the entry area but need not be located at the end of the command.

Performing the CANCEL Action

On a 3036, 3056, 3158, or 3277 Display Console: If you want to cancel a request, press the CANCEL (PA2) key.

On a 3066 Display Console: If you want to cancel a request, press the CANCEL key.

On a 2250 Display Console: If you want to cancel a request, hold down the ALT key and press the CANCEL (numeric 0) key.

On a 2260 Display Console: If you want to cancel a request, position the cursor to the first character position to the right of the START MI symbol, and press the SHIFT and ENTER keys *twice*.

Performing a Cursor Detect Action

On a 3036, 3056, 3158, and 3277 Display Console: To perform a cursor detect, position the cursor under the desired character and press the ENTER key.

On a 3066 Display Console: To perform a cursor detect, position the cursor under the desired character and press the END key.

On a 2260 Display Console: To perform a cursor detect, position the cursor under the desired character and hold down the SHIFT key and press the ENTER key.

Performing a Selector or Light Pen Detect

On a 3036, 3056, 3158, and 3277 Display Console or the 2250 display console, which uses the light pen, place the selector or light pen over the desired indicator in the control line. Then, press the pen against the screen (or press foot pedal on some 2250's so equipped).

Operating Procedures

The basic operating procedures–interpreting messages and entering commands–produce the same results on the typewriter console and the display console. However, display consoles require that you perform these basic procedures in slightly different ways:

- Interpreting Messages: Several special screen characters inform you of the status of certain messages. Also, the location of the message on the screen tells you certain things about the message; for example, messages in the instruction line usually pertain to console control, and messages in the warning line usually describe conditions that require action.
- Command Entry: To enter commands, you use the cursor, the entry area, and the typewriter keyboard on all consoles, and the PFK and the light pen on some consoles. You must become familiar with procedures for changing information in the entry area and entering commands in conversational mode (wherein the system writes the command in the entry area, and you change it as required before entering it).

Display consoles require you to perform several procedures that have no counterparts on typewriter consoles. These procedures include:

- Message Deletion: Because the display console screen has a limited number of lines, only a limited number of messages can be displayed on the screen at one time. To make room for additional messages, you must delete old messages and messages for which action has been taken.
- Controlling Status Displays: On typewriter and printer consoles, status displays are written to the console in response to the DISPLAY command. On display consoles, you may set up display areas, route displays to the areas, examine the displays, and erase the displays when you no longer need them; you may also display status displays in the general message area of the screen. You should also become familiar with the procedures for controlling time-interval updated displays that are updated whenever an operator-defined time interval elapses.
- Responding to Security Messages: When you reply to a WTOR (Write-to-Operator with Reply) message, your reply is written in the message area of your operator's console. When the system processes your reply, the following message appears:

```
IEE600I   REPLY TO  nn  IS:  reply
```

If the program issued the WTOR with a routing code of 9 (indicating a security classification) the text of your reply is replaced with the word 'SUPPRESSED' in the message area of the screen and in the message IEE600I.

After you learn to operate the system with one type of display console, it is easy to switch to another type. The programming is basically the same for each device. Because of certain hardware differences, however, some specific operations differ from device to device. Whenever you require information about a new device, read through the appropriate appendix for the specific display console in this book.

The following text describes the operating procedures used to control and modify the functions of a display console. Where procedures differ between types of display consoles, the differences and type of display console are noted.

In addition to a description of the CONTROL and MSGRT commands, which are used to control display consoles, the operating procedures described are:

- How to enter commands using the keyboard, selector or light pen, and program function keys
- How to change information in the entry area
- How to delete messages manually or automatically
- System status displays and how to use them
- Console modes (3277 type and 2260 display consoles)
- Error conditions

CONTROL Command

The CONTROL command (abbreviated K) controls the display console. Each function of this command is described in an appropriate place in this chapter. To request a summary of the CONTROL command operands and the functions that they perform, enter the following commands:

```
{DISPLAY}          C,K[,L={a  }]
{    D }               {cc  }
                       {cca }
```

C,K
specifies that a summary of Control command operands is to be displayed.

L=a, cc, cca
specifies the display area (a), console (cc), or both (cca) at which the display is to be presented. If you omit this operand, the display is presented in the first available display area on the console through which you entered the command; (unless routing instructions are in effect) see "MSGRT Command" below and in chapter 5.

For example, to display a summary of CONTROL command operands in display area A of console 10, enter:

```
D    C,K,L=10A
```

Note: Because the CONTROL command controls console functions rather than system functions, it is not recorded on the hardcopy log.

MSGRT Command

If you plan to route one type of display to the same display area or to the same output-only console, you can avoid entering the L=cca operand each time that you request the display by using the MSGRT (MR) command to predefine display routine instructions.

To establish routing instructions for DISPLAY commands, use the following form of the MSGRT command:

```
{MSGRT}  (d=(oprnd[,oprnd...])[,L= {a}])
{ MR  }                            {cc}
                                   {cca}
```

D
specifies that the display produced by the DISPLAY command, in conjunction with the operands specified by oprnd, is to be routed to the display area and console specified by the L operand.

oprnd
specifies one operand of the DISPLAY command that produces a status display (CONSOLES, DUMP, D,M,A,R,S,C,U,3850).

L=a, cc, cca
specifies the display area (a), console (cc), or both (cca) where the specified display is to be routed. If you omit this operand, all previously established DISPLAY command routing instructions are removed. If MR D=A, L=cca is specified, D TS and D J are also routed.

To establish routing instructions for the TRACK command and the STOPTR command, use the following form of the MSGRT command:

```
{MSGRT}  (TR=A[,L= {a}])
{ MR  }            {cc}
                   {cca}
```

TR
specifies that the display produced by the TRACK command, and the action of the STOPTR command, is to be routed to the console and display area specified by the L operand. If MR TR=A, L=cca is specified, TR TS,TR J, PT J,PT TS, and PT A are also to be routed.

L=a, cc, cca
specifies the display area (a), console (cc), or both (cca) where the specified display is to be routed or where the STOPTR command action is to take place. If you omit this operand, all previously established TRACK and STOPTR routing instructions are removed.

To establish routing instructions for the CONTROL (K) command, use the following form of the MSGRT command:

```
{MSGRT}  (K[,L= {a}])
{ MR  }         {cc}
                {cca}
```

K
specifies that CONTROL commands entered through the console used to enter the MSGRT command apply to the console specified in the L operands. This operand of MSGRT can be entered only through a non-CRT console.

L=a, cc, cca
specifies the display area (a), console (cc), or both (cca) where the action of the CONTROL command is to take place. If you omit this operand, all previously established CONTROL command routing instructions are removed.

Note: One MSGRT command can define routing instructions for all three commands (DISPLAY, TRACK, and CONTROL) or any combination of them; separate the definitions with a comma (see the example). The MSGRT command establishes routing instructions only for the console through which it is entered. You can establish different routing instructions for each console in the system.

To examine the routing instructions currently in effect, use the following form of the MSGRT command:

```
{MSGRT} [REF]
{ MR }
```

REF

specifies that a display of the MSGRT command operands currently in effect is to be presented in the entry area. REF is the default.

After MR REF, if the operand CONT appears in the display, there are additional routing instructions. To examine them press the entry key, or clear the entry area and enter:

```
MR  CONT
```

If there are not enough additional routing instructions to continue the display, the same display that results from MR REF appears.

Examples: If you plan to present all DISPLAY UNIT status displays in area B of console 12, and all TRACK A status displays in area A of the console that you use to enter all commands, establish routing defaults by entering:

```
MR(D=(U),L=12B),(TR=A,L=A)
```

Once this command is entered, all D U status displays that you request are presented in display area B of console 12, and all TR A displays are presented in display area A of the console where you enter the MR command. If you code the L=cca operand with a subsequent D U or TR A command, you override the MSGRT defaults for that command entry only.

To remove routing instructions previously established by the MSGRT command, code the applicable MSGRT command without the L operand. For example to remove routing instructions established for TRACK A commands, enter:

```
MR  (TR=A)
```

To examine the MSGRT operands currently in effect, enter:

```
MR REF
```

In response to this command, a display of MSGRT operands appears in the entry area and in the message area. You can change the routing instructions at this time by changing the command form that appears in the entry area (see "How to Change Information in the Entry Area" later in this chapter.

If no area ID (a) is specified on the routing instruction, an asterisk is displayed in the area ID position of the output.

The second way to remove routing instructions established by the MSGRT command is to issue the following form of the MSGRT command:

```
MR NONE
```

This command removes all routing instructions.

How to Change Information in the Entry Area

You might want to change information in the entry area to correct a typing error or to change a command form during conversational command entry or message deletion. Both conversational command entry and message deletion are described in subsequent sections. You might also want to blank the entry area without entering a command to the system.

Blanking the Entry Area

The ERASE INPUT Key: (3277 type display consoles only) To remove all of the data that you have typed in the entry area without causing it to be passed to the system, press the ERASE INPUT key. This key erases the entry area and moves the cursor to the first position in the entry area.

The CANCEL Action: You can also use the CANCEL action to erase the entry area. Performing this action or selecting the CANCEL indicator in the instruction line:

- Erases the entry area.
- Moves the cursor to the first position in the entry area.
- Rewrites the message area and the instruction line.
- Removes deletable-message indicators (if any are displayed).
- Removes message line numbers (if line numbers are displayed).

Note: The CANCEL key is marked PA2 on certain 3277 type keyboards.

Inserting Characters (3277 type only)

To insert one or more characters within data in the entry area:

1. Position the cursor at the character position following the point where the missing data should appear.
2. Press the INS MODE key (the console mode marker appears adjacent to the INSERT MODE indicator on the right side of the console).
3. Type in the missing data.
4. Press the RESET key to return the keyboard to its normal mode of operation.

Example: To insert the console identifier 10 in the following command:

```
TR  A,L=C
```

Move the cursor back to the C, press the INS MODE key, type in 10, and press the RESET key. The command then reads:

```
TR  A,L=10C
```

Note that the characters to the right of the inserted characters are shifted to make room for the inserted characters. If required, characters are shifted to the second line of the entry area.

Deleting Characters (3277 type only)

1. Position the cursor at the character to be deleted.
2. Press the DEL key.

Example: To delete the extra 0 from the following command:

```
TR  A,L=100A
```

Position the cursor at either 0 and press the DEL key. The command reads:

```
TR  A,L=10A
```

All characters are shifted to the left to fill the space occupied by the deleted character. Characters must be deleted one at a time.

Character Substitution

To substitute one or more characters for characters that you have already typed in the entry area:

1. Move the cursor back to the location of the first character that you want to change.
2. Type in the correct characters.

Example: If you type in the following reply to WTOR message:

```
R  22,'DISLPAY REQUESTED'
```

and then note (before performing the enter action) that the proper response text is 'DISPLAY REQUESTED', you can position the cursor under the L, and type PL. The message then reads:

```
R  22,'DISPLAY REQUESTED'
```

In the same example, if you decide that the correct response is 'NO', positioning the cursor under the D in DISPLAY and typing 'NO' leaves the following in the entry area:

```
R  22,'NO' PLAY REQUESTED'
```

To correct this situation, perform the CANCEL action to clear the display area and retype the entire command. Alternatively, you can position the cursor at the P and press the ERASE EOF key. This key erases the remainder of the entry area (from the cursor to the last character position), leaving the following in the entry area:

```
R  22,'NO'
```

How To Enter Commands

You can enter commands with the typewriter keyboard, the program function keyboard, or the selector pen (in conjunction with the PFK display line). Also, on the 3158 display console, commands associated with PFK numbers are entered by using the typewriter keyboard in a special way.

Entering Commands with the Typewriter Keyboard

This section contains information on entering commands through a 3036, 3056, 3158, or 3277 display console. Refer to the appropriate Appendix if you are using a different console.

Use the following procedures to enter commands and to reply to WTOR messages:

1. Move the cursor to the first position in the entry area.
2. Type in the command.
3. Enter the command by *one* of the following methods:
   - Perform the ENTER action.
   - Position the selector pen over the ENTER indicator in the instruction line.
   - Position the cursor under the ENTER indicator in the instruction line and press the ENTER key.

Moving the Cursor: Move the cursor to the first position in the entry area by one of the following methods:

- Press the cursor control keys.
- Press the tab key→|, the back-tab key|←, or the new line key↵|.
- Press the ENTER key when the cursor is in the entry area (any data in the entry area is passed to the system when the ENTER key is pressed).

Typing the Command: Type in the command just as you would on a typewriter console. As you type each character, the corresponding character appears in the entry area, and the cursor advances to the next character position. When the end of the first entry area line is reached, the cursor advances automatically to the the first character position of the next line, permitting continuation of the command. The maximum number of characters that can be entered is 126, but only one command can be entered at a time.

Most commands can be entered in either lowercase or uppercase. The system converts the commands to uppercase, if required. However, information within a command that is contained within single quotes (for example, a reply to a WTOR message) is not converted to uppercase by the system. If the system requires the information within the single quotes in uppercase, be sure to type it in uppercase when you enter the command.

Entering the Command: When you enter the command, the cursor must be in the entry area, but it need not be at the end of the command. Pressing the ENTER key or selecting the ENTER indicator causes the command to be read into storage and processed by the system. Commands other than the CONTROL (K) command disappear from the entry area and reappear in the message area. If the message area is full, the command may not appear immediately; to have it displayed, you may have to delete some messages. The CONTROL command is not moved to the message area; it remains in the entry area until the requested action takes place, and then it is removed from the screen.

Commands Entered with Errors: If you enter a CONTROL or MSGRT command with errors, the audible alarm sounds (if the console is equipped with an audible alarm), and the command appears in the entry area. The location of the cursor indicates the source of the error:

- If the error is an invalid operand, the cursor appears under the invalid operand:

  K X,N

- If the error is an invalid erase request, the cursor appears under the first invalid request.

  K E,31,19

- If the CONTROL command exceeds 126 characters, the cursor appears at location 127 in the entry area.

To correct any of these errors, use the procedures described earlier in this chapter under "How to Change Information in the Entry Area."

If the system detects an error in a command other than a CONTROL or MSGRT command, it writes the command in the message area with an appropriate error message. Follow the procedures indicated for the error message in the operator's library publication *System Messages* for the system you are using.

Entering Commands with the PFKs

The program function keyboard is a group of 12 keys (called PFKs) located near the operator console keyboard. On the 3158 display console, commands associated with PFK numbers are entered by using the typewriter keyboard in a special way (see "Entering Commands Associated with PFK numbers on the 3036, 3056, and 3158" in this chapter).

One or more PFKs might be available to you for entering commands. The PFKs are designated for operator command entry by the system programmer during system generation. You can define or redefine the commands for each key available to you (see "see Defining Commands in PFKs" in this section).

Each PFK is defined as conversational or nonconversational. The commands associated with a nonconversational PFK are entered immediately when you press the key. The commands associated with a conversational PFK are presented in the entry area, one at a time, when you press the key. You can make changes to them before you enter them.

How to Enter Commands in Nonconversational Mode

Press the key associated with the commands that you want to enter. All of the commands are entered in the order in which they were associated with the key, just as if you had typed each command and performed the ENTER action.

Note: PFKs that are defined as conversational function in the conversational mode even though the console is in nonconversational mode. Use these keys as if you were in conversational mode as described under "How to Enter Commands in Conversational Mode."

How To Enter Commands in Conversational Mode

1. Press the PFK associated with the command that you want to enter, causing the first command associated with the key to appear in the entry area.
2. According to your requirements:
   - Enter the command by performing the ENTER action or by selecting the ENTER or * ENTER indicator with the selector or light pen; the next command associated with the PFK (if any) appears in the entry area.
   - Change the command from the keyboard before entering it (see "How to Change Information in the Entry Area" in this chapter).
   - Cancel the command that appears in the entry area by performing a CANCEL action or by selecting the CANCEL indicator with the selector or light pen; the next command associated with the PFK (if any) appears in the entry area.
   - Cancel the request initiated by the first press of the PFK by pressing any PFK while the command is still in the entry area.

In conversational mode, each command associated with the PFK is presented in the entry area where it can be entered, altered and entered, or canceled. Altering a command in the entry area is effective only for the command entry in progress; the original definition is retained for future use of the PFK. To permanently redefine a PFK, use the procedures described later in this section under "Defining Commands for PFKs."

PFK Errors

If you press a PFK that was not designated for command entry, the following message appears in the instruction line:

```
IEE721I  PFK nn NOT SUPPORTED
```

If you press a PFK for which support has been requested but not defined, the following message appears in the instruction line:

```
IEE722I  PFK nn NOT DEFINED
```

How to Determine which Commands Have Been Associated with Each PFK

Use the DISPLAY PFK command to determine the commands associated with each PFK:

```
{DISPLAY PFK}
{    D      }
```

PFK
specifies that a display of the commands defined for each PFK is to be presented in the message area.

For example, to request a display of the commands associated with each key, enter:

```
D  PFK
```

In response to this command, the following message appears in the message area:

```
IEE724I  PFK   DEFINITION
 KEY#    CONV  DEFINITION
  •       •       •
  •       •       •
  •       •       •
```

The definitions for each key appear under the headings. The definitions always refer to the console on which they are displayed; the DISPLAY PFK command cannot be routed to a display area or to another console.

Defining Commands for PFKs

Use the CONTROL N,PFK command to define commands for PFKS:

```
{ CONTROL }N,PFK=( nn1{,CMD='text[;text...]'} )[,CON=  { Y }]
{    K    }         {,KEY=nn2[,nn2...]    }            { N }
```

N,PFK
specifies that a PFK command definition is to be altered.

nn1
specifies the number of the PFK being defined. The nn1 value must be the number of a PFK designated for command entry during system generation.

CMD
specifies that the text of one or more commands are to be associated with PFK nn1.

'text[;text...]'
is the text of the operator commands to be associated with PFK nn1. Up to 101 characters can be included within the quotes; if more than one command is to be associated with a PFK, the commands must be separated by a semicolon (do not put a semicolon after the last command). Text characters may be entered in uppercase or lowercase; the system converts all characters to uppercase.

KEY
specifies that the commands associated with other PFKs are to be associated with PFK nn1.

nn2[,nn2...]
is the number of the PFKs whose commands are to be associated with PFK nn1. Up to 52 key numbers (numbers can be repeated) can be included in the list. Separate key numbers with a comma.

CON
specifies whether conversational mode of command entry is in effect. (Conversational mode is described in this section under "Conversational and Nonconversational Mode.")

Y
specifies that conversational mode of command entry is in effect.

N
specifies that conversational mode of command entry is not in effect (non-conversational mode is in effect).

Examples: To associate a START READER command with PFK 5, enter:

```
K  N,PFK=(5,CMD='S RDR.001'),CON=N
```

To associate a START READER and a START WRITER command with PFK 5, enter:

```
K  N,PFK=(5,CMD='S RDR.0001;S WTR.292'),CON=N
```

All 'text' characters in the CONTROL N,PFK command are converted to uppercase during the definition process. If a command must be entered in lowercase (this may be true of a reply to a WTOR, for example), then the command cannot be entered by means of the PFK command entry function.

If PFK 3 is associated with commands S RDR.001 and S WTR.292, and PFK 4 is associated with the command S INIT,,,ABC, you can associate all three of these commands with PFK 5 by entering:

```
K  N,PFK=(5,KEY=3,4),CON=Y
```

The commands associated with PFK 5 would now be S RDR.001; S WTR.292, and S INIT,,,ABC, *in that order* (because that is the order in which the commands were defined for each key in the list, and that is the order in which the keys were listed after the KEY parameter). Up to 52 key numbers can be included in the list (key numbers can be repeated) if the limits of the entry area are not exceeded.

Note: A PFK defined as a list of keys cannot be included in a list of keys assigned to another PFK. For example, if PFK 5 is associated with a list of keys (KEY=3,4) and you attempt to associate PFK 6 with a list of keys including PFK 5 (KEY=1,2,5), the system rejects the request (see "PFK Definition Errors" in this section).

Redefining PFKs: Use the same procedures for redefining PFKs that you use for defining PFKs. The new definition replaces the current definition.

Deleting a Definition: To return a PFK to the undefined state, enter the CONTROL N,PFK command with the CMD keyword, but include no data between the quotes; for example:

```
K  N,PFK=(5,CMD='')
```

Conversational and Nonconversational Mode: The CON parameter of the CONTROL N,PFK command specifies conversational or nonconversational mode. In nonconversational mode, the commands associated with a key are entered immediately when the key is pressed. For example, if you define PFK 5 as follows:

```
K  N,PFK=(5,CMD='D U,L=09A'),CON=N
```

then pressing PFK 5 has the same effect as typing D U,L=09A and pressing the ENTER key. On the other hand, if you specify conversational mode by entering:

```
K  N,PFK=(5,CMD=''D U,L=09A'),CON=Y
```

then pressing PFK 5 causes the command D U,L=09A to appear in the entry area (but no enter action takes place). The command can be altered, entered, or canceled according to your requirements. If the CON parameter is omitted, CON=N is assumed.

Cursor Location: In conversational mode, the cursor is normally positioned under the third nonblank character when the command appears in the entry area. If you want the cursor to appear in a different location, when you define the command, type an underscore before the character under which the cursor is to appear. For example, if you enter:

```
K  N,PFK=(5,CMD='D     U,L=_XXX'),CON=Y
```

pressing PFK 5 causes the following to appear in the entry area:

```
D  U,L=XXX
```

Note that space occupied by the underscore is deleted in the actual command.

PFK Definition Errors: If you enter an invalid CONTROL N,PFK command, the audible alarm sounds (if the console is equipped with an audible alarm), and the command is written to the entry area. The location of the cursor indicates the source of the error:

- If the cursor is positioned under the first letter of a keyword (CMD,KEY,PFK, or CON), that keyword or its trailing equal sign is incorrect.
- If the cursor is positioned under the number of the PFK being defined, that number is either not a numeric character, not a number of a PFK that was designated for command entry during system generation, or it is being associated with a list of key numbers when it is already contained within a list of key numbers.
- If the cursor is positioned under a number following the KEY operand, the key number indicated is either a non-numeric character, the number of the PFK that is being defined, the number of a PFK that has already been defined as a list of key numbers, or it is the number of a PFK that was not designated for command entry during system generation.

To correct these errors follow the procedures described under "How to Change Information in the Entry Area" in this chapter.

How to Display the PFK Numbers

Use the CONTROL, D,PFK and CONTROL E,PFK commands to display and erase the numbers in the PFK line:

```
{CONTROL}     D,PFK
{   K   }     E,PFK
```

D,PFK
specifies that the numbers of the PFKs designated for command entry are to be displayed in the PFK display line.

E,PFK
specifies that the numbers are to be erased from the PFK display line.

Example: To request display in the PFK display line (this line is located immediately above the instruction line), enter:

```
K              D,PFK
```

In response to this command, a display similar to the following appears in the PFK display line:

```
1   2   3   4   5   6   7   8   9   10  11  12
```

Only those numbers that have been designated for PFK command entry appear in the display. Once you have requested this display, you can leave it on the screen; the PFK display line is not used for any other purpose, even when the key numbers are not displayed. To erase the display, enter:

```
K  E,PFK
```

Entering Commands Associated with PFK Numbers on the 3036,3056, and 3158

The top row of the keyboard can be used as a program function keyboard. Keys 1-9 serve as PFK keys 1-9, and keys 0,-,and & serve as PFK keys 10,11, and 12 respectively.

When you press the REQ or PFK SEL key, the top row of the keyboard assumes its PFK function and the letters PFK appear on line 25. Now press a key in the top row to enter the PFK command(s) associated with it. Before you press one of the keys that serve as PFK keys, the procedure can be canceled (and the top row of the keyboard returned to its normal function) by pressing the keyboard RESET key.

After taking into account these considerations, the description of how to use PFKs under the heading "Entering Commands Associated with PFKs" applies to these display consoles.

Entering Commands with the Selector Pen or Light Pen

The selector pen or light pen is used in conjunction with the PFK display line to enter commands. The numbers appearing in the display line represent PFK numbers, and selecting a number with the selector pen has the same effect as pressing a PFK. The PFK numbers available for selector pen command entry must be designated for PFK command entry during system generation. If your console has both a program function keyboard and a selector pen, both forms of command entry can be used.

How to Enter Commands with the Selector Pen or Light Pen in Nonconversational Mode

1. Display the PFK numbers in the PFK display line by entering the CONTROL D,PFK command.
2. Select the PFK number associated with the command that you want to enter.
3. Position the selector pen or light pen over the selected number, causing immediate entry of the command.

In nonconversational mode, all of the commands associated with a PFK are entered in the order in which they were associated with the key number. All commands (except CONTROL commands) appear in the message area when screen space is available. No commands appear in the entry area.

How to Enter Commands with the Selector Pen or Light Pen in Conversational Mode

1. Display the PFK numbers in the PFK display line by entering the CONTROL D,PFK command.
2. Select the PFK number associated with the command you want to enter.
3. Position the selector pen over the selected number, causing the first command associated with the PFK number to appear in the entry area.
4. According to your requirements, you may:
   - Change the command from the keyboard before entering it (see "How to Change Information in the Entry Area" earlier in this chapter).
   - Cancel the command in the entry area by performing a CANCEL action or by positioning the selector pen over the CANCEL indicator in the instruction line; the next command (if any) appears in the entry area.
   - Cancel the request initiated by the first selection of the PFK number by positioning the selector pen over any PFK number while a command associated with the first key number is still in the entry area.

In conversational mode, each command associated with a PFK number is presented in the entry area, one command at a time, where it may be entered as is, altered and entered, or canceled. Altering a command in the entry area is effective only for the command entry in progress; the original definition is retained for the key number. To permanently redefine a PFK number, use the procedures described in this chapter under "Defining Commands for PFKs".

How to Define Commands for Selector Pen or Light Pen Command Entry

Use the procedures described under the heading "Defining Commands for PFKs." On consoles equipped only with a selector pen, these procedures define commands for the numbers in the PFK display line; on consoles equipped with both a PFK and a selector pen, the procedures define commands for the PFKs and the numbers in the PFK display line.

How to Delete Messages

As programs execute during system operation, the message area of the screen gradually fills with messages. To make room for more messages, you should delete nonaction messages and messages for which action has been taken.

You may delete messages from the screen *manually* using the CONTROL E command, the cursor, or the selector pen, or you can request that the system delete messages *automatically* in automatic mode, roll mode, or roll-deletable mode. ***Note:*** Procedures for deleting system status displays are described in the section of this chapter called "System Status Displays" under the heading "How to Erase Status Displays."

Manual message deletion, by selector pen or cursor, is useful for providing screen space quickly; the CONTROL E command is more flexible for deleting groups of messages selectively. Automatic message deletion is most useful when messages appear frequently, and when it is important that messages not back up on system queues while waiting for screen space.

Message deletion is also *conversational* or *nonconversational*. In conversational mode, the system allows you to verify your deletion request before the messages are removed from the screen. In nonconversational mode, your deletion request causes the designated messages to be deleted immediately. Conversational message deletion is in effect when the console is initialized. Use the CONTROL S command to request conversational or nonconversational message deletion:

```
{CONTROL}     S  [ ,CON=  {Y} ]
{   K   }        [ ,REF   {N} ]
```

S
specifies that console specifications are to be altered or referenced.

CON
specifies that conversational message deletion is requested or cancelled.

Y
specifies that conversational message deletion is requested.

N
specifies that conversational message deletion is cancelled (nonconversational message deletion is to go into effect).

REF
specifies that the CONTROL S command values are to be displayed in the entry area in CONTROL command form.

Manual Message Deletion in Nonconversational Mode

How to Delete Nonaction Messages with the Selector Pen or Light Pen in Nonconversational Mode

To delete the designated message *and any nonaction messages above it,* position the pen anywhere within the nonaction message to be deleted. All messages are moved toward the top of the screen to fill the lines that were occupied by the deleted messages. As message lines become available at the bottom of the message area, any messages waiting for screen space are moved to fill them.

How to Delete Nonaction Messages Using the Cursor in Nonconversational Mode

1. Move the cursor to any position within a nonaction message in the message area.
2. Perform an ENTER action, causing the indicated message *and all nonaction messages above it* to be removed from the screen.

In addition to deleting the messages, the enter action blanks the entry area and repositions the cursor to the first data entry position. All messages in the message area are moved toward the top of the screen to fill the lines that were occupied by the deleted messages. As message lines become available at the bottom of the message area, they are filled by any messages waiting for screen space.

Note: This procedure does not apply to a 2250 display console.

How to Delete Action Messages Using the Cursor in Nonconversational Mode

1. Move the cursor to the asterisk or at sign (@) of the action message.
2. Perform an ENTER action, causing only the designated action message to be deleted.

In addition to deleting the message, the ENTER action blanks the entry area and repositions the cursor to the first data entry position. Messages below the deleted message are moved up on the screen to fill the line that was occupied by the deleted message.

On a 2260 Display Console:

1. Select the action message you want to delete.
2. Place the console in hold mode by holding down the SHIFT key and pressing the ENTER key.
3. Position the cursor at the asterisk or at sign (@) in the message you want to delete.
4. Perform an ENTER action.

How to Delete Action Messages With the Light Pen in Nonconversational Mode - 2250 Only

To delete action messages, position the pen at the asterisk or at sign (@) in front of the action message you want to delete. This deletes the designated action message. All messages below the deleted message are moved toward the top of the screen; as message lines become available at the bottom of the message area, any messages waiting for screen space are moved in to fill them.

How to Delete Messages with the CONTROL (K) Command in Nonconversational Mode

Use the CONTROL E command to delete one message, a segment of messages, or flagged messages:

```
{CONTROL}    E  [,nn[,nn]]
{   K   }       [,SEG    ]
                [,F      ]
```

E
specifies that messages are to be removed from the screen.

nn[,nn]
specifies that a single message (nn) or the deletable messages within a range of messages (nn,nn) are to be removed from the screen. The value nn, when used alone, must be a number from one to the highest message number on the screen; the values nn,nn must be a range of numbers in ascending order.

SEG
specifies that the deletable messages in the predefined message segment are to be removed from the screen.

F
specifies that all flagged messages (messages marked with a vertical or horizontal bar in position three) are to be removed from the screen.

How to Establish the Scope of Message Deletion

```
{CONTROL}    S,SEG=nn
{   K   }
```

S
specifies that console specifications are to be altered or referenced.

SEG
specifies that the size of the message segment is to be altered.

nn
specifies the number of lines to be included in the segment of messages deleted when a CONTROL E,SEG command is entered. The nn value is a number from one to the number of lines in the message area.

Examples: To delete the message at line 10, which appears on the screen as follows:

```
10  IEE334I HALT EOD SUCCESSFUL
```

enter:

```
K  E,10
```

To delete the nonaction messages from a segment of messages, enter:

```
K  E,SEG or K E
```

To delete the nonaction messages in message lines 4-10, enter:

```
K  E,4,10
```

Note: Message numbers can help you determine which messages you want to delete; see "Numbering Messages" in this chapter.

To delete all flagged messages (messages marked with a vertical or horizontal line in position 3), enter:

```
K  E,F
```

To set SEG equal to 10 lines, enter:

```
K  S,SEG=10
```

To determine the current value of SEG, enter:

```
K  S,REF or K S
```

Manual Message Deletion in Conversational Mode

Conversational message deletion allows you to verify all messages that have been selected for deletion by cursor, selector pen, or CONTROL command. Conversational mode it requested by means of the CONTROL S, CON=Y command. To delete messages in conversational mode:

1. Follow the procedures described for nonconversational message deletion by selector pen, cursor, or CONTROL command.
2. After you enter the deletion request, a vertical line appears in position three of each deletable message (all other vertical lines are temporarily removed from the screen), and the following message appears in the instruction line:

```
IEE157E DELETION    REQUESTED
```

3. Message line numbers are written for all messages on the screen, and the deletion request appears in the entry area in CONTROL command form.

What Appears in the Entry Area: If your deletion request is made by selector pen or cursor, or is a CONTROL E,nn,nn or CONTROL E,SEG command, the CONTROL E,nn,nn, command form appears in the entry area. For example, if SEG has been defined as 10, and you enter CONTROL E,SEG, the following appears:

```
K  E,1,10
```

If you made your deletion request by positioning the selector pen or the cursor on the fifth message line, that message and all nonaction messages above it are marked with vertical bars, and the following appears in the entry area:

```
K  E,1,5
```

If your deletion request was a CONTROL E,F command, the following appears in the entry area:

```
K  E,F
```

Verifying the Request: The system now requires verification of whatever deletion request appears in the entry area. To provide this verification:

1. Study the command in the entry area and the messages marked with vertical bars. Make certain that the indicated messages are the ones that you want to delete. If you want to make any changes, use the procedures described in this chapter under "How to Change Information in the Entry Area." If you want to cancel the deletion request, perform a CANCEL action.

2. When the command is in the proper form, perform an ENTER action or position the selector pen or cursor on the ENTER indicator. If you originally used the selector pen to designate the messages to be deleted, you can enter the command by positioning the selector pen over the same line again.

The messages selected for deletion are removed from the message area, and any remaining messages are moved up toward the top of the screen. Whether you enter the command in the entry area or cancel it, message line numbers are removed, and any flags that existed prior to the request are restored. Also, the entry area is blanked, and the cursor is repositioned to the first data entry position.

Note: If roll mode or roll-deleteable mode is specified, message deletion is handled as if the console were in nonconversational mode.

Automatic Message Deletion

Automatic message deletion is a means of deleting messages from the screen without operator intervention. There are three modes of automatic message deletion:

- *Automatic Mode:* In this mode, all flagged messages are removed from the screen whenever the screen becomes full.
- *Roll Mode:* In this mode, a specified number of messages are deleted if the screen is full when a specified time interval elapses.
- *Roll-deletable Mode:* In this mode, the flagged messages in a specified group of messages are deleted if the screen is full when a specified time interval elapses.

Automatic Mode

Automatic mode of message deletion is in effect when the console is initialized. In automatic mode, messages are deleted whenever the message area is full and a message is waiting to be displayed, or when a status display is overlaying messages in the bottom portion of the message area. Use the CONTROL S command to request or cancel automatic message deletion:

```
{CONTROL}     S  [,DEL=  {Y}]
{  K    }        [,REF   {N}]
```

S
specifies that console specifications are to be altered or referenced.

DEL
specifies that message deletion mode is to be changed.

Y
specifies that automatic mode of message deletion is to go into effect.

N
specifies that automatic mode of message deletion is cancelled. Messages must be removed manually.

REF
specifies that the current values for the CONTROL S operands (including DEL) are to be displayed in the entry area.

Example: To request automatic message deletion, enter:

```
K  S,DEL=Y
```

Messages flagged with a vertical line (|) in position three are the only messages removed under automatic mode. These messages include:

- Action messages for which the action has been taken.
- System or problem program messages that are marked deletable by the issuer.
- Messages that are indicated as deletable at job step end.
- WTOR messages that have been answered.
- WTOR messages that have not been answered, but are associated with a job step that has ended.

If there are no messages marked with a vertical line when a message is waiting to be displayed, the following message appears in the warning line:

```
IEE159E  MESSAGE WAITING
```

You must then delete messages by using the CONTROL command, cursor, or the selector pen.

Roll Mode and Roll-deletable Mode

Roll mode is a form of automatic message deletion where the system deletes a specified number of messages from the screen when a time interval elapses. Deletion occurs only if the screen is full and messages are waiting to be displayed.

Roll-deletable mode is a form of automatic message deletion where the system deletes the flagged messages from a specified number of messages each time that a time interval elapses. Deletion occurs only if the screen is full and messages are waiting to be displayed.

Roll mode and roll-deletable mode are requested by means of the CONTROL S command:

```
{CONTROL}     S  [ ,DEL=  {R}  ]
{  K   }         [        {RD} ]
                 [        {N}  ]
                 [ ,RTME=nnn   ]
                 [ ,RNUM=nn    ]
                 [ ,REF        ]
```

S
specifies that console specifications are to be altered or referenced.

DEL
specifies that the message deletion mode is to be changed.

R
specifies that roll mode is in effect.

RD
specifies that roll-deletable mode of message deletion is in effect.

N

specifies that roll mode or roll-deletable mode (whichever is in effect) is cancelled. Manual message deletion is then required.

RTME=nnn

specifies the number of seconds in the time interval between message rolls. The nnn value can be any decimal number from 1 to 999.

RNUM=nn

specifies the number of message lines included in the message roll. The nn value is a decimal number form one to the number of lines in the message area.

REF

specifies the current value for the CONTROL S operands is to be displayed in the entry area in CONTROL command form.

Example: To request that roll mode go into effect and that ten messages be rolled every 30 seconds, enter:

```
K  S,DEL=R,RNUM=10,RTME=30
```

When a console is operating in roll mode or roll-deletable mode, messages are *not* numbered. Instead, a two-digit number is displayed in the first new message line on the screen after each roll. This number indicates the number of messages waiting to be displayed (including any messages that are hidden by a status display). If the number of messages waiting to be displayed exceeds 99, AA is displayed.

Recommendation: Roll mode is not recommended for normal operator console use because:

- Messages may be lost before you see them.
- System overhead may be increased.

One possible use for roll mode is for a console used to monitor message traffic in a tape or disk library.

Establishing Several Specifications at One Time

You need not enter a separate CONTROL command for each specification; you can define all or any portion of the specifications whenever you enter a CONTROL S command. For example, if roll mode is in effect with five messages being deleted every 20 seconds, and you want to change to roll-deletable mode with the same number of messages being deleted every 30 seconds, enter:

```
K  S,DEL=RD,RTME=30
```

In this case, roll-deletable mode becomes effective, and the time interval changes to 30 seconds; the roll number (RNUM), segment (SEG), and conversational mode (CON) remain unchanged.

Checking the Specifications

To determine which specifications are in effect, enter:

```
K  S,REF
```

In response to the CONTROL S,REF command the following display appears in the entry area:

```
K  S,DEL=x,SEG=xx,CON=x,RNUM=xx,RTME=xxx
```

In the actual display, each x is replaced by the specifications currently in effect. You can change any specification at this time by following the procedures prescribed for changing information in the entry area (see "How to Change Information in the Entry Area") earlier in this chapter.

Note: When you enter the CONTROL S,REF and you want to retain the values as they are shown, perform a CANCEL or ENTER action.

Initial Console Specifications

The following table summarizes the message deletion specifications in effect when a console is initialized.

Console Type	DEL	SEG	CON	RTME	RNUM
3036, 3056, 3158, 3277-2	Y	9	Y	46	19
3066	Y	20	Y	60	30
2250	Y	35	Y	94	47
2260	Y	8	Y	16	8

Numbering Messages

Message numbers are useful in determining the range of messages to delete with the CONTROL E,nn command and the CONTROL E,SEG command. You can request that message numbers appear in positions one and two of the message area lines. Consecutive numbers appear for each message line, including continuation lines, for all message area messages except status displays. A numbered message appears as follows:

```
12  IEE334I HALT EOD SUCCESSFUL
```

Use the CONTROL D,N and CONTROL E,N commands to display and erase message numbers:

```
{CONTROL} {D,N[,HOLD]}
{  K    } {E,N       }
```

D
specifies that information is to be displayed on the screen.

N
specifies that consecutive numbers are to be displayed in character positions one and two of each message area line containing a message; the numbers are to be removed from the screen after the operator deletes a message or performs a cancel action.

N,HOLD
specifies that consecutive numbers are to be displayed for each message on the screen, and that the messages are to be renumbered after each message deletion that the operator performs.

E
specifies that information is to be removed from the screen.

N
specifies that message numbers requested by the CONTROL D,N,HOLD command are to be removed from the screen.

Example: To request continual message numbering, enter:

```
K  D,N,HOLD
```

To cancel continual message numbering, enter:

```
K  E,N
```

Note 1: Automatic message deletion (in automatic mode, roll mode, or roll-deletable mode) also stops message numbering requested by the CONTROL D,N,HOLD command.

Note 2: Because the CRT console screen can be "burned" by the number images, it is recommended that you do not have the messages numbered all of the time. When you are in conversational mode and delete messages by the CONTROL command, all messages are temporarily numbered so that you can verify that you have entered the correct delete command.

System Status Displays

A status display is a formatted, multiple-line display of information about some part of the system. It is written to the operator's console in response to a DISPLAY, TRACK, or subsystem command.

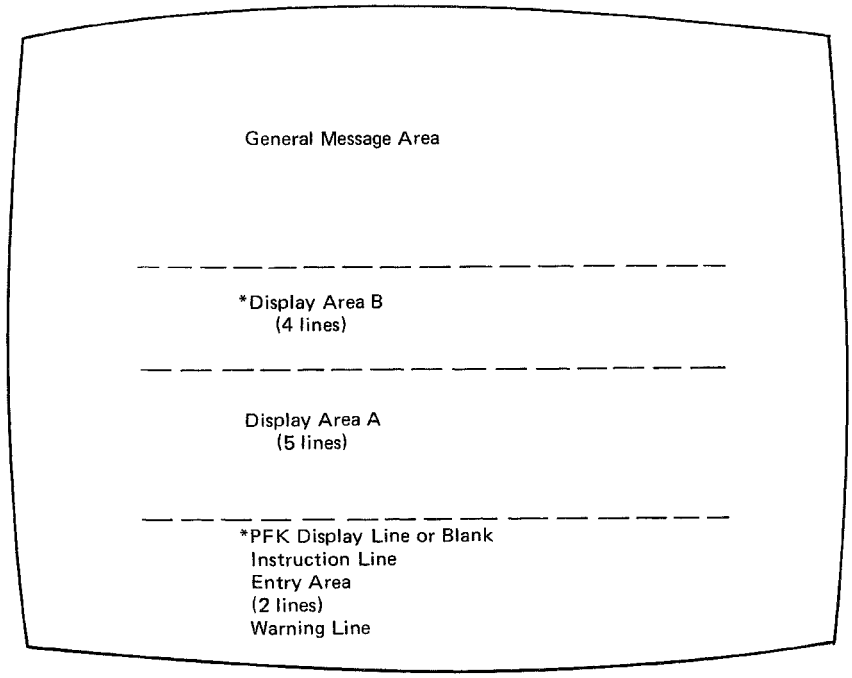
Status displays can be used to obtain information such as the status of system devices and the identification of the jobs active in the system. This information can help you decide how best to use system resources. The operands of the DISPLAY command are described in Chapter 5.

On display consoles, status displays are normally presented in display areas set aside for their use. A status display is either static or dynamic. A status display is static if it remains the same until it is removed from the screen. A dynamic display is updated by the system each time a preset time interval elapses. The different procedures required for dynamic displays are described in this section under "Dynamic Status Displays."

Display Areas

Display areas are blocks (minimum size is 4 lines) of screen lines designated to receive status displays. They enable you to define the location and the number of screen lines that are used for status displays. Display areas are defined beginning with the bottom lines of the message areas and working toward the top. Figure 3.3 shows the screen format for all of the display consoles; two display areas are defined for the screen. The first (bottom-most) area is six lines long, and the second is four lines long. Status displays can be routed (using the L=cca operand of the DISPLAY or TRACK command) to area A or B, or to the general message area.

The alphabetic display area identifiers are assigned by the system. The bottom-most area is assigned identifier A and additional areas are assigned identifiers in alphabetic order, working toward the top of the screen. The identifier Z always refers to the portion of the message area that is not assigned to a display area.



*Not available on a 2260 display console

Figure 3.3. Screen Format Showing Two Display Areas

How to Establish Display Areas

Display areas can be established during system generation; you can alter the original specification, or establish a new specification using the CONTROL A command:

```
{CONTROL}     A  [ ,nn[,nn...] [,L=cc] ]
{   K   }        [ ,NONE[,L=cc]       ]
                 [ ,REF[,L=cc]        ]
```

A
specifies that display area specifications are to be altered or referenced.

nn[,nn...]
specifies the number of message lines in each display area. The first nn defines the bottom area on the screen; additional nn's define areas working toward the top of the screen. The minimum number of lines in one display area is four; the total of all nn's cannot exceed the number of lines in the message area of the screen.

NONE
specifies that no display areas are defined for the console.

REF
specifies that the display area operands are to be displayed in the entry area in CONTROL command form.

L=cc
specifies the console whose display areas are to be changed or referenced. Use this operand when changing display area specifications for output-only consoles. If this operand is omitted, the command affects the display areas of the console through which you enter the command.

Examples: To establish two display areas, the first of eight lines and the second of four lines, enter the following command:

```
K    A,8,4
```

You can check the display area specifications in effect at any time by entering:

```
K    A
```

This command displays the specifications in the entry area in the CONTROL command form. For example, if three display areas of eight, four, and four lines are defined for a screen, entering K A causes the following to appear in the entry area:

```
K    A,08,04,04
```

You can then change the specifications by following the procedures described elsewhere in this chapter, under the heading "How to Change Information in the Entry Area". To change the specification in the above example to two display areas of four and six lines, position the cursor at the 8, type in 4,6; blank the remainder of the area, and enter the command.

To remove all display area definitions, enter:

```
K    A,NONE
```

Using Display Areas

Display areas can be thought of as addressable blocks of message area lines that overlay parts of the message area. The blocks of lines assigned to a display area function as message area lines until a status display is written to them. When this happens, any messages occupying the lines of the display area are replaced by the status display. If these messages are general operator messages, they are not lost; they reappear, higher up on the screen if screen lines above the status display become available for their use. When messages are overlayed by a status display, the following message appears in the warning line:

```
IEE160I UNVIEWABLE MESSAGE
```

Erasing a display from a display area restores the display area lines to general message use. Unless there are other status displays above the restored display area, any messages that were overlayed by the erased display reappear. If there are status displays in display areas above the restored area, the lines of the restored area remain blank. General messages are not displayed in any line in the message area below a display area containing a status display.

Note: It is good practice to erase status displays from display areas when the displays are no longer required. The undisplayed frames of lengthy status displays can tie up buffer space which may be required for other messages.

Using the Location Operand

Use the location operand (L=cca) of the CONTROL, DISPLAY, STOPTR, and TRACK commands to route status displays to specific display areas on the console you are using or to route displays to output-only consoles. Alphabetic display area identifiers (a) are assigned by the system; identifier Z is assigned to the portion of the message area not assigned to a display area. Numeric console identifiers (cc) are also assigned by the system. You can obtain the console identifier and display area identifiers for each console by issuing a DISPLAY CONSOLES command. The routing location operand requesting a console other than the requesting console can be issued only through the master console or through consoles assigned to command group CONS (console control).

The routing location operand is used with the CONTROL,TRACK,STOPTR, and MSGRT commands. It can also be used with the following forms of the DISPLAY command described in Chapter 5.

```
{ DISPLAY }   ( CONSOLES )
{   D     }   ( DUMP     )
              ( DMN      )
              ( M        )
              ( TS       )
              ( J        )
              ( A        )
              ( R        )
              ( SLIP     )
              ( U        )
              ( C,K      )
              ( 3850     )
```

Where Status Displays are Presented

If you use the location operand with the DISPLAY or TRACK command, the system presents the status display on the console and in the display area that you specify. If the area is already occupied, the new status display replaces the old display (except that dynamic displays are not replaced). If you do not use the location operand, the system presents the display according to the MSGRT instructions in effect. If there are no MSGRT instructions in effect, the system presents the display in the issuing console's lowest unoccupied display area or, if all areas are full,in the area containing the oldest display. If no display area can be found (because the screen has no display areas or because all areas are being used by dynamic displays), the system presents the status display in the portion of the message area not assigned to a display area (area Z).

How to Request Status Displays

You can route status displays to specific display areas on the console screen by coding the L=a operand, or to the general message area by coding L=Z, with the command that you use to request the status display. Status displays cannot be routed to a full-capability console other than the one you are using unless you are at the master console.

To request a status display:

1. Type in the appropriate DISPLAY or TRACK commands, including the location (L=cca) operand if you wish to specify the display area in which the display is to be presented.
2. Perform an ENTER action.

Example: To request a display of active jobs in display area B of console 12 enter:

```
D   A,L=12B
```

How to Frame Status Displays

Depending on the size of the display and the number of lines available in the message area or display area, status displays can be divided into two or more frames. The control line (first line) of each display contains a frame number, and the last frame is marked FRAME LAST.

Note: You can only frame forward through a display; to re-examine previous frames of the display you must enter a request for a new display.

Framing by CONTROL Command: To move a status display forward to the next frame, use the the CONTROL D,F command:

```
{CONTROL}      D,F[,L={a  }]
{  K    }              {cc }
                       {cca}
```

D

specifies that information is to be displayed on the screen.

F
specifies that the next frame of a status display is to be displayed.

L = a, cc, cca
specifies the display area (a), the console (cc), or both (cca) where the specified action is to take place. If you omit this operand, the command affects the bottom-most display on the console where you enter the command.

Example: To display the next frame of a status display in display area B of console 12, enter the following:

```
K  D,F,L=12B
```

Framing by Cursor: The control line of a status display contains a framing indicator that appears as F on the screen. To frame by cursor:

1. Position the cursor at the F.
2. Perform an ENTER action.

Framing by Selector Pen or Light Pen: Position the selector pen or light pen over the F in the control line of the display.

How to Erase Status Displays

The method of deleting status displays depends on whether they are displayed in the message area of the screen (inline) or in a defined display area (out-of-line).

Erasing Inline Status Displays: Erase an inline status display by the message deletion methods used for other inline messages (that is, CONTROL E,SEG; CONTROL E,nn,nn; selector pen; or cursor). These methods are described under "How to Delete Messages" earlier in this chapter.

An inline status display that has not been completely displayed (that is, a display that is in progress) can be halted by performing the selector pen or cursor detect on the C (*C on a 2250) indicator in the control line or by using the CONTROL C,D command.

```
{CONTROL}     C,D,id,L=cc
{  K    }
```

C,D
specifies that an inline status display is to be cancelled.

id
is the three-digit identification number located in the control line of the display to be cancelled.

L = cc
specifies the console (cc) from which the display is to be removed.

Because of the speed of display consoles, this command has limited usefulness in controlling inline displays on the screen. It is used primarily to halt displays in progress on hardcopy consoles, such as the 3284 printer.

Example: To halt the status display with identification number 121, which is in progress in the general message area of console number 10, enter:

```
K  C,D,121,L=10Z
```

Erasing an Out-of-line Display: To erase an out-of-line status display (one that is presented in a display area), use the CONTROL E,D command:

```
{CONTROL}      E,D[,L={a  }]
{   K   }              {cc }
                       {cca}
```

E,D
specifies that a status display is to be erased.

L=a, cc, cca
specifies the display area (a), console (cc), or both (cca) where the specified action is to take place. If you omit this operand, the command affects the bottom-most display on the console through which you enter the command.

For example, to erase the displaying display area A of console number 12, enter the following:

```
K    E,D,L=12A
```

You can also erase out-of-line status displays by positioning the cursor at the E or *E indicator in the control line and performing an enter action, or by positioning the selector pen over the same indicator.

Note: It is not necessary to erase a status display from a display area in order to present another status display in that area. If a new status display is routed to an area that already contains a status display, the new display replaces the old display. However, it is good practice to erase status displays to avoid using buffer space that may be required for other messages.

Dynamic Status Displays

A dynamic status display is one that is requested once by the operator and then updated and redisplayed by the system each time a preset time interval elapses. The TRACK command is used to request a dynamic status display. (See Chapter 5 for a complete description of TRACK.)

```
{TRACK} {TS      }  [,L[IST]]  [,L= {a  }]
{  TR } {J[OBS]  }                  {cc }
        {A       }                  {cca}
```

A
specifies that a dynamic display of active jobs is to be presented.

LIST
requests a display of information about active jobs. When coded with TS, LIST requests information about active time-sharing jobs. With J, a list of mounts, started tasks, and initiated tasks is requested. With A, LIST requests information about all active jobs.

L=a, cc, cca
specifies the display area (a), console (cc), or console and display area (cca) where the display is to be presented. If you omit this operand, the display appears in a display area on the console through which you enter the command (unless routing instructions are in effect; see "MSGRT Command" earlier in this chapter).

The CONTROL D command has operands that are used to hold, frame, and update dynamic status displays:

```
{CONTROL}      D      {,H[,L= {a}]}
{  K    }             {      {cc}}
                       {      {cca}}
                       {,F[,L= {a}]}
                       {      {cc}}
                       {      {cca}}
                       {,U[,L= {a}]}
                       {      {cc}}
                       {      {cca}}
```

D
specifies that information is to be displayed on the screen.

H
specifies that updating of a dynamic status display is to be suspended.

F
specifies that the next frame of a status display is to be presented.

U
specifies that updating of a dynamic status display is to be resumed.

L=a, cc, cca
specifies the display area (a), console (cc), or console and display area (cca) where the specified action is to take place.

Dynamic status displays can only be presented in display areas. Also, once the dynamic display appears in an area, it has exclusive use of the area until it is terminated; it cannot be overlayed by another status display.

Example: To request a dynamic display of active jobs in display area A on console 12, enter the following command:

```
TR   A,L=12A
```

To suspend updating of a dynamic display in area A of console 12, enter the following:

```
K    D,H,L=12A
```

You can also place a dynamic display in hold mode by positioning the cursor at the H or *H indicator in the control line (first line) of the display and performing an ENTER action, or by positioning the selector or light pen over the H indicator.

To resume updating of the display in display area A of console 12 (assuming that you have previously requested hold mode for the display), enter the following command:

```
K    D,U,L=12A
```

You can also request that updating be resumed by positioning the cursor under the U or *U indicator in the control line of the display and performing an ENTER action, or by positioning the selector or light pen over the same indicator. The U indicator appears in the control line only when the display is in hold mode.

To frame a dynamic display in display area A of console 12, first place the display in hold mode by entering:

```
K    D,H,L=12A
```

Then display the next frame by entering:

```
K    D,F,L=12A
```

When you want to resume updating of the display, enter:

```
K    D,U,L=12A
```

Updating resumes immediately and the first frame of the updated display appears on the screen. If the time interval is changed by a K T command, the next update occurs after the new interval.

You can also perform the same hold-frame-update function by means of the cursor:

1. Place the display in hold mode by positioning the cursor at the H or *H indicator in the control line.
2. Perform an ENTER action.
3. Frame forward by positioning the cursor at the F or *F indicator (the F indicator only appears when the display is in hold mode).
4. Perform an ENTER action once for each frame that you want to see.
5. Resume updating of the display by positioning the cursor at the U or *U indicator in the control line (the U indicator only appears when the display is in hold mode).
6. Perform an ENTER action.

You can also perform the same hold-frame-update function with the selector pen or light pen:

1. Request hold mode by positioning the pen over the H or *H indicator.
2. Display the frames by positioning the pen over the F or *F indicator *once per frame.*
3. Resume updating by positioning the pen over the U or *U indicator.

How to Erase a Dynamic Display

A dynamic display initiated by a TRACK command is terminated by means of the STOPTR(PT) command:

```
{STOPTR}{TS      }   [,L=  {a  }]
{  PT  }{J[OBS]  }        {cc }
        {A       }        {cca}
```

A
specifies that a dynamic display initiated by a TRACK A,TS, or JOBS command is to be terminated and erased from the screen.

L=a, cc, cca
specifies the display area (a), console (cc), or console and display area (cca) where the specified action is to take place. If you omit this operand, the command affects the dynamic display in the bottom-most display area on the console through which you enter the command.

For example, to terminate and erase a dynamic display in display area A of console 12, enter the following:

```
PT   A,L=12A
```

You can terminate and erase a dynamic status display by positioning the cursor at the PT or *PT indicator in the control line of the display and performing an ENTER action (the PT indicator does not appear when the display is in hold mode). You can also terminate and erase the display by performing a selector pen or cursor detect on the same PT indicator.

STOPTR
can reduce the amount of information displayed. TR A and PT J have the effect of TR TS the next time the display is updated.

How to Change the Time Interval for Display Updating

To change the time interval for updating of dynamic displays, use the CONTROL T command.

```
{CONTROL}     T   {,UTME=nnn}   [,L=cc]
{ K      }        {[,REF]   }
```

T
indicates that the time interval for updating status displays is to be changed or displayed.

UTME
specifies the number of seconds in the time interval for updating status displays. The nnn value can be any number from 10 to 999.

REF
specifies that the current value of the time interval is to be displayed in the entry area in CONTROL command form. REF is the default.

There is a separate time interval for each console where a dynamic display is in progress. To set the time interval to update the display every two minutes (120 seconds), enter the following through the master console:

```
K   T,UTME=120
```

You can check the current time interval by entering:

```
K   T,REF or K T
```

In response, the system displays the current time interval in the entry area in CONTROL command form; for example:

```
K   T,UTME=120
```

You can then change the time interval by using the procedures described in this chapter under "How to Change Information in the Entry Area."

Note 1: If the issuing console is not a display console, the system responds to the CONTROL T,REF command with the following:

```
IEE922I K   T,UTME=nnn
```

The current time interval is indicated by the nnn figures. You can change the time interval by entering another CONTROL T,UTME command, and cc is the console specified in K T,L=cc.

Note 2: If there is a dynamic display in progress when you change the time interval, the new interval does not take effect until the interval in progress elapses. If a K D,U command is directed to the display area, the new interval takes effect immediately.

Example of the Use of Display Areas

The following example illustrates how to establish and use display areas on a display console.

```
A.  K A,4,6
B.  TR A
C.  D U
D.  K D,F
    K E,D
```

A. When the command, K A,4,6 is issued, two display areas are established:

- Area A, which consists of 4 lines
- Area B, which consists of 6 lines

B. By issuing the command TR A, you cause a TRACK display to appear in area A, providing a constant indication of system activity.

C. If you then issue any status display command, for example D U, the display appears in area B.

D. If a static display is present on the screen, the K, D,F command frames it and the K E,D can be used to erase it. If no static display is present on the screen, when the K D, F is issued, the K D,F frames the TRACK (dynamic) display. In this case, the K E, D is rejected.

Note: When issued, the K D,F and K E,D commands locate the static display on the screen and handle it in preference to the TRACK display. Area B, then, functions as if it were the only display area. The routing location operand (L=cca) is not required.

Console Mode

The 2260, 3277 model 2 and similar display consoles can operate in two different ways:

- *Full-capability Mode:* A full-capability console has both input and output capability; the console can be used to enter commands and to display messages. One full-capability console is the master console; there can be many full-capability consoles in the system.
- *Output-only Mode:* An output-only display console can be used to display messages, but it cannot be used to enter commands. There are two ways that an output-only console can operate: *status display* mode, which applies to output-only display consoles designated for the presentation of status displays: and *message stream* mode, which applies to output-only display consoles designated for the presentation of messages other than status displays.

Varying Console Mode

You can vary a console's mode of operation: (1) from full-capability mode to output-only mode for presentation of either general messages *(message stream* mode) or status displays *(status display* mode): (2) from one output-only mode to the other (from status display to message stream, or vice versa); or (3) from output-only mode to full-capability mode.

System Requirements: A console qualified for mode change cannot be changed to output-only mode while it is the master console in the system; you must first designate another console to take over the master console, see the description of the VARY MSTCONS command in Chapter 5.

Screen Format: When you vary a full-capability console to output-only mode, the PFK display line, the instruction line, and the entry area are incorporated into the message area as shown in Figure 3.4. Once a display console has been placed in output-only mode, its input capability is nullified. You must use another console to make any requests concerning an output-only console.

The Command Form: Use the CONTROL V command to vary console mode:

```
{CONTROL}      V,USE= {SD}  ,[L=cc]
{  K   }               {MS}
                       {FC}
```

V

specifies that the operating mode of a display console is to be changed.

USE

specifies the new mode for the console.

SD

specifies that the console is to be changed to output-only mode for the presentation of status displays (see "Status Display Mode" in this chapter).

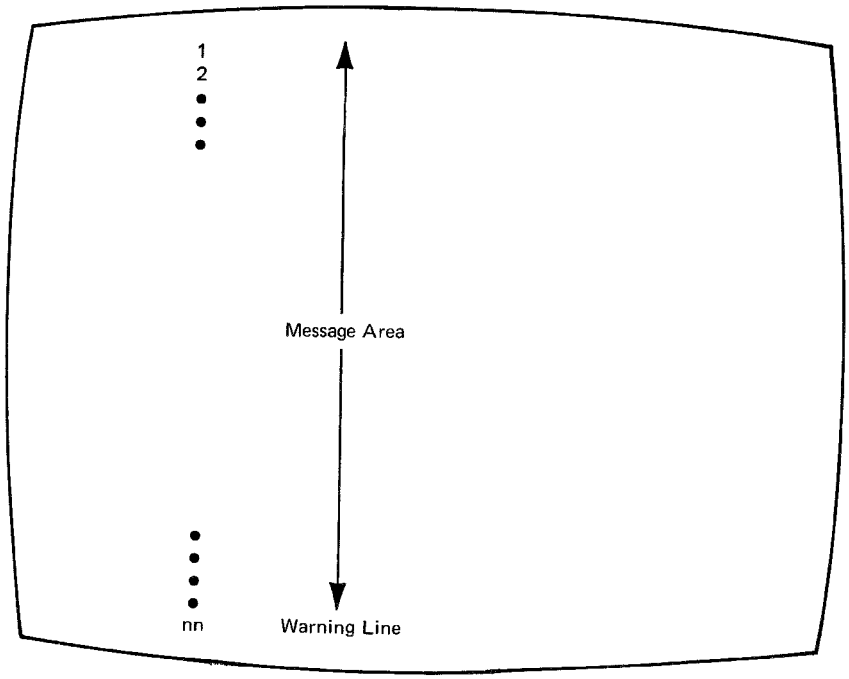


Figure 3.4. Format of the 3277 Model 2 Screen in Output-Only Mode

MS

specifies that the console is to be changed to output-only mode for the presentation of messages other than status displays (see "Message Stream Mode" in this chapter).

FC

specifies that the console is to be changed to full-capability (input/output) mode of operation.

L=cc

specifies the identifier (cc) of the console whose mode is to be changed. If you omit this operand, the command affects the console through which you enter the command (unless routing instructions are in effect; see "MSGRT Command" earlier in this chapter).

Note: When SD or MS mode is specified, the console is considered an output-only console. If you want to change any console characteristics using the VARY command, you must use the O-unit parameter of the VARY command (see "Assigning and Controlling MCS Consoles" under the VARY command in Chapter 5). Otherwise, the VARY command is rejected.

Status Display Consoles

A status display console provides a convenient area for displaying system status information and frees screen space on the master console for use by other system messages.

To change console 10 from full-capability mode to status display mode, enter:

```
K    V,USE=SD,L=10
```

In response to this request, the console's message area expands to 23 (11 for a 2260) lines, any information that had appeared on the screen disappears, and the display area specifications that were made during system generation go into effect.

When moving from message stream mode to status display mode, the message area remains the same size (23 lines, or 11 lines for a 2260), and the display area specifications made during system generation go into effect.

Controlling Displays on Status Display Consoles: Procedures for requesting, framing, holding, and terminating status displays (static and dynamic) are described in this chapter under "System Status Displays." Since the output-only console has no input capability, you must enter each request concerning the console on a separate, full-capability console. Use the routing location operand (L=cca) with each command to designate the console and display area at which the action is to take place, or establish routing defaults by means of the MSGRT command.

The routing location operand can be entered only from a console with a command group entry authorization of CONS (console control). Command group entry authorizations is described in Chapter 2.

Message Stream Consoles

In message stream mode, the console provides an area for presentation of operator's messages away from the master console. The type of messages sent to a message stream console depends on the routing codes assigned to that console. Message stream consoles can provide system monitoring in tape or disk libraries, or assist in system security.

To change console 10 from full-capability mode to message stream mode, enter:

```
K    V,USE=MS,L=10
```

In response to this request, the message area expands to 23 lines, (11 lines for a 2260), and any information on the screen disappears. All display area specifications are removed. When changing from status display mode, the message area remains the same size but the display area specifications are removed.

Routing Messages to Message Stream Consoles: The routing codes assigned to each console determine the messages routed to that console. Routing codes are assigned to consoles during system generation, but you can change the assignments with the VARY command. Check the routing codes assigned to each console with the DISPLAY CONSOLES command. (VARY and DISPLAY CONSOLES are described in Chapter 5. Routing codes are described in the *OS/VS2 Message Library: Routing and Descriptor Codes.*

Deleting Messages from Message Stream Consoles: When a console is placed in message stream mode, roll-deletable message deletion goes into effect automatically (see "Automatic Message Deletion" and "How to Delete Messages" earlier in this chapter). Messages are removed from the screen without any intervention on your part. ***Note:*** In unusual circumstances, the screen could become full of action messages for which no action can be taken. If this occurs, it may be necessary for you to vary the console out of operator console mode (with the VARY command, described in Chapter 5) in order to remove the unwanted action messages.

Returning to Full-Capability Mode

To change console 10 to full-capability mode, enter:

```
K    V,USE=FC,L=10
```

In response to this request, the message area of console 10 returns to 19 (8 lines for a 2260) lines, line 20 becomes the PFK display line, (there is no PFK line on a 2260), line 21 becomes the instruction line, and lines 22 and 23 (lines 10 and 11 on a 2260) become the entry area. The following message appears in the instruction line:

```
IEE152I ENTER    CANCEL  D    C,K
```

The display area specifications return to the specifications made during system generation, and you can check these specifications for console 10 by entering:

```
K    A,REF,L=10
```

The specifications appear in the entry area, and you can change the specifications using the procedures described in this chapter under "How to Change Information in the Entry Area."

Error Conditions

Several types of errors can occur that directly affect the operation of display consoles. In some cases, the error is made apparent by a sudden screen failure, the appearance of an error message, or the locking of the keyboard. In other cases, the error might not be immediately apparent. Errors can be caused either by a programming problem (system error) or a console malfunction (hardware error).

System Errors

When certain types of system errors occur, the screen is blanked, and an error message appears in the center of the screen. Other types of system error conditions are characterized by an abnormal lack of console activity.

Blank Screen and Error Message

If the error message indicates that a recoverable system error has occurred, perform the action specified by the error message, and then perform a CANCEL action. This should restore the screen. It is good practice to review the messages at this time to make certain that no messages were lost during error recovery.

If the error message indicates that an unrecoverable system error has occurred, the system must be loaded again. Follow normal procedures for initial program load (IPL), and notify the programmer responsible for the system.

Console Inactivity

Console inactivity is characterized by a lack of messages or system response to commands. It could be due to the level of system activity, or the result of a problem in the message handling portion of the control program.

One function of the message handling portion of the control program is check for the end line of a status display or other multiple-line message. If for some reason the system fails to detect the end line of either a status display or a multiple-line message to the operator, your console could be put into a condition of inactivity awaiting completion of the display or message. This situation occurs because the system is designed to present all of the lines of a status display or multiple-line message, once it is begun, before presenting any other message on the console.

If your console seems to be abnormally inactive, check the system response by requesting a display of the time:

```
D    T
```

The system should respond immediately (within a few seconds) with the time and date. If it does not respond, cancel any status displays being presented on the inactive console using the procedure for erasing a status display. If this does not return the console to normal activity, cancel any jobs that have written multiple-line messages to the console.

If neither of these procedures returns the console to normal activity, assume that there is some other problem related to the console. Check for a console hardware error. If possible, switch control to another console. If the system must be loaded again, follow normal procedures for initial program load (IPL). Report the occurrence of this problem to the programmer responsible for the system.

Console Hardware Errors

A console hardware error may be signified by one or more of the following conditions:

- An error message is centered on the screen (the remainder of the screen is blank).
- The screen is blank (and no error message appears).
- The screen appears normal, but the keyboard is locked and command entry is not possible.

Error Message Response

If a console hardware error occurs, one of the following messages can appear centered on the screen:

```
IEE170E RETRYABLE ERROR.  RECENT ACTION MAY NEED TO BE REPEATED.
IEE170E PRESS THE CANCEL KEY TO RESTORE THE SCREEN.
-- or --
IEE171E CONDITIONAL ERROR.  RECENT ACTION MAY NEED TO BE REPEATED.
IEE171E PRESS CANCEL TO CONTINUE OR SWITCH CONSOLES.
```

Perform the indicated action (perform a CANCEL action). This should restore restore the screen, including messages displayed in-line in the message area, the instruction line, and the warning line. The entry area and the PFK line are blanked, any out-of-line displays are erased, and the cursor is positioned to the first data entry position. Message numbering is terminated (if it was previously in effect).

Note 1: If you do not perform a CANCEL action, the system rewrites the screen (same effect as CANCEL) after about 30 seconds have elapsed. If a console hardware error results from keyboard input when the CANCEL action is performed, the system regards it as a permanent I/O error and performs an automatic console switch. All messages (except status displays) are moved to the new console. (Console switch is described in Chapter 2.)

Note 2: For some console-detected errors, the 3036 console re-IMPL's. This can result in message IEE171E. An operator-initiated IMPL can also result in message IEE171E. In either case, follow the procedure for the CLEAR key under the section "Special Keys" in Appendix E.

Blank Screen Response

If the console screen goes blank, a console switch is probably taking place. The following message should appear on the new console:

```
IEE143I  OLD=xxx, NEW=xxx, VALDCMD=xx
IEE143I  ROUTCDE=xx[,xx]  T=x H=x
```

In the actual message, the appropriate values appear in place of the x's. Use the alternate console to continue operating the system, and have the old console checked for the source of the error.

Note: It is normal for the screen to go blank for a few seconds if the back-tab key is pressed when the cursor is not in the entry area.

Locked Keyboard Response

Sometimes the system is unable to blank the screen. If you find that you cannot enter commands through a console that appears normal, try to restore the screen by performing a CANCEL action.

If a console switch has taken place, operate the system from the alternate console, and have the old console examined for the source of the error.

Note: Inhibited input, with or without keyboard locking, can also occur when the system goes into an ABEND wait state or when a problem occurs in the message handling portion of the control program. Check the procedures described for console inactivity under "System Errors."

Chapter 4. The Shared DASD Option

The shared direct access storage device (DASD) option allows multiple processors to access common data residing on direct access storage devices. This is accomplished through a hardware feature of the DASD control unit and the reserve/release function of the operating system. The reserve function allocates a specified device for use only by the processor issuing the 'RESERVE'. The release function removes a specified device from reserved status, allowing the device to be reserved by another processor.

The shared direct access storage device (DASD) option is included in the operating system during system generation. It allows up to four processors to share a pool of 2319/2314 DASD devices or up to eight processors to share a pool of 3330/3333, 3330V, 3340/3344, or 3350 direct access storage devices. Two processors can share a pool of 2305 direct access storage devices. Up to two paths from a processor to a control unit are allowed.

The advantages of the shared DASD option include:

- Reducing the amount of time you have to spend moving volumes from one system to another.
- Minimizing the updating of data sets. You have to update only one instead of two or more duplicates.
- Simplifying scheduling. Unless the job has other special requirements, you can run a job needing a specific data set on a shared device on any of the sharing systems.

A device is reserved when it is allocated to a particular system: a device is released when it is no longer allocated to that particular system.

A multichannel hardware switch allows the control program to control the reserving and releasing of the shared device and data. The multichannel switch allows a common control unit to be switched on a first-come, first-served basis among several channels, each possibly from a different system. On systems with 2314 or 3330 devices, be sure any hardware toggle switches marked "tagged" and "untagged" are always set to the "tagged" position.

If the control unit or device is busy with one of the other sharing systems, an attempt to initialize the system from a shared device fails. Continue to retry the initialization until it is successful; the retry has no adverse effects on the other systems. Likewise, in a uniprocessing environment, system reset only resets the functions of a shared control unit or device that belongs to the system that issues the system reset. Any function related to another system remains undisturbed. However, in a tightly coupled multiprocessing configuration, hitting SYSTEM RESET resets both the processors and the attached I/O. A selective or malfunction reset has no effect on device reservations or status.

If, when you initialize a system having shared DASD, a second system has reserved the shared device, the initializing system will wait a half a second and then issue this message:

```
* id IEA120A DEVICE ddd SHARED. REPLY 'CONT' or 'WAIT'
```

The recommended reply is 'WAIT'. 'WAIT' causes the system to wait until the device is released by the processor that is presently using it. If your system waits more than one minute, re-IPL. If the device is still reserved on the other system, message IEA120A is issued again. Reply 'CONT' and the device is marked offline to the processor performing the IPL. If you reply 'CONT', you might be unable to establish the proper combination of mount characteristics (listed later in this chapter). Therefore, it is recommended that you run *no* jobs on the IPL processor that might require the unavailable device, and notify your system programmer of the problem. These requirements do not apply to the 3330V. See *Operator's Library: IBM 3850 Mass Storage Systems (MSS) Under OS/VS*, for further information about the 3330V.

Note: Care must be taken to reply WAIT for 3344-emulated 3340 devices and 3350-emulated 3330 devices which are to be marked permanently resident by the volume attribute list (VATLST) facility. Continuing with an IPL, without marking emulated devices as permanently resident, can cause unpredictable results.

Shared DASD Guidelines

When operating a system with shared DASD be aware of the following considerations:

- Whenever you change the mount characteristics of a shared volume, the status of a device, or both, you must maintain a valid combination of mount characteristics across all sharing systems. Remember that the VARY OFFLINE command is always the first step of the mount procedure and that the procedure must be done in parallel on all sharing systems.
- If there is a hardware malfunction on other than the system residence device, you must vary the device offline on all sharing systems. You can then mount the shared volume on another shared device, if one is available, as long as you follow parallel mount procedures on all sharing systems.
- In a multiprocessing environment, shared DASD support during the I/O restart function of ACR (alternate CPU recovery) allows the system to re-reserve a device through the alternate path. If channel reconfiguration hardware (CRH) is present (3168), the re-reserve is not necessary since the path is not lost.
- For DASD that is attached only to both processors of a tightly coupled multiprocessing system, a special sysgen option (SHAREDUP) is provided. This option allows serialization of the device without reserve/release overhead. That is, sharing systems access the device sequentially. One system must finish before the next can access the device. When the system is reconfigured, either automatically by ACR or an operator-initiated VARY command, these devices are marked shared, and reserve/release serialization is employed.
- A path to a device cannot be removed by a VARY command if there is a reserved device on that path. If this occurs, message IEE379I or IEEE719I is issued.

Specifying Shared DASD Mount Characteristics

Volume characteristics, device status, and volume mounting and demounting are all affected by the shared DASD option. One of the following combinations of mount characteristics and device status must be in effect for each device that is physically shared between processors.

System A	Other Systems
Permanently resident	Permanently resident
Reserved	Reserved
Removable	Offline — Non-JES3 devices
Removable	Removable — JES3-managed devices
Offline	Permanently resident, reserved, or removable (In JES2, if a device is removable in one system, it must be offline in all others)

After initial program loading, a valid set of mount characteristics must be established before the system begins device allocation. This can be accomplished by the system programmer specifying the mount characteristics of shared devices in the VATLST member of SYS1.PARMLIB. The mount characteristics of a volume, the status of a device, or or both, can be changed in one system as long as the resulting combination across all systems is valid.

Volumes that are not permanently resident must be put in reserved status by using the MOUNT command before jobs needing the shared volume are started. To change volumes that are reserved, you must:

- Use the VARY command to put the device to offline status to each sharing system and wait for the offline message in each system. The device does not go offline until the message is issued. If no jobs are in progress, the offline message does not appear on the console. Issue a "S DEALLOC" command to receive the message that the device is offline.
- Use the MOUNT command to notify each sharing system of the units where the new volume is being placed, and to put that volume in reserved status.
- Use the MOUNT command to mount the volume.

Jobs requiring this new volume should not be selected by an initiator before the volume is mounted. Job selection can be held by one of the following:

1. Using the TYPRUN=HOLD parameter on the job card.
2. Using the appropriate subsystem command.
3. Assigning the job to a job class and not activating that class for subsystem scheduling.

After the volume is mounted, use the job entry subsystem command to let the job be selected in cases 1 and 2; activate the class for subsystem scheduling in case 3.

When you want a shared non-JES3 device to be allocated by only one system, the operator of each system sharing the device should use the VARY command to place the device offline to their systems.

Sharing Data Sets

Any of your installation's own data sets can be shared. In addition, the following system data sets can be shared.

- Private catalogs.
- Volume tables of contents (VTOC) on all shared volumes.
- JES2 and JES3 spool allocations.

The following data sets can be shared as long as concurrent updating is not attempted:

- SYS1.LINKLIB if all sharing systems have the same modules with common name configurations.
- SYS1.PROCLIB when not on the IPL volume.
- SYS1.LPALIB if all sharing systems have the same modules with common name configurations.
- SYS1.IMAGELIB.

The following system data sets cannot be shared:

- Master catalog
- PASSWORD
- SYS1.BRODCAST
- SYS1.DCMLIB
- SYS1.DUMP
- SYS1.LOGREC
- SYS1.MANX
- SYS1.MANY
- SYS1.NUCLEUS
- SYS1.STGINDEX
- SYS1.SVCLIB
- SYS1.UADS
- Page Data Sets

Chapter 5: System Commands

This chapter contains a description of each system command, its parameters, and the functions that it performs.

The commands are described in alphabetical order and follow a standard command format and syntax, which are described at the beginning of this chapter.

The introduction to each command contains a general description of the command and its functions, followed by the complete syntax of the command. This introduction is followed by a detailed description of the command's functions, including the correct syntax, and parameters needed to perform that function.

Appendix A lists each command, its functions, and the page number within this chapter where the function is located. Also identified are the operator command groups to which each command belongs, and whether the command can be entered from the job stream or a time sharing terminal.

When entering system commands, follow the syntax rules illustrated in Figure 5.1 unless specifically indicated otherwise.

Notation	Description	Example	
		Book Syntax	Sample Entry
Apos-trophes	Must be entered as shown.	SEND 'message', NOW	SEND 'listings ready', NOW
Brackets	Enter one of the enclosed parameters. Do not enter the brackets.	D M [=CPU / =DEV / =n]	D M=DEV
Comma	Must be entered as shown.	DISPLAY C,K	DISPLAY C,K
Ellipsis (...)	The parameter can be repeated. Do not enter the ellipsis.	VARY (unitaddr[unitadr] ...), ONLINE	VARY (282,283,287), ONLINE
Lower Case	A parameter must be substituted. You can enter the parameter in either upper or lower case.	MOUNT unitaddr	MOUNT A30 or mount a30
Parentheses and special characters	Must be entered as shown.	DUMP COMM=(text)	DUMP COMM=(PAYROLL)
Stacked items in braces	*One* of the alter-natives must be entered.	{MONITOR / MN} {DSNAME / SPACE / STATUS}	MN SPACE
Under-score	The underscored parameter is the system default. If you do not enter any of the parameters, the system supplies the underscored parameter.	MR {D=(operand[,operand] ...) / TR=A / K} <u>REF</u> / NONE	MR
Upper Case	The parameter must be spelled as shown. You can enter the parameter in either upper or lower case.	DISPLAY T	DISPLAY T or display t

Figure 5.1. System Command Syntax

Command Format

Operator commands can be entered in uppercase or lowercase. Unless enclosed in apostrophes, lowercase letters are converted to uppercase. Therefore, when a lowercase response is required, you must enclose the text in apostrophes. Also, when an apostrophe appears in the text of a command and the text is enclosed in apostrophes, you must enter two apostrophes in the text. For example, you would enter:

```
SEND    'Your job''s done'
```

In a JES2 environment, system commands can be entered either through the input stream (card reader) or through the Multiple Console Support (MCS) console. Examples in this publication show the format for console entry. In a JES3 environment, system commands can be entered through the input stream, through an MCS console, or through a JES3 console. The Appendix in this publication indicates which system commands can be entered through the input stream. Examples in this publication show the format for MCS console entry (refer to *Operator's Library: OS/VS2 MVS JES3 Commands* for details on entering system commands through a JES3 console).

Notes:

1. If you enter a system command through a card reader in a JES2 environment and the command is placed between jobs, enter $VS, 'system command'; if the command is placed within a job, enter //b. In a JES3 environment, enter //**T, sysname; syscommand.
2. Do not use the JES backspace character within a system command.

The system command format is as follows:

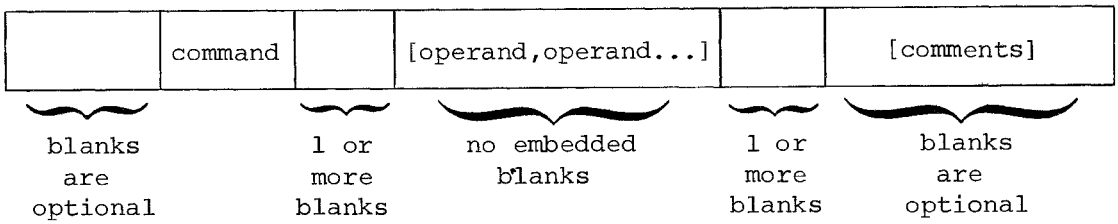


Figure 5.2. System Command Format

Note: If no operands are entered, a blank must follow the command and a comma and a blank must precede the comment.

You can enter one command per line with a maximum of:

- 126 characters from a console.
- 80 characters through a card reader.

Routing Command Responses

There are two ways to route command responses:

- The L=cca operand of any system command that produces a display
- The MSGRT command

The L=cca operand routes status displays to a specific display area, console, or both. If you do not use the L=cca operand, the display is presented in the first available area of the console through which you enter the command.

The MSGRT command sets up routing defaults for displays from the DISPLAY, TRACK, STOPTR, and CONTROL commands. The defaults remain in effect until another MSGRT command that specifies the same command or operand is issued.

CANCEL Command

Use the CANCEL command to immediately terminate:

- A job in execution
- A time sharing user
- A MOUNT command
- An external writer allocation
- The output processing for a job

Note: If the function fails to cancel after serveral attempts, consider using the FORCE command.

The complete syntax for the CANCEL command is:

```
{CANCEL}    {jobname      [,DUMP]}
{   C  }    {U=userid             }
            {unitaddr             }
            {devicetype           }
            {identifier           }
```

jobname
The name of the job to be canceled.

Note: If a job is executing, you can use either the system CANCEL command or the appropriate subsystem command. However, if the job is not executing, you must use the subsystem command to cancel the job.

DUMP
A dump is to be taken.

U = userid
The user identifier of the time sharing user you want to cancel. To display the user identifiers of the active TSO terminal users, enter DISPLAY TS,LIST.

unitaddr
The unit address specified when the MOUNT command was issued.

devicetype
The type of device specified when the MOUNT command was issued.

Note: A MOUNT command for a tape unit can end without the volume being mounted. If the MOUNT command has ended and the mount request was not satisfied, issue the UNLOAD command to free the tape unit.

identifier
The identifier or the devicename specified on the START command for the external writer. Entering this command during device allocation causes the writer to terminate; entering this command when the writer is processing causes the output processing for a job to terminate, but the external writer continues to process other data sets. You can cancel an output writer *only* during device allocation.
In place of the identifier or device name specified on the START command for the writer, you can also use the device type (for example, 2400) associated with the unit address specified in the START command or the device type associated with the cataloged procedure started by the START command.

Example 1: Cancel the job named example and take a dump.

```
c example,dump
```

Example 2: Cancel the job named example. You might or might not get a dump depending on the system routine in control at the time of cancel.

Example 3: Log user A237 off the system.

```
c u=a237
```

Example 4: Log user A237 off the system and take a dump.

```
c u=a237,dump
```

Example 5: Cancel the request for a volume to be mounted on unit 232.

```
c 232
```

Example 6: Cancel the request for a volume to be mounted on a 2314 device.

```
c 2314
```

Example 7: Discontinue the device allocation for writer 00E.

```
c 00e
```

Example 8: Discontinue the output processing currently being done on 00E and select another output data set for processing.

```
c 00e
```

CHNGDUMP Command

The CHNGDUMP command dynamically alters the mode and system dump options list for any dump type. The DUMP types are SYSABEND, SYSMDUMP, SYSUDUMP, and SDUMP. If you issue multiple CHNGDUMP commands, the effects on the system dump options are cumulative.

Dump Options and Modes

MVS uses the dump mode and dump option information each time the system or a user requests a dump. The mode determines how MVS processes the options specified on the dump request and the options contained in the system dump options list. The dump options identify, for each dump type, the data areas to dump.

Each time you IPL the system, MVS establishes system dump option lists. The lists specify the dump mode and dump options currently in effect for each dump type. For SYSABEND, SYSMDUMP, and SYSUDUMP, the options set during IPL are obtained from SYS1.PARMLIB members IEAABD00, IEADMR00, and IEADMP00, respectively. Because SDUMP has no corresponding SYS1.PARMLIB member, it starts with an empty dump list. Initially the mode for all four dump types is set to ADD.

Dump Modes

In addition to ADD, other dump modes are OVER and NODUMP. The meaning of each mode is:

- ADD - When a dump is requested for a dump type that is in ADD mode, the system merges the options specified on the dump request with the options specified in the system dump options list. The result of the merge determines the data areas to dump. If there is a conflict between an option specified on the dump request and an option specified in the options list, the options list has priority.
- OVER - When a dump is requested for a dump type that is in OVER mode, the system ignores the options specified on the dump request and uses only the options specified in the system dump options list.
- NODUMP - When a dump is requested for a dump type that is in NODUMP mode, the system ignores the request and does not take a dump.

Changing the Dump Mode and Options

Before changing the dump mode or options you can determine the current mode and options by issuing DISPLAY DUMP, OPTIONS.

Changing the mode of a dump type can also affect its system dump options list as follows:

- If a dump type is changed from OVER mode to ADD mode, its SYS1.PARMLIB options are added to its system dump options list.
- If a dump type is changed from ADD mode to OVER mode, its SYS1.PARMLIB options are deleted from its system dump options list.
- If a dump type is set to NODUMP mode, all options are deleted from its system dump options list.

If you make an error specifying the CHNGDUMP command, the system rejects the entire command and issues an error message.

The complete syntax of the command is:

```
{CHNGDUMP}
{CD      }
   SET   {{,NODUMP}
          {,ADD   }
          {,OVER  }
          ,SDUMP    [{,NODUMP}
                     {[=(option[,option]...)]} [,Q={YES}] [{,OVER}]]
                                                        {NO }  {,ADD}
          {,SYSABEND}
          {,SYSUDUMP}  [{,NODUMP}
                        {[,SDATA=(option[,option]...)]} [PDATA=(option[,option]...)] [{,OVER}]]
                                                                                      {,ADD}
          ,SYSMDUMP    [{,NODUMP}
                        {[=(option[,option]...)]} [{,OVER}]]
                                                   {,ADD}
         }
         [{,SDUMP     [{[=(option[,option]...)]} [,Q={YES}]]
           {,ALL}                                        {NO }
   DEL    {,SYSABEND}
          {,SYSUDUMP}  [{[,SDATA=(option[,option]...)] [,PDATA=(option[,option]...)]}]
           {,ALL}
          ,SYSMDUMP    [{=(option[,option]...)}]
           ,ALL
         }]
   RESET [,SDUMP
          ,SYSABEND
          ,SYSMDUMP
          ,SYSUDUMP
          ,ALL      ]
```

Setting the Dump Mode and Options

Use the following form of the CHNGDUMP command to set the dump mode and turn on options in the dump options list.

```
{CHNGDUMP}
{CD      }
   SET   {{,NODUMP}
          {,ADD   }
          {,OVER  }
          ,SDUMP    [{,NODUMP}
                     {[=(option[,option]...)]} [,Q={YES}] [{,OVER}]]
                                                        {NO }  {,ADD}
          {,SYSABEND}
          {,SYSUDUMP}  [{,NODUMP}
                        {[,SDATA=(option[,option]...)]} [PDATA=(option[,option]...)] [{,OVER}]]
                                                                                      {,ADD}
          ,SYSMDUMP    [{,NODUMP}
                        {[=(option[,option]...)]} [{,OVER}]]
                                                   {,ADD}
         }
```

SET
Set the dump mode and turn on specified options in the system dump options list.

SET,NODUMP
Set the SDUMP, SYSABEND, SYSMDUMP, and SYSUDUMP dump modes to NODUMP, and delete all options from their system dump options list.

SET,ADD
Set the SDUMP, SYSABEND, SYSMDUMP, and SYSUDUMP dump modes to ADD. If a dump type was previously in OVER mode, add its SYS1.PARMLIB options to its system dump options list.

SET,OVER
Set the SDUMP, SYSABEND, SYSMDUMP, and SYSUDUMP dump modes to OVER. If a dump type was previously in ADD mode, delete its SYS1.PARMLIB options from its system dump options list.

SET,SDUMP
Set the SDUMP dump mode to ADD.

SET,SDUMP,NODUMP
Set the SDUMP dump mode to NODUMP

SET,SDUMP=(options)
Turn on the specified options in the SDUMP system dump options list.

SET,SDUMP,Q=YES or NO
Specifies whether or not SDUMP is to quiesce the system while dumping SQA and CSA.

SET,SDUMP,OVER or ADD
Set the SDUMP dump mode to the specified mode.

SET,SYSABEND
Set the SYSABEND dump mode to ADD.

SET,SYSABEND,NODUMP
Set the SYSABEND dump mode to NODUMP.

SET,SYSABEND,SDATA=(options)
Turn on the specified SDATA options in the SYSABEND system dump options list.

SET,SYSABEND,PDATA=(options)
Turn on the specified PDATA options in the SYSABEND system dump options list.

SET,SYSABEND,OVER or ADD
Set the SYSABEND dump mode as specified.

SET,SYSUDUMP
Set the SYSUDUMP dump mode to ADD.

SET,SYSUDUMP,NODUMP
Set the SYSUDUMP dump mode to NODUMP.

SET,SYSUDUMP,SDATA=(options)
Turn on the specified options in the SYSUDUMP system dump options list.

SET,SYSUDUMP,PDATA=(options)
Turn on the specified PDATA options in the SYSUDUMP system dump options list.

SET,SYSUDUMP,OVER or ADD
Set the SYSUDUMP dump mode as specified.

SET,SYSMDUMP
Set the SYSMDUMP dump mode to ADD.

SET,SYSMDUMP,NODUMP
Set the SYSMDUMP dump mode to NODUMP.

SET,SYSMDUMP=(options)
Turn on the specified options in the SYSMDUMP system dump options list.

SET,SYSMDUMP,OVER or ADD
Set the SYSMDUMP dump mode as specified.

Resetting the System Dump Options List to Initial Specifications

Use the following form of the CHNGDUMP command to reset all dump option lists to values specified at system generation.

```
{CHNGDUMP}  DEL [ ( ,SDUMP        [{[=(option[,option]...)] [,Q= {YES} ]}]
{   CD   }        ( ,ALL                                       {NO }  )
                  {,SYSABEND}     [{[,SDATA=(option,[,option]...)] [,PDATA=(option[,option]...)]}]
                  {,SYSUDUMP}     [{                ,ALL                                          }]
                  ,SYSMDUMP       [{ =(option[,option]...)}]
                                   {,ALL                  }
                  ,ALL                                                                            ) ]
```

DEL
Reset all system dump options lists to the values that were established during system initialization.

DEL,SDUMP
Set the dump mode for each dump type to ADD, and reset the system dump options list for each type to the values established during system IPL. (SYSABEND, SYSMDUMP, and SYSUDUMP options are set from values found in SYS1.PARMLIB members IEAABD00, IEADMR00, and IEADMP00 respectively. All SDUMP options are turned off.)

DEL,SDUMP=(options)
Turn off the specified options in the SDUMP system dump options list. If the mode is OVER and all SDUMP options have been turned off, RESET the SDUMP system dump options list.

DEL,SDUMP,Q=YES or NO
Specifies whether or not SDUMP is to quiesce the system while dumping SQA and CSA.

DEL,SDUMP,ALL
Set the dump mode for each dump type to ADD, and reset the system dump options list for each type to the values established during system IPL. (SYSABEND, SYSMDUMP, and SYSUDUMP options are set from values found in SYS1.PARMLIB members IEAABD00, IEADMR00, and IEADMP00 respectively. All SDUMP options are turned off.)

DEL,SYSABEND
Set the dump mode for each dump type to ADD, and reset the system dump options list for each type to the values established during system IPL. (SYSABEND, SYSMDUMP, and SYSUDUMP options are set from values found in SYS1.PARMLIB members IEAABD00, IADMR00, and IEADMP00 respectively. All SDUMP options are turned off.)

DEL,SYSABEND,SDATA=(options)
Turn off any specified SDATA options that were turned on in the SYSABEND system dump options list by previous CHNGDUMP commands. SYS1.PARMLIB options remain unchanged. If the mode is OVER and all SDATA and PDATA options have been turned off, RESET the SYSABEND system dump options list.

DEL,SYSABEND,PDATA=(options)
Turn off any specified PDATA options that were turned on in the SYSABEND system dump options list by previous CHNGDUMP commands. SYS1.PARMLIB options remain unchanged. If the mode is OVER and all SDATA and PDATA options have been turned off, RESET the SYSABEND system dump options list.

DEL,SYSABEND,ALL
Set the dump mode for each dump type to ADD, and reset the system dump options list for each type to the values established during system IPL. (SYSABEND, SYSMDUMP, and SYSUDUMP options are set from values found in SYS1.PARMLIB members IEAABD00, IEADMR00, and IEADMP00 respectively. All SDUMP options are turned off.)

DEL,SYSUDUMP
Set the dump mode for each dump type to ADD, and reset the system dump options list for each type to the values established during system IPL. (SYSABEND, SYSMDUMP, and SYSUDUMP options are set from values found in SYS1.PARMLIB members IEAABD00, IEADMR00, and IEADMP00 respectively. All SDUMP options are turned off.)

DEL,SYSUDUMP,SDATA=(options)
Turn off any specified SDATA options that were turned on in the SYSUDUMP system dump options list by previous CHNGDUMP commands. SYS1.PARMLIB options remain unchanged. If the mode is OVER and all SDATA and PDATA options have been turned off, RESET the SYSUDUMP system dump options list.

DEL,SYSUDUMP,PDATA=(options)
Turn off any specified PDATA options that were turned on in the SYSUDUMP system dump options list by previous CHNGDUMP commands. SYS1.PARMLIB options remain unchanged. If the mode is OVER and all SDATA and PDATA options have been turned off, RESET the SYSUDUMP system dump options list.

DEL,SYSUDUMP,ALL
Same as RESET,SYSUDUMP.

DEL,SYSMDUMP
Same as RESET,SYSMDUMP.

DEL,SYSMDUMP=(options)
Turn off any specified options that were turned on in the SYSMDUMP system dump options list by previous CHNGDUMP commands. If the mode is OVER and all SYSMDUMP options have been turned off, RESET the SYSMDUMP system dump options list.

DEL,SYSMDUMP,ALL
Same as RESET,SYSUDUMP.

The options that can be specified on the CHNGDUMP command are as follows:

SDUMP options:

ALLPSA or {NOALLPSA / NOALL} — prefix storage area for all CPUs. NOALLPSA or NOALL specifies that these areas are not to be dumped.
CSA — common storage area.
LPA — link pack area.
LSQA — local system queue area.
NUC — resident nucleus.
PSA — prefix storage area of the dumping CPU.
RGN — entire private area.
SQA or NOSQA — system queue area. NOSQA specifies that this area not be dumped.
{SUMDUMP / SUM} or {NOSUMDUMP / NOSUM} — SUM or SUMDUMP requests the SUMMARY DUMP function. See *OS/VS2 System Programming Library: Supervisor*, for a description of the SUMMARY DUMP function.
NOSUM or NOSUMDUMP requests that the function not be performed.
SWA — scheduler work area.
TRT — GTF or supervisor trace data.

SDATA options (for SYSABEND and SYSUDUMP types):

ALLSDATA — sets all of the other SDATA options.
CB — task related control blocks.
DM — data management control blocks.
ENQ — enqueue control blocks.
ERR — RTM control blocks.
IO — I/O supervisor control blocks.
LSQA — local system queue area.
NUC — resident nucleus.
SQA — system queue area.
SWA — scheduler work area.
TRT — GTF or supervisor trace data.

PDATA options (for SYSABEND and SYSUDUMP types):

ALLPDATA — sets all of the other PDATA options.
ALLPA — sets both the LPA and JPA options.
JPA — job pack area.
LPA — link pack area for this job.
PSW — program status word.
REGS — general registers.
SA — save area trace (long form).
SAH — save area trace (short form).
SPLS — allocated storage subpools.

SYSMDUMP options:

ALL — sets all of the other SYSMDUMP options.
CSA — the portions of the common storage area that are not fetch protected.
LPA — link pack area for this job.
LSQA — local system queue area.
NUC — resident nucleus.
RGN — entire private area.
SQA — system queue area.
SWA — scheduler work area.
TRT — GTF or supervisor trace data.

Resetting the Dump Type to ADD

Use the following form of the CHNGDUMP command to reset the dump type to ADD and the dump options list to initial values.

```
{CHNGDUMP}                  [,SDUMP    ]
{   CD   }      RESET       [,SYSABEND ]
                            [,SYSMDUMP ]
                            [,SYSUDUMP ]
                            [,ALL      ]
```

RESET
Set the dump mode for each dump type to ADD, and reset the system dump options list for each type to the values established during system IPL. (SYSABEND, SYSMDUMP, and SYSUDUMP options are set from values found in SYS1.PARMLIB members IEAABD00, IEADMR00, and IEADMP00 respectively. All SDUMP options are turned off.)

RESET,SDUMP or SYSABEND or SYSMDUMP or SYSUDUMP
Set the dump mode for the specified dump type to ADD, and reset the dump type's system dump options list to the values established during system IPL. (SYSABEND, SYSMDUMP, and SYSUDUMP options are set from values found in SYS1.PARMLIB members IEAABD00, IEADMR00, and IEADMP00 respectively. All SDUMP options are turned off.)

RESET,ALL
Same as RESET.

Example:

The following example (using SYSABEND) illustrates how system dump options and modes are set during system initialization and then modified by a CHNGDUMP command or supplemented by options specified on a dump request. The example consists of a matrix listing system and operator actions followed by an explanation of each action.

- The FUNCTION column lists the IPL, CHNGDUMP commands, and dump requests (ABEND) in the order they occur. An explanation of each function is provided following the matrix.
- The OPTIONS column identifies the SYSABEND dump options that are in effect at each point in the example.
- The MODE column identifies the dump mode that is in effect at each point in the example.
- Assume the SYS1.PARMLIB member IEAABD00 specifies dump options CB, DM, and ENQ.

FUNCTION (See Note)	OPTION											MODE		
	CB	DM	ENQ	ERR	IO	LSQA	NUC	SQA	SWA	TRT	JPA	NODUMP	ADD	OVER
1. *IPL*	X	X	X										X	
2. ABEND SDATA=(SWA,TRT)	X	X	X						X	X			X	
3. *CD SET,SYSABEND,SDATA=ERR*	X	X	X	X									X	
4. ABEND	X	X	X	X									X	
5. ABEND SDATA=(SWA,TRT)	X	X	X	X					X	X			X	
6. *CD SET,SYSABEND,OVER,SDATA=(CB,IO)*	X			X	X									X
7. ABEND SDATA=(SWA,TRT)	X			X	X									X
8. *CD SET,SYSABEND,SDATA=(ENQ,LSQA),OVER*	X		X	X	X	X								X
9. *CD SET,SYSABEND,SDATA=NUC*	X	X	X	X	X	X	X						X	
10. ABEND SDATA=(CB,SWA,NUC)	X	X	X	X	X	X	X		X				X	
11. *CD SET,SYSABEND,NODUMP*												X		
12. ABEND SDATA=(CB,SWA,NUC)												X		
13. *CD SET,SYSABEND,SDATA=TRT*	X	X	X							X			X	
14. *CD DEL,SYSABEND,SDATA=(DM,IO)*	X		X							X			X	
15. *CD DEL,SYSABEND,SDATA=(CB,ENQ,TRT)*													X	
16. ABEND SDATA=(SQA)								X					X	
17. *CD RESET,SYSABEND*	X	X	X										X	

Note: Operator actions and commands are shown in italics.

Figure 5.3. Example of Dump Options and Modes.

The following is an explanation of each of the functions that appear in the preceding matrix:

1. During system initialization (IPL) SYS1.PARMLIB options are set on, and the mode is set to ADD.
2. Because the mode is ADD, both the options specified on the dump request and the options set on in the options list determine the areas to dump.
3. An additional option is set on in the options list.
4. Because the dump request specifies no additional options, only the options set on in the options list determine the areas to dump.
5. Because the mode is ADD, both the options specified on the dump request and the options set on in the options list determine the areas to dump.
6. The mode is changed to OVER, and the SYS1.PARMLIB options are deleted from the options list. The only options that remain on are those that have been set on by CD commands (CD commands are cumulative).
7. Because the mode is OVER, options specified on the dump request are ignored. Only the options set on in the options list determine the areas to dump.

8. Two additional options are set on, and the mode remains unchanged.
9. The mode is changed to ADD, SYS1.PARMLIB options are set on, and options specified on the CD command are set on. Options set by previous CD commands remain on.
10. Because the mode is ADD, both the options specified on the dump request and the options set on in the options list determine the areas to dump.
11. The mode is changed to NODUMP, and all options are set off in the options list.
12. Because the mode is NODUMP, the request is ignored.
13. The mode is changed to ADD, and the SYS1.PARMLIB options and the option specified on the CD command are set on.
14. Because DEL was specified, the specified options are set off in the options list.
15. Because DEL was specified, the specified options are set off in the options list.
16. Only the option specified on the dump request determines the area to dump because there are no options on in the options list.
17. The options list and the mode are reset to the values that were established during system initialization.

Clearing the SYS1.DUMP Data Sets

When you receive the message

```
*IEA911A   COMPLETE/PARTIAL   DUMP ON   SYS1.DUMPxx/UNIT=yyy
FOR ASID (x,...x)
ERRORID=SEQ vvvvv CPU zz ASID aaaa TIME hh.mm.ss.t
```

do one of the following:

- If the dump is on a direct access volume (SYS1.DUMP), either schedule the print dump program (AMDPRDMP) to print the contents of the data set, or move the data set to another device for later processing. The latter makes the data set, which can hold only one storage dump at a time, ready to receive another storage dump.
- If the dump is on tape (UNIT=yyy), the tape is not rewound and unloaded after the storage dump is taken. Although the tape can hold more than one storage dump, it is recommended that you manually unload it and mount a fresh unlabeled tape. Schedule the print dump program (AMDRPDMP) if you wish to print out the storage dump recorded on the tape.

Your system programmer can describe the procedures for using the print dump program; a complete description of AMDPRDMP can be found in *OS/VS2 System Programming Library: Service Aids*. You can print the SYS1.LOGREC machine check and software records using the SYS1.LOGREC print routine (IFCEREP1). You should scan the records for the error identifier that is printed in message IEA911A. In this way, the three sources of diagnostic material (console log, SYS1.LOGREC records, and storage dump) can be correlated for the system programmer.

Your system programmer can describe the procedures for using the SYS1.LOGREC print routine. A complete description of IFCEREP1 can be found in *OS/VS2 System Programming Library: SYS1.LOGREC Error Recording*.

CONTROL Command

The CONTROL system command is used to control display consoles. You use this command to:

- Delete messages, either automatically or manually
- Define the use of program function keys (PFK) for consoles having PFK capability
- Halt in-line status displays
- Establish display areas
- Erase status displays
- Change the time intervals for dynamic status displays
- Define a console's mode of operation
- Number messages
- Establish and change a display console's specifications

Listed below are the keywords and parameters allowed on the CONTROL command followed by a description of their functions. Refer to Chapter 3 of this book for more information of the functions that the CONTROL command performs.

The complete syntax of the CONTROL command is:

```
{CONTROL}
{   K   }

  C,D,id[,L=cc]

  S [,CON={Y}]
          {N}
    ,REF
    ,SEG=nn
    ,DEL= {Y}
          {N}
          {R}
          {RD}
    ,RTME=nnn
    ,RNUM=nn

  D [,N, HOLD
    ,F
    ,H  [,L=cca]
    ,U
    ,PFK]

  E [[,nn] ...
    ,SEG
    ,F
    ,N
    ,D[,L=cca]
    ,PFK]

  A [[,nn] ...]      [,L=cc]
    [,NONE
    [,REF]

  T [,REF[,L=cc]
    [,UTME=nnn[,L=cc]]

  N,PFK=(nn1 {,CMD=text [,text]...}) [,CON={Y}]
             {,KEY=nn2[,nn2]...   }      {N}

  V,USE={FC} [,L=cc]
        {SD}
        {MS}
```

C,D,id L=cc

This option of the CONTROL command is used to halt the printing of a status display on a printer or typewriter that is not the hard-copy log device. This command must be entered while the display is being printed out.

C,D

The inline display, indicated by the id operand, is to be stopped.

id

The three digit identification number of the status display you wish to stop. The identification number is found in the control line of the display.

L=cc

The identification number of the active MCS console where the display is to be stopped.

Example:

```
k c,d,121,1=20
```

This example stops the status display, identification number 121, which is in progress in the general message area of console number 20.

```
                 ,CON = { Y }
                        { N }
                 ,REF
                 ,SEG=nn
      K S        ,DEL = { Y }
                        { N }
                        { R }
                        { RD }
                 ,RTME=nnn
                 ,RNUM=nn
```

S
The current console specifications are to be altered or referenced.

CON
Conversational message deletion is requested or canceled.

Y
Conversational message deletion is requested.

N
Conversational message deletion is canceled (non-conversational message deletion is to go into effect).

REF
The current console specifications values are to be displayed in the entry area in CONTROL command form. REF is the default.

SEG=nn
The size of the message segment is to be altered. nn indicates the number of lines deleted from a segment of messages when a CONTROL E,SEG command is entered. The nn value is a number from 1 to the number of lines in the message area.

DEL
The message deletion mode is to be changed.

Y
The automatic mode of message deletion is to go into effect.

N
The automatic mode of message deleting is canceled. Messages must be removed manually.

R
Roll mode is to go into effect.

RD
Roll-deletable mode of message deletion is to go into effect.

RTME=nnn
The time interval in seconds between message rolls. The nnn value can be any decimal number form 1 to 999.

RNUM=nn
The number of lines in the message roll. The nn value is a decimal number from 1 to the number of lines in the message area.

REF

The current values for the CONTROL S operands are to be displayed in the entry area in CONTROL command form.

```
K D        [ ,N,HOLD      ]
           [ ,F           ]
           [ ,H   [L=cca] ]
           [ ,U           ]
           [ ,PFK         ]
```

D

Information is to be displayed on the screen.

N

Messages on the screen are to be consecutively numbered. The numbers are to be removed from the screen when the operator deletes a message or performs a cancel action.

HOLD

In conjunction with the N operand, HOLD specifies that consecutive numbers are to be displayed for each message on the screen and that messages are to be re-numbered after each message deletion that the operator performs.

F

The next frame of a status display is to be displayed.

H

The updating of a dynamic status display is to be suppressed.

U

The updating of a dynamic status display is to be resumed.

L=a, cc, cca

The display area (a), the console (cc), or both (cca) where the specified action is to take place.

PFK

In conjunction with the D operand, PFK specifies that the numbers of the program function keys (PFKs) designated for command entry are to be displayed in the PFK display line. This operand applies only to display consoles that have the PFK function.

```
K E        [ [,nn]...     ]
           [ ,SEG         ]
           [ ,F           ]
           [ ,N           ]
           [ ,D  [,L=cca] ]
           [ ,PFK         ]
```

E

Messages are to be removed from the screen.

nn

Single messages (nn) or deletable messages within a range of messages (nn,nn) are to be removed from the screen. The value of nn must be a decimal number from 1 to the highest message number on the screen. nn,nn must be a range of numbers.

SEG

Deletable messages in the predefined message segment are to be removed from the screen.

F

All flagged messages (messages marked with a vertical or horizontal bar in position 3) are to be removed from the screen.

N

Messages on the screen are to be consecutively numbered. The numbers are to be removed when the operator deletes a message or performs a cancel action.

D

In conjunction with the E operand, D specifies that a status display is to be deleted.

L=a, cc, cca

Along with the D operand, specifies the display area (a), the console (cc), or both (cca) where the specified action is to take place.

PFK

In conjunction with the E operand, PFK specifies that the numbers are to be erased from the program function key (PFK) display line.

```
K A          [ [,nn]...          ]
             [  ,NONE    [L=cc]  ]
             [  ,REF             ]
```

A

The display area specifications are to be altered or referenced.

nn[,nn]..

The number of message lines in each display area. The first number (nn) defines the bottom area of the screen; additional numbers (,nn) define areas working toward the top of the screen. The minimum number of lines in one display area is 4. The total number of all nn's cannot exceed the number of lines in the message area of the screen.

NONE

No display areas are defined for the console.

REF

The display area sizes are to be displayed in the entry area in CONTROL command form.

L=cc

The display console whose display area is to be changed or referenced. Use this operand when changing display area specifications for output-only consoles.

```
K T          [ ,REF          [L=cc] ]
             [ ,UTME=nnn            ]
```

T

The time interval for updating status displays is to be changed or displayed.

REF

The current value of the time interval is to be displayed in the entry area in CONTROL command form. REF is the default.

UTME

The time interval for updating status displays in seconds. nnn can be a decimal number from 1 to 999.

L=cc

The console where the time interval updating or change is to occur.

Note: The following operands of the CONTROL command apply only to display consoles with program function key (PFK) capability.

```
K N,PFK=(nn1 {,CMD='text[;text]...'} )[,CON= {Y} ]
             {,KEY=nn2[,nn2...]     }        {N}
```

N,PFK

A PFK command definition is to be altered.

nn1

The number of the PFK being defined. The nn1 value must be the number of a PFK designated for command entry at system generation.

CMD

The text of one or more commands is to be associated with PFK nn1.

'text[;text]..'

The text of the operator's commands to be associated with PFK nn1. Up to 101 characters can be included within the quotes. If more than one command is to be associated with a PFK, the commands must be separated by a semicolon (do not put a semicolon after the last command). Text characters can be entered in upper or lower case; the system converts all characters to uppercase.

KEY

The commands associated with other PFK's are to be associated with nn1.

nn2[,nn2]...

The number(s) of the PFK whose commands are to be associated with PFK nn1. Up to 52 key numbers (numbers can be repeated) can be included in the list. Separate key numbers with a comma.

CON

This specifies whether conversational mode of command entry is in effect.

Y

Conversational mode of command entry is to be in effect.

N

Conversational mode of command entry is not to be in effect (non-conversational mode of command entry is to be in effect).

```
K V,USE= {SD} ,[L=cc]
         {MS}
         {FC}
```

V

The operating mode of a console is to be changed.

USE

This specifies the new mode of the console.

SD

The console is to be changed to output-only for presentation of status displays.

MS

The console is changed ot output-only for presentation of messages other than status displays.

FC

The console is to be changed to full-capability mode (input/output capability).

L=cc

The console where the specified action is to take place.

DISPLAY Command

The DISPLAY system command allows the operator to display system, I/O, and job status as well as the time of day and console status.

Through the use of the DISPLAY command, the operator can display:

- Device status and allocation
- The status of the hardware configuration (processor, channels, devices, and storage)
- The system consoles set-up and status
- CONTROL command functions
- The number of active jobs, time sharing users, and active initiators
- System requests
- A display console's PFK (program function key) associated commands (the console must have PFK capability)
- The current time and date
- The domain descriptor table
- SLIP (serviceability level indication processing) definitions
- Dump options or dump data set status
- TCAM teleprocessing functions and activity. Refer to *Operator's Library: OS/VS2 TCAM,* Level 10, for the DISPLAY command syntax and functions.
- VTAM Network activity and functions. Refer to *Operator's Library: VTAM Network Operating Procedures* for the DISPLAY command syntax and functions.
- 3850 activity and functions. Refer to *Operator's Library: IBM 3850 Mass Storage System (MSS)* under OS/VS.

The DISPLAY command supports the location operand which is used to route status displays to a specified display area or console. The syntax is as follows:

```
L= { a   }
   { cc  }
   { cca }
```

This specifies the display area (a), console (cc), or both (cca) where the display is to be presented. If you omit this operand, the display is presented in the first available display area or the message area of the console through which you entered the command (unless routing instructions are in effect; see MSGRT command in this chapter).

The complete syntax of the DISPLAY command is:

```
Command        Parameters
{DISPLAY}
{   D   }   { ( U [,TP        ] [,ONLINE ] [,xxx] [,nnn] )  (see note )
              [,GRAPHIC     ]  [,OFFLINE]                  ( below)
              [,TAPE        ]
              [,DASD        ]
              [,UR          ]

            M [=( {CPU  } [,CPU  ]...)]
                  {DEV  }  [,DEV  ]
                  {n    }  [,n    ]
                  {STOR }  [,STOR ]
                  {HIGH }  [,HIGH ]
                  {LIST }                                  ,L= {a  }
                                                               {cc }
              CONSOLES                                         {cca}
              C,K
              DMN
              DUMP
              {TS      }
              {J[OBS]  }  [,LIST]
              {A       }
              {R       }

              PFK
              SLIP
              T
              TP
              NET
              3850
```

Note 1: All commas between DISPLAY U and a specified operand must be supplied. For example, DISPLAY U,,ONLINE

Note 2: Refer to Operator's Library: OS/VS2 TCAM, Level 10, Operator's Library: VTAM Network Operating Procedures, and Operator's Library: IBM 3850 Mass Storage System (MSS) under OS/VS for the TP, NET, and 3850 DISPLAY command operands.

The specific uses and functions of the DISPLAY command are as follows:

Displaying Device Allocation

Use the DISPLAY U command to help you keep track of the status of the devices attached to the system. This command allows you to request the status of:

- A specific device type
- All devices
- Devices online
- Devices offline
- A range of devices

```
{DISPLAY} | U,[devicetype],[ONLINE ],[xxx ],[nnnn]  [,L={a  }]
{   D   }|                [OFFLINE],[000 ],[100 ]   [   {cc }]
                                                     [   {cca}]
```

Note: All commas between DISPLAY U and a specified operand must be supplied. For example, `DISPLAY U,,OFFLINE`.

U
The unit status information about all devices or the specified devices is to be displayed.

devicetype
The unit status of the indicated device type is to be displayed. The following operands can be entered for device type:

TP — communication equipment
GRAPHIC — graphic devices
TAPE — magnetic tape units
DASD — direct access storage devices
UR — unit record devices

Note: If no device type is specified, all device types are displayed.

ONLINE
Only those devices specified by the device type parameter that are online are to be displayed. If no device type parameter is specified, all devices that are online are displayed.

OFFLINE
Only those devices specified by the device type parameter that are offline are to be displayed. If no device type parameter is specified, all devices that are offline are displayed.

If neither ONLINE nor OFFLINE is specified, the status of both online and offline devices is displayed.

xxx,nnnn
The unit status information requested is to be displayed starting at device address xxx for nnnn number of devices. If xxx is omitted, the starting address is 000. If nnnn is omitted, the number of devices is 100.

Example:

```
d u,dasd,,400,8
```

This example lists the status of the first eight (if any) direct access devices that have a device address of 400 or higher.

Displaying Configuration Information

Use the DISPLAY M command to display the status of processors, channels, devices, and storage in a uniprocessing or tightly coupled multiprocessing environment.

```
{DISPLAY} | M [=CPU   ]  [,L= {a  }]
{   D   }|   [=DEV   ]       {cc }
             [=n     ]       {cca}
             [=STOR  ]
             [=HIGH  ]
             [=(list)]
```

M
The status of all processors, channels, devices, and storage is displayed when DISPLAY M is entered with no operands. For a description of the display format, refer to *OS/VS Message Library: VS2 System Messages.*

CPU
The status, 10-character serial, and model number of each processor and channels are to be displayed.

DEV
The status of all devices is to be displayed.

n
The status of all devices on channel n is to be displayed.

STOR
The status of all real storage is to be displayed. The display includes: storage offline, storage waiting to go offline, and reconfigurable storage ranges (tightly-coupled MP). For storage waiting to go offline, the address space identifier (ASID) and jobname of the user of the storage are displayed.

HIGH
The highest possible real address is to be displayed in decimal K bytes. This address is the greater of two values, the highest address specified at system generation or the highest address available at system initialization.

(list)
The status of the resources specified is to be displayed. This list consists of any combination of CPU, DEV, n, STOR, and HIGH. The resources must be separated by commas and enclosed in parentheses. Do not use blanks within the parentheses.

Examples:

```
d m=1
```

The status of all devices on channel 1 is displayed when this command is entered.

```
d m=(cpu,dev,high,stor)
```

This command causes the following to be displayed:

- The status of all processors and all channels
- The status of all devices on all channels
- The highest real storage address available
- The status of real storage

Displaying Console Configuration Information

Use the DISPLAY C command to display multiple console support (MCS) console configuration and status information.

```
{DISPLAY} {CONSOLES} [,L={a}]
{  D  }   {  C     }   [   {cc}]
                         [   {cca}]
```

CONSOLES or C
The system console configuration and status is to be displayed. Refer to *OS/VS Message Library: VS2 System Messages*, for a discussion of the output.

Displaying CONTROL Command Functions

Use the DISPLAY C,K command to request a summary of the CONTROL command operands and the functions they perform.

```
{ DISPLAY } | C,K [,L={a  }]
{   D     }   [    {cc }]
                   {cca}
```

C,K
A summary of CONTROL command operands is to be displayed.

Example:

```
d c,k,L=5A
```

This command displays the CONTROL command operands and their functions in display area A of console 5.

Displaying the Current System Activity

Use the following form of DISPLAY command to display information about the current system activity.

```
{DISPLAY} | {J    } [{,LIST}] [,L= {a  }]
{   D   }   {JOBS}  {,L   }        {cc }
            {A   }                 {cca}
            {TS  }
```

J or JOBS
The system is to display the number of:

- Active batch jobs (MOUNT commands in execution are treated as jobs)
- Started tasks
- Active initiators

A
The system is to display the number of:

- Active batch jobs (MOUNT commands and started tasks in execution are treated as jobs)
- Active time-sharing users
- Active initiators

TS
The system is to display the number of active time-sharing users and maximum number of TSO/VTAM users.

LIST or L
The system is to include the jobnames, stepname, procedure stepname, and V=R region boundaries in the JOBS or A display, or the user identifier of each active terminal in the TS or A display.

Example:

```
d jobs,list or d j,l
```

This command causes the following to be displayed:

- The number, jobname, current stepname and current procedure stepname of all batch jobs including MOUNT commands and started tasks being processed in the system.
- The number of tasks in the system and their region boundaries if they are V=R.
- The number of active initiators.

Displaying System Requests

Use the DISPLAY R command to display:

- The identification numbers and, optionally, the texts of all system requests waiting for replies.
- The unit addresses of devices waiting for mount requests to be fulfilled.
- The unit addresses of devices waiting for operator intervention.

```
{DISPLAY} | R [{,LIST} ] [,L={a  }]
{  D    }   [{,L   }]     {cc  }
                          {cca }
```

R
The system is to display the identification numbers of messages awaiting replies, unfulfilled mount requests, and any units requiring operator intervention.

LIST or L
The text of all outstanding requests is to be displayed along with the message identification. If the message text is greater than 65 characters and the 66th character is not blank, the text is truncated at the last blank before the 65th character.

Example:

```
d r,l,l=2a
```

The identifications and texts of all unanswered system requests, the unit addresses of all devices waiting for mount requests to be fulfilled, and the addresses of all devices waiting for operator intervention are to be displayed in area A of console 2.

Displaying Commands Associated with PFK Keys

Use the DISPLAY PFK command to display the commands associated with each PFK key.

Note: The console must have PFK capability.

```
{DISPLAY} | PFK
{  D    }
```

PFK

The commands defined for each PFK are to be displayed in the message area. The definitions that are displayed always refer to the console on which they are displayed. The DISPLAY PFK display cannot be routed to a display area or different console.

Displaying the Local Time and Date

Use the DISPLAY T command to display the local time of day and the date.

```
{DISPLAY}
{   D   }   T
```

T
The time of day and the date are to be displayed.

Displaying the Domain Description Table

Use the DISPLAY DMN Command to display domain information. A domain is a group of users whose characteristics are defined by the installation performance specification (IPS). The system resources manager (SRM) controls the number of swappable in-storage users in a domain or in a system (for example, the multiprogramming level). There may be from 1 to 128 domains in the system.

```
{DISPLAY}
{   D   }   DMN [=domainnum] [,L={a  }]
                               {cc }
                               {cca}
```

DMN
The domain description table is to be displayed.

domainnum
A specific domain table entry (0-128) to be displayed.

Examples:

```
display DMN=5
```

Information from domain 5 is displayed.

```
d DMN
```

Information from all entries in the domain description table are displayed.

Display a SLIP Definition

Use the DISPLAY SLIP command to display information about SLIP definitions.

```
{DISPLAY} | SLIP   [=xxxx]   [L={a  }]
{D      }                    [  {cc }]
                             [  {cca}]
```

SLIP
Indicates that summary information about SLIP definitions is to be displayed.

= xxxx
The system is to display detailed information about the SLIP definition associated with identifier xxxx. If not specified, summary information is displayed about all the SLIP definitions in the system.

Example:

```
DISPLAY SLIP
```

The DISPLAY command is used to display summary information about all SLIP definitions in the system.

Displaying Dump Options or Dump Data Set Status

To determine the dump options and dump mode currently in effect for each dump type, or to determine which SYS1.DUMP data sets are full, use the DISPLAY DUMP command.

```
{DISPLAY} {DUMP}  [({,STATUS}  )]  [,L={a  }]
{D      } {D   }  [({,S     }  )]  [  {cc }]
                  [({,OPTIONS}  )]  [  {cca}]
                  [({,O      }  )]
```

D or DUMP
Specifies that dump information is requested.

STATUS or S
The name and title of all direct access SYS1.DUMP data sets that are full, and the number of full and available SYS1.DUMP data sets, including tape data sets are to be displayed.

OPTION or O
The dump mode and dump options that are currently in effect for SDUMP, SYSABEND, SYSMDUMP, and SYSUDUMP are to be displayed.

DUMP Command

System and User Dumps

OS/VS2 MVS provides two types of dumps. System oriented dumps (SDUMPS) are taken, as required, by system recovery routines. These dumps are also taken each time you issue the DUMP command. User oriented dumps are taken by the system during error termination processing of a user task or in response to a dump request made by a user with a system macro.

System Dumps

System oriented dumps are written to the SYS1.DUMP data sets that were allocated during system initialization. These data sets may be allocated to either a magnetic tape unit or a direct access device and must remain in place for the life of the IPL.

User Dumps

A user oriented dump (SNAP, SYSABEND, SYSMDUMP, SYSUDUMP, or SVCDUMP) can be requested by a user with an ABEND, SDUMP, SETRP, or SNAP macro. In all cases, the user must provide a DD statement defining the dump data set. If the DD statement is not provided, the dump is not taken.

Use the DUMP command to request that a dump of virtual storage be stored in the pre-allocated SYS1.DUMP data set.

This data set is defined at system initialization either through the SYS1.PARMLIB member or your response to the SPECIFY SYSTEM PARAMETERS message. Refer to *OS/VS2 System Programming Library: Initialization and Tuning Guide*, for further information on this subject.

Note: For JES3, a storage dump can be requested if console lockout occurs.

The complete syntax of the DUMP command is:

DUMP	COMM=(text)

COMM=(text)
The identifier (1-100 characters) that is to appear as the first record in the dump data set and is to appear as the title on the dump.

The system prompts you with message IEE094D for the dump options you want to specify. See the REPLY command in this chapter for the REPLY command format and options.

Printing Unformatted Dumps

Use the service aid AMDPRDMP to print unformatted dumps (system dumps and SYSMDUMP or SVCDUMP user dumps). Refer to *OS/VS2 System Programming Library: Service Aids* for a complete description of this service aid.

FORCE Command

Use of the FORCE command can be considered whenever the CANCEL command fails to perform its specified function. The FORCE command is not intended to replace the CANCEL command.

By using the FORCE command, the operator can:

- Force the termination of a MOUNT command
- Force the termination of a job in execution
- Force the termination of an external writer allocation
- Force the termination of output processing for a job
- Force the termination of a terminal session

Note: Before using the FORCE command, carefully review the following considerations:

- It could be necessary to re-IPL after the use of the FORCE command. It is not possible to predict the situations that can cause this. The address space affected by the FORCE command is deleted from the system, and only limited recovery is possible.
- The FORCE command is not designed to be used in place of CANCEL. A CANCEL command must be issued before FORCE can be issued. If CANCEL is not issued first, message IEE8381 notifies you that CANCEL must be issued before FORCE. FORCE should only be used as a last resort.
- The FORCE command can only be entered from the master console; it cannot be entered through the input stream.
- If a dump of the address space is desired, issue the DUMP command before you issue the FORCE command. However, depending on the condition of the address space, a dump might not always be obtainable.

The complete syntax of the FORCE command is:

```
        {unitaddr  }
FORCE   {devicetype}
        {jobname   }
        {identifier}
        {U=userid  }
```

unitaddar
The unit address specified when the MOUNT command was issued.

devicetype
The type of device specified when the MOUNT command was issued. This is an IBM-supplied name (for example, 2314) that identifies a device by its device number. A list of IBM device types is included in *OS/VS2 System Programming Library: System Generation Reference*.

jobname
The name of the job to be forced to terminate.

Note: When the FORCE command is issued, the address space for the job being forced and any tasks executing in that address space terminate. If FORCE is issued for a job running under an initiator, the initiator also terminates. If you have JES2 on your system, you must issue another START command for the initiator; this might not be necessary if you have JES3 on your system.

identifier
The identifier can be:

- The identifier specified on the START command for the writer.
- The device type (for example, 2400) or unit address specified on the START command for the writer.
- The device type associated with the cataloged procedure specified on the START command for the writer.

U = userid
The identifier of the time-sharing terminal user you want to force off the system.

Example:

```
FORCE 2314
```

The earlier request for a 2314 is terminated.

```
FORCE JOBXYZ
```

Job JOBXYZ is removed from the system.

```
FORCE 00E
```

Device allocation for writer 00E is discontinued, and the writer is terminated.

```
FORCE 00E
```

The output processing on 00E is discontinued, and the writer is terminated.

```
FORCE U=A237
```

User A237 is logged off the system.

HALT Command

The HALT system command can be used to record statistics prior to stopping the operating system. After all subsystem processing is stopped (through the use of the appropriate subsystem command) and the system notifies you that all system activity has completed, you can issue HALT EOD command to ensure that important job and system statistics and data records in storage are recorded.

The complete syntax of the HALT command is:

{ HALT } { Z }	EOD

The HALT EOD command causes the system to take the following steps prepatory to closing down:

- The internal I/O device error counts are stored in the SYS1.LOGREC data set.
- The SMF buffers are emptied onto the SYS1.MANX or SYS1.MANY data sets.
- The system log is closed and put on the print queue.

When these actions are completed, the system sends you the message:

```
IEE334I HALT EOD SUCCESSFUL.
```

LOG Command

Use the LOG command to make an entry into the system log.

The complete syntax of the LOG command is:

{LOG / L}	'text'

'text'
The comment (up to 126 characters) to be entered into the system log. The system sends the comment to the master console if the system log is temporarily inactive; enter the command again when the log becomes active.

Note: Lowercase characters in quotes are not converted to uppercase.

Example:

```
1 'DEVICE 235 OFFLINE FOR REPAIRS'
```

The comment would be included in the system log.

MODE Command

Use the MODE command to control the actions of recovery management when certain types of machine check interruptions occur. You can control the following actions:

- The recording of system recovery or degradation machine check interrupts on the SYS1.LOGREC data set. The recording of these interrupts can be controlled by specifying the SR or DG parameters on the MODE command. Specify the recording mode by using the RECORD or QUIET parameters. If RECORD=ALL is specified, then the frequency with which the system reports the specified machine check interruption can be controlled by using the REPORT parameter.
- The enabling of the high speed buffer (Models 3032, 3033, 168, and 165-II only) after it has been disabled because of degradation machine check interruptions. Use the ENABLE parameter (in conjunction with the DG parameter) to control the enabling of the high speed buffer.
- The monitoring of hard machine checks, including machine checks that indicate timing-facilities damage. Control system monitoring of the frequency of hard machine check interruptions by specifying the PD, SD, IV, TC, PT, or CC parameters on the MODE command.
- If the RECORD=ALL parameter is specified, the system does not monitor the frequency of the specified type of hard machine check interruption occurring on the specified processor. All interruptions of the specified type occurring on the specified processor are recorded on the SYS1.LOGREC data set. If RECORD=nnn is specified, the system monitors the frequency of the specified hard machine check interruption on the specified processor. The specified processor, unless it is a uniprocessor or host processor of an attached processor (AP) system, is taken offline by alternate CPU recovery (ACR) when the specified number of hard machine check interruptions of the specified type occur on the specified processor within the specified number of seconds. A uniprocessor or host processor of an AP system enters an A01 wait state. Specifying the RECORD and INTERVAL parameters controls the number of machine check interruptions and the number of seconds.
- The recording/monitoring status for each type of machine check interruption controlled by the MODE command can be displayed by using the STATUS parameter.

You can enter the MODE command any number of times for any processor. When you issue the MODE command specifying a particular type of machine check, only the recording or monitoring mode for that type of machine check is changed, and it is changed only for the processor(s) specified (or defaulted to). To change the recording or monitoring mode for several types of machine checks, a series of MODE commands must be issued. Each MODE command would specify one type of machine check and the desired recording or monitoring mode for that type of machine check. If the MODE command is issued more than once specifying the same type of machine check, only the last (most recent) takes effect.

The complete syntax of the MODE command is:

```
MODE  { STATUS
        { SR                                  }  [,CPU= {ALL}]  [{,QUIET                                  }]
        { DG  [,ENABLE[=nnn]]                 }        {X  }    [{,RECORD [{=nnn                        }]}]
        { PD  [,INTERVAL=nnnnn]               }                 [{        [{=ALL[,REPORT=nnn]}]         }]
        { SD  [,INTERVAL=nnnnn]               }
        { IV  [,INTERVAL=nnnnn]               }
        { TC  [,INTERVAL=nnnnn]               }
        { PT  [,INTERVAL=nnnnn]               }
        { CC  [,INTERVAL=nnnnn]               }
```

STATUS
The event counters and recording/monitoring status associated with each type of machine check interruption are to be displayed for each processor. STATUS is the default; if this parameter is specified, it must be the only parameter specified.

The other parameters of the MODE command are described according to their application in the recording/monitoring of the different types of machine check interruptions.

System Recovery and Degradation Machine Check Interruptions

The following parameters are used for system recovery and degradation machine check interruptions.

DG
Degradation machine checks are to be placed in the specified recording mode.

SR
System recovery machine checks are to be placed in the specified recording mode.

CPU=ALL
All processors in the system are to be put in the specified mode.

CPU=x
The address (0 or 1) of the processor to be put in the specified mode. If the parameter is omitted, ALL is assumed.

RECORD=nnn
After the specified number (1 to 999) of system recovery or degradation machine check interruptions occur on the specified processor, the system is to notify the operator and switch the recording mode to QUIET for that type of interruption on that processor. If no number is specified or the RECORD parameter is omitted, RECORD=4 is assumed.

RECORD=ALL
All system recovery or degradation machine check interruptions occurring on the specified processor are to be recorded on the SYS1.LOGREC data set. The operator is notified each time a defined number of interruptions occur, but the system does not switch to QUIET mode. The default number of interruptions is 4 unless changed by using the REPORT=nnn parameter along with the RECORD=ALL parameter.

REPORT=nnn
The operator is notified each time the specified number (1 to 999) of system recovery or degradation machine check interruptions occur on the specified processor. This parameter may be used only in conjunction with the RECORD=ALL parameter.

QUIET
No system recovery or degradation machine check interruptions are to occur or be recorded for the specified processor.

High Speed Buffer

The following parameter is used to enable the high speed buffer.

ENABLE=nnn
The high speed buffer is to be enabled on the specified processor. Each time the specified number (1 to 255) of degradation machine check interrupts occurs on the specified processor, the high speed buffer is to be re-enabled. If no number is specified or the ENABLE parameter is omitted, ENABLE=3 is assumed. If the ENABLE parameter is used in conjunction with the QUIET parameter, no number can be specified on the ENABLE parameter. This parameter is valid only for the model 168, 165-II, 3032, and 3033 processors.

Note: If the recording mode is QUIET for DG machine check interrupts for a particular processor, the high speed buffer can be enabled only by issuing the MODE command for that processor. Automatic enabling of the high speed buffer does not occur when the recording mode is QUIET for degradation machine check interrupts.

Hard Machine Check Interrupts

The following parameters are used for hard machine check interrupts.

PD
Instruction-processing damage machine checks are to be monitored in the specified mode.

SD
System-damage machine checks are to be monitored in the specified mode.

IV
Machine checks indicating invalid PSW or registers are to be monitored in the specified mode.

TC
Machine checks indicating TOD clock damage are to be monitored in the specified mode.

PT
Machine checks indicating CPU timer damage are to be monitored in the specified mode.

CC
Machine checks indicating clock comparator damage are to be monitored in the specified mode.

CPU=ALL
All processors in the system are to be monitored in the specified mode.

CPU=x
The address (0 or 1) of the processor to be monitored in the specified mode. If the parameter is omitted, ALL is assumed.

RECORD=nnn
After the specified number (1 to 999) of hard machine checks (PD, SD, IV, TC, PT, CC) of the specified type occur on the specified processor in the specified interval, the processor is taken offline via alternate CPU recovery (ACR) unless the processor is a uniprocessor or host processor of an AP system. A uniprocessor or host processor enters an A01 wait state. All interrupts of that type occurring on that processor are recorded on the SYS1.LOGREC data set until ACR is invoked, including the interrupt that caused the invocation of ACR. If no number is specified or if the RECORD parameter is omitted, RECORD=5 is assumed.

RECORD=ALL

All specified hard machine check interrupts occurring on the specified processor are to be recorded on the SYS1.LOGREC data set. The system no longer monitors the frequency of hard machine check interrupts of that type occurring on that processor.

INTERVAL=nnnnn

This parameter is used in conjunction with the RECORD=nnn parameter. It defines the number of seconds used in counting hard machine check interrupts. If the specified number of seconds elapses before the specified number of interrupts of the specified type occur on the specified processor, the count of that type of interrupt is set to zero, and the counting is started again from zero. If the specified number of hard machine check interrupts do occur in the specified interval, then the system invokes ACR to take the failing processor offline. If the INTERVAL parameter is omitted, then INTERVAL=300 is assumed.

The following table shows the parameters allowed for each interrupt type:

	CPU=	RECORD=	REPORT=	QUIET	ENABLE=	INTERVAL=
DG	Y	M,T	T	M,E	Y, E	
SR	Y	M,T	T	M		
PD	Y	Y,X				Y,X
SD	Y	Y,X				Y,X
IV	Y	Y,X				Y,X
TC	Y	Y,X				Y,X
PT	Y	Y,X				Y,X
CC	Y	Y,X				Y,X

Y – YES
M – QUIET and RECORD= are mutually exclusive
T – REPORT= can only be used with RECORD=ALL
X – RECORD=ALL and INTERVAL are mutually exclusive
E – If QUIET is specified, ENABLE can be specified, but not ENABLE=n

Figure 5.4. Parameters Allowed for Hard Machine-Check Interrupts

Note: When you specify more than one option, the parameters can be entered in any order and must be separated by commas. For the procedure to follow when printing the SYS1.LOGREC data set, refer to *OS/VS2 System Programming Library: SYS1.LOGREC Error Recording*.

Examples:

```
MODE DG,CPU=1,ENABLE=2
```

Degradation machine check interrupts are to be counted on processor 1. If the default number (4) occurs, the system notifies the operator and switches the recording mode to QUIET for these interrupts. The high speed buffer is enabled on processor 1, after 2 degradation machine check interrupts, and again after 2 more (total of 4) degradation machine check interrupts have occurred. The recording mode for DG interrupts is switched to QUIET after 4 interrupts; automatic buffer enablement also stops when the recoding mode is switched to quiet after 4 degradation machine check interrupts have occurred. No other processors in the system are affected and no other types of machine check interrupts for processor 1 are affected.

```
mode pd,record=7,interval=600,cpu=0
```

Instruction processing damage machine check interrupts are to be monitored on processor 0. If 7 of these interrupts occur in 600 seconds on processor 0, the system invokes ACR to take processor 0 offline.

```
MODE IV,CPU=ALL,RECORD=ALL
```

All machine check interrupts indicating invalid PSW or registers are to be recorded on the SYS1.LOGREC data set but are not monitored for any processor in the system.

```
mode sd
```

The frequency of system damage machine check interrupts is to be monitored on all processors using the default values of 5 for the RECORD=parameter and 300 for the INTERVAL=parameter. After 5 system damage machine checks have occurred on a given processor within 5 minutes (300 seconds), the system invokes ACR to take that processor offline.

```
MODE DG,QUIET,ENABLE,CPU=1
```

Degradation machine checks are to be put in QUIET mode on processor 1. The high speed buffer is enabled on processor 1. Automatic re-enabling of the buffer does not occur because MODE is QUIET for degradation machine checks on processor 1.

MODIFY Command

The MODIFY system command is used to change the existing parameters of an existing job, TSO/TCAM, TSO/VTAM, or an external writer. When issued, any specified changes take place immediately.

The complete syntax of the MODIFY command is:

Command	Parameters
{MODIFY | F}	[procname[,identifier,]
	Jobname Parameters
	For TCAM Only:
	TS= {START | STOP} [,member]
	For TSO/VTAM Only:
	,USERMAX=nnnnn
	,USER= {SIC | FSTOP}
	For External Writer Only:
	,CLASS=[classes]
	,JOBID=[job-id]
	,WRITER= [STDWTR | user-writer-name]
	,FORMS=[forms-name]
	,DEST= [LOCAL | remote-workstation-name]
	PAUSE= {FORMS | DATASET}]]

Modifying Job Parameters

Use the MODIFY command to change the characteristics of a job. The specified parameters are modified only if the programmer has set the proper indicators.

{MODIFY | F}	jobname,parameters

jobname
The jobname used on the job card.

parameters
Parameters specified by a programmer that change the corresponding parameters in a job currently being processed.

Example: To change the parameters in jobname tmasgx02 by specifying 00 as indicated by the programmer, enter:

```
f tmasgx02,00
```

Starting TSO/TCAM Time Sharing

Use the following form of the MODIFY command to start TSO/TCAM time sharing once TCAM is active.

{MODIFY / F}	[procname.]identifier,TS=START[,member]

procname
The name of the cataloged TCAM procedure, in SYS1.PROCLIB, used to start TCAM.

identifier
The user-determined identifier used on the START command to identify TCAM.

TS=START
Initiate TSO/TCAM time sharing.

member
The SYS1.PARMLIB member containing TSO/TCAM time sharing system parameters.

Example: The following example modifies the TCAM procedure TCAM2 to start TSO/TCAM time sharing. To use the time sharing system parameters in SYS1.PARMLIB member IKJPRM02 rather than the default parameters, enter:

```
s tcam2
f tcam2,ts=start,ikjprm02
```

Stopping TSO/TCAM Time Sharing

Use the following form of the MODIFY command to stop TSO/TCAM time sharing.

{MODIFY / F}	[procname.]identifier,TS=STOP

procname
The cataloged TCAM procedure in SYS1.PROCLIB used to start TCAM.

identifier
The user-determined identifier used on the START command to identify TCAM.

TS=STOP
Halt TSO/TCAM time sharing. If you enter TS=STOP a second time before time sharing activity has completely ended, you are asked to respond to the following message:

```
* id IKJ024D TIME SHARING STOP IN PROGRESS,
        REPLY 'U' OR 'FSTOP'
```

Reply 'U' to allow the system to wait for time sharing to stop normally. This allows terminal users to receive all messages queued for them. If the system is unable to stop normally, reply 'FSTOP'. This reply forces time sharing to stop immediately.

Example:

```
f tcam2,ts=stop
```

This example modifies the TCAM procedure TCAM2 in order to stop TSO/TCAM time sharing.

Modifying TSO/VTAM Time Sharing

Use the following form of the MODIFY command to control the number of users allowed to be logged on to TSO/VTAM and to terminate TSO user address spaces.

```
{MODIFY}   [procname.]identifier
{  F   }   [,USERMAX=nnnnn]
           [,USER={SIC   }]
                  {FSTOP}
```

procname
The name of the cataloged procedure that was used in the START command to start TSO/VTAM time sharing.

identifier
The identifier that was specified in the START command that started TSO/VTAM time sharing.

USERMAX=nnnnn
The maximum number (0 to 32,767) of users that can be logged on to TSO/VTAM time sharing at one time. Note that specifying USERMAX=0 causes terminal control address space (TCAS) to suppress logons.

USER=SIC
Causes TCAS to cancel all TSO/VTAM terminal user address spaces normally. The terminal users do receive any messages queued for them. TCAS remains active.

USER=FSTOP
Forces TCAS to cancel all TSO/VTAM terminal user address spaces immediately. The terminal users do not receive any messages queued for them. TCAS remains active. Specify FSTOP only if a system problem causes SIC to be ineffective.

Specifying the External Writer Selection Criteria

Use the following form of the MODIFY command to specify the criteria that the external writer is to use in selecting data sets for processing.

```
{MODIFY}  [procname.]identifier
{  F   }  {,CLASS=[classes]                    }
          {,JOBID=[job-id]                     }
          {,WRITER=[STDWTR                ]    }
          {        [user-writer-name      ]    }  ,...
          {,FORMS=[forms-name]                 }
          {,DEST=[LOCAL                   ]    }
          {      [remote-workstation-name ]    }
```

procname
The name of the cataloged procedure (for example, XWTR) used in the START writer command.

identifier
The identifier, from the START command, of the writer to be modified. You can also use the device address associated with the writer you are modifying as the identifier.

CLASS=[classes]
Select only data sets enqueued in the specified classes. You can specify up to eight output classes, in priority order. If classes is omitted, the external writer does not select data sets by class. CLASS can be abbreviated as C. The output classes are named in sequence without separating commas.

JOBID=[joh-id]
Select only data sets from the job with this subsystem-assigned job-id. If job-id is omitted, the external writer does not select data sets by job. JOBID can be abbreviated as J.

WRITER=

STDWTR
Select only data sets that are to be processed by the standard writer (that is, data sets that are to be processed by the IBM-supplied writer).

user-writer-name
Select only data sets that are to be processed by the named user writer.

If WRITER= is specified with STDWTR or user-writer-name, the external writer does not use the writer program as a data set selection criterion and automatically invokes the correct writer programs. WRITER can be abbreviated as W.

FORMS=[forms-name]
Select only data sets that specify this forms name.

If forms-name is omitted, the external writer does not use the forms name as a data set selection criterion, and notifies you whenever a forms change is needed. FORMS can be abbreviated as F.

DEST=

LOCAL
Select only data sets destined for the central processor complex.

remote-workstation-name
Select only data sets destined for the specified remote workstations.

If DEST= is specified without LOCAL or remote-workstation-name, the external writer does not use the destination as a data set selection criterion. DEST can be abbreviated as D.

Previously specified options remain in effect until respecified. Before the first MODIFY command is issued, the default options are:

```
CLASS=(see note),JOBID=,WRITER=,FORMS=,DEST=LOCAL
```

Note: If no default class list is specified in the cataloged procedure to start the external writer, and you do not provide a class list in the START command, the external writer does not begin processing until you enter a MODIFY command with CLASS=[classes].

The MODIFY command passes the entire command buffer, including comments, to the processor that is to be modified. Therefore, all modifiable processors should be sensitive to embedded blanks in their parameter fields.

Causing the External Writer to Pause

Use the following form of the MODIFY command to cause an external writer to pause for operator intervention.

{MODIFY / F}	[procname.]identifier,PAUSE={FORMS / DATASET}

PAUSE=FORMS
The writer is to pause when a change of forms is necessary. The writer stops when a form number other than the installation default form number is encountered on a SYSOUT DD statement. PAUSE can be abbreviated as P.

PAUSE=DATASET
The writer is to pause before starting to process each data set. The writer can be restarted by entering a single character response to message IEF382A when you are ready to continue processing. PAUSE can be abbreviated as P.

Example:

```
F 00E,PAUSE=DATASET
```

Device 00E stops before processing each new data set.

MONITOR Command

Use the Monitor command to continually display jobnames, data set status, and time sharing user sessions.

The complete syntax of the MONITOR command is:

```
{MONITOR}   {JOBNAMES}   [,T]
{   MN  }   {SESS    }   [,T]
            {DSNAME  }
            {SPACE   }
            {STATUS  }
```

JOBNAMES
The name of each job is to be displayed when the job starts and terminates, and unit record allocation is to be displayed when the step starts. If a job terminates abnormally, the job name appears in a diagnostic message.

SESS
The user identifier for each time sharing terminal is to be displayed when the session is initiated and when it is terminated. If the terminal session terminates abnormally, the user identifier appears in the diagnostic message.

DSNAME
The system is to display, in the mount messages, the name of the first nontemporary data set allocated to the volume to which the messages refer. No data set name appears in messages for data sets with a disposition of DELETE.

SPACE
The system is to display, in demount messages, the available space on the direct access volume.

STATUS
The data set names and volume serial numbers of data sets with dispositions of KEEP, CATLG, or UNCATLG are to be displayed whenever they are freed.

T
The system is to display the time (in hours, minutes, and seconds) along with the user identifier or job name information.

Example:

```
mn jobnames,t
```

The system is to display job name information and the time whenever a job starts or terminates.

MOUNT Command

Use the MOUNT command to allow allocation of a non-JES3 I/O device to all job steps that require a particular volume without intervening demountings and remountings of the volume. There is a short delay after you issue the command before the volume is mounted since the MOUNT command must be scheduled by the system. Note that the MOUNT and UNLOAD commands must not be used for JES3-managed devices; refer to *Operator's Library: OS/VS2 MVS JES3 Commands*.

If you are mounting an IBM 3348 Model 70F Data Module, you must make sure that the 3340 drive specified has the fixed-head feature installed.

Once a MOUNT command has been executed, the specified device becomes reserved and remains mounted and reserved until an UNLOAD or VARY OFFLINE command is issued. A reserved direct access volume can be assigned the USE attribute of PUBLIC, PRIVATE, or STORAGE. A reserved tape volume can be assigned the USE attribute of PRIVATE or PUBLIC.

The complete syntax of the MOUNT command is:

```
{MOUNT}  {unitaddr   }                  [          {STORAGE}]
{  M  }  {devicetype}, VOL=({NL},serial) [,USE={PUBLIC }]
                            {SL}          [         {PRIVATE}]
                            {AL}
```

unitaddr
The address of the input/output device to be mounted.

devicetype
The type of device to be mounted. It can be any IBM-supplied name (for example, 2314). A list of IBM device types is included in *OS/VS2 System Programming Library: System Generation Reference*.

VOL=(NL,serial)
The volume specified does not have a standard label. This parameter must not be used for direct access volumes. The serial number, up to six characters long, is used for allocation references.

VOL=(SL,serial)
The volume specified has a standard label. The serial number, up to six characters long, is used for label verification and allocation references. The VOL parameter is optional when using SL tapes.

Note: Tape label verification is not performed until the tape is opened.

VOL=(AL,serial)
The volume has an American National Standard label. The serial number, up to six characters long, is used for label verification and allocation references. AL can be specified only if it was selected as an option at system generation. The VOL parameter is optional when using AL tapes.

Note: Tape label verification is not performed until the tape is opened.

USE=STORAGE, PUBLIC, or PRIVATE
The USE attribute, defined by your installation procedures, to be assigned to the specified volume. Refer to *OS/VS2 System Programming Library: Job Management*, for further discussion of USE attributes.

Note: The MOUNT command is not recommended for use in a JES3 complex and must not be used for JES3 managed devices.

Example:

```
m 282,vol=(sl,222222),use=private
```

This example causes the system to request that a volume with a standard label of 222222 be mounted on device 282.

MSGRT Command

Use the MSGRT command to establish message routing defaults for displays from the DISPLAY, TRACK, STOPTR, or CONTROL system commands to a specified message area, console, or both. The defaults specified remain in effect until another MSGRT command is issued for that message area or console. The command can route display information or stop the routing of display information.

The complete syntax of the MSGRT command is:

```
{MSGRT}  {{D=(operand[,operand]...)} [,L={a}]}
{ MR  }  {{TR=A                      }  [   {cc}]}
         {{K                         }  [   {cca}]}
         {
         {NONE
         {REF
         {CONT
         }
```

D=operand

The output produced by the DISPLAY command is to be routed to the specified MCS console. Any display operand that produces a status display is valid on this command. If only one operand is specified, the parentheses are not necessary. The display operands are:

A

Display the number of active batch jobs, active time-sharing users, and active initiators.

C

Display a summary of the CONTROL command operands and the functions they perform.

CONSOLES

Display status of system console configuration.

D

Display the domain description table.

DUMP

Display DUMP options and status of the SYS1.DUMPxx data sets.

M

Display status of all processors, channels, devices, and storage.

R

Display identification numbers of messages awaiting replies, mount requests not fulfilled, and any units requiring operator intervention.

S

Display status of SLIP definitions.

U

Display unit information about all devices.

3850

See *Operator's Library: IBM 3850 Mass Storage System (MSS) Under OS/VS,* for information concerning MSS commands.

TR=A
The display produced by the TRACK command and the action of the STOPTR command are to be routed to the specified MCS console.

K
The action caused by subsequent CONTROL commands is to affect the specified MCS console until the option is discontinued. This operand can be entered only from a non-display console. Only those CONTROL functions that can be explicitly routed by the L=cca can be default routed via MSGRT.

L=a, cc, or cca
The display area (a), console identifier (cc), or both (cca) of the active MCS console where the specified display is to appear.

NONE
All message routing defaults currently in effect for the entering console are discontinued.

REF
The routing defaults currently in effect are to be displayed.

CONT
Continuation of a previous MR REF command is specified. If this operand appears in the display of an MR REF command, it indicates that all current routing defaults have not been displayed. Issuing MR CONT causes the remaining defaults to be displayed.

Note 1: You can enter multiple DISPLAY (D), TRACK (TR), and CONTROL (K) command routings in one MSGRT command by enclosing the operands in parentheses and separating them with commas, for example, MR (D=(A,C,R),L=1),(K,L=2).

Note 2: Routeable commands internally generated by the system can override routing defaults for a console set by the MSGRT command for that one issuance of the command only. For example, where a VARY command is issued for a range of devices, the VARY command processor issues a DISPLAY command, defining the target console for the output as the issuer of the VARY command. For that one issuance of the DISPLAY command, the routing defaults for the console are overridden. In the case of an instream command or a command from member COMMNDxx of SYS1.PARMLIB, the master console receives the output.

Examples:

```
mr k,l=2
```

The action caused by the CONTROL command is to affect console 2.

```
mr tr=a,l=02
```

All output produced by the TRACK command is displayed on console 2.

Stopping Message Routing

To stop established message routing, enter the applicable MSGRT command without the L operand. For example, to remove routing defaults established for the CONTROL command, enter:

```
mr k
```

PAGEADD Command

Use the PAGEADD command only at the request of your system programmer. PAGEADD adds auxiliary storage space (page and swap data sets) to the system. The page and swap data sets added remain available to the system until you IPL with the CLPA (create link pack area) option.

You might need to add auxiliary storage space if any of the following conditions exist:

- The planned system load increases
- The space provided during system initialization proves insufficient
- Space is lost because of a hardware failure

If the system detects a shortage of available auxiliary storage space, it issues the following message:

```
IRA200I AUXILIARY STORAGE SHORTAGE
```

The system reject logons and START commands until the shortage is relieved. If the shortage increases, the following message is issued:

```
IRA201I CRITICAL AUXILIARY STORAGE SHORTAGE
```

The system rejects logons and START commands and might delay the starting of certain initiators until the shortage is relieved.

Only one type of data set (page or swap) can be added to the system with a single PAGEADD command. Requested page or swap data sets are placed in use in the order specified in the command. The system informs you when each is available for use.

The number of swap or page data sets that can be added to the system is limited by the number specified on the PAGNUM system parameter (see *OS/VS2 System Programming Library: Initialization and Tuning Guide*). If these limits are exceeded, the system issues a message and no more data sets of that type can be added during this IPL. However, if the limit has been exceeded for one type of data set it could still be possible to add data sets of the other type.

The page or swap data sets must be defined before issuing the PAGEADD command. If the volume containing the data set is not mounted before entering the command, the system issues a mount message. For information on defining data sets, see *OS/VS2 System Programming Library: System Generation Reference*.

The complete syntax of the PAGEADD command is:

```
{ PAGEADD }  { dsname[,dsname2,dsname3,...]          }
{   PA    }  { SWAP=dsname[,dsname2,dsname3,...]     }
```

dsname[,dsname2,dsname3,...]
The name of one or more page data sets to be added.

SWAP=dsname[,dsname2,dsname3,...]
The name of one or more swap data sets to be added.

Examples:

```
pageadd page3
```

Add one page data set.

```
pa SYS1.PAGE01,SYS1.PAGE02,PAGE3
```

Add three page data sets.

```
pageadd SWAP=swp11,swap9
```

Add two swap data sets.

QUIESCE Command

Use the QUIESCE command when you want to put the system in a MANUAL state without affecting job step timing; for example, when you want to alter storage. The system can be restarted by performing the RESTART function.

QUIESCE	

Current activity is suspended and the system enters a MANUAL state or a wait state with a code of hexadecimal 00000CCC. Refer to the *OS/VS Message Library: VS2 System Codes*, for further information on wait state code CCC. When the RESTART function is performed, you receive the following message and the system continues normal processing:

```
IEE752I QUIESCE WAS SUCCESSFUL
```

Note: The QUIESCE command can be entered only from the master console.

REPLY Command

Use the REPLY command to respond to system requests for information. The system associates an identification number with each request. The REPLY command for a specific request must contain the same identification number as the request.

Using the REPLY command, you can:

- Reply to system requests for information
- Specify system parameters
- Set the time of day clock and specify IPS (installation performance specification) parameters
- Specify SMF options
- Specify dump options as a result of issuing the DUMP command.

The complete syntax of the REPLY command is:

```
{REPLY}   id, { 'text'
{  R  }       'textparm=,'
              'parm=,parm'
              'parm=value'
              'parm=(value[,value]...)'
              'parm=(value,value),parm=value'
              'DATE=yy.ddd,[CLOCK=hh.mm.ss][,GMT][,IPS=nn]

              ,U                                  [,SDATA=[=(option[,option]...)]]
              {,ASID=(n[,n]...)          }         ,STOR=(beg,end[,beg,end]...)
              {,JOBNAME=(name[,name]...)}         {,CONT}
              {,TSONAME=(name[,name]...)}         {,END }
              [,SDATA[=(option[,option]...)]]
              ,STOR=(beg,end[,beg,end]...)
              {,CONT}
              {,END }
              }
```

Replying to System Information Requests

At times, the system issues requests that require an operator response. This response is accomplished through use of the REPLY command. (See "Displaying System Requests" under the DISPLAY command in this chapter for further information.)

```
{REPLY}   id [, { 'text' }]
{  R  }        { text   }
```

id

The identification number (0-99), as specified in the message requesting a response. Leading zeros can be omitted.

'text'

The response to the message. The apostrophes are optional and need only be included if your answer contains uppercase and lowercase characters or a comment. If you include the apostrophes and your answer contains an apostrophe, two apostrophes must be used in the message text.

A blank or EOB entered immediately after the identification number indicates a null reply. Comments can be included following the blank in a null reply. Once JES2 or JES3 is active, there is an abbreviated format for the REPLY command which does not require the entry of either REPLY or R (refer to the appropriate operator's library publication, OS/VS2 JES2 commands or JES3 commands).

Examples:

```
r 03,u
```

The answer to system message 3 is 'U'.

This is the system command format for responding to a system information request.

```
r xx, 'text' please call the operator
```

In this example, you can add a comment after the text of the reply to communicate additional information to the programmer issuing a WTOR (write-to-operator with reply). See 'System Command Format' in this chapter concerning the use of the comment field.

Specifying System Parameters

When the nucleus initialization program (NIP) has completed its preparatory work, the system is ready to begin processing. You receive the following message:

```
* 00 IEA101A SPECIFY SYSTEM PARAMETERS FOR RELEASE xx.yy.zzz
```

where xx is the release number, yy is the release level, and zzz is the system type.

You must reply to this message with the system REPLY command or enter EOB. The abbreviated JES replies are invalid because JES is not yet active. You can accept the default system parameters or use the following form of the REPLY command to enter the parameters given to you by your system programmer.

```
{REPLY}
{  R  }   id,  {'U'                                  }
               {'parm=,'                             }
               {'parm=,parm,'                        }
               {'parm=value'                         }
               {'parm=(value[,value]...)'            }
               {'parm=(value,value),parm=value'      }
```

id

The identification number (0-99) as specified in the message requesting information. Leading zeroes can be omitted.

'U'

No parameters are to be changed. The system uses the default list of system parameters in SYS1.PARMLIB. Entering EOB has the same effect.

'parm = ,'

The parameter, as specified in SYS1.PARMLIB, is to be canceled for this IPL. If a system default exists for this parameter, it is used.

'parm = ,parm'

The parameters, as specified in SYS1.PARMLIB, are to be canceled for this IPL. If system defaults exist for these parameters, they are used.

'parm = value'
'parm = (value[,value]...)'
'parm = (value,value),parm = value'

The specified parameters are to override the corresponding parameters in SYS1.PARMLIB. When specifying system parameters, (1) A blank or comma must separate multiple parameters and (2) U is not a valid value for a parameter. The reply can be no greater than 80 characters per line. If the reply is longer than one line, follow the last parameter with a comma or a blank and CONT. The system prompts you for the remaining values; for example:

```
R 00,'MLPA=(00,01,02,CONT'
IEA116A CONTINUE SYSTEM PARAMETERS
R 00,'03,L),BLDL=02'
```

L

A list of system parameters is to be supplied.

If you are uncertain of the format of a system parameter, refer to the *OS/VS2 System Programming Library: Initialization and Tuning Guide*, or ask your system programmer.

After you have specified the system parameters, the system asks you to state the reason for the IPL:

```
* id IFB010D ENTER 'IPL REASON, SUBSYSTEM ID,' OR 'U'
```

You must reply to this message with the system REPLY command.

Setting the Time-of-Day Clock and Specifying the Installation Performance Specification

Once the system has been initialized, it can issue one of two messages depending on whether the time-of-day clock is set or not.

If the time-of-day (TOD) clock is not set, the system asks you to set it:

```
* id IEA886A TOD CLOCK(S) MUST BE SET
```

Use the following form of the REPLY command to set the time of day clock and specify the installation performance specifications.

```
R id,'DATE=yy.ddd[,CLOCK=hh.mm.ss][,GMT][,IPS=nn]'
```

where yy is the year (00-99), ddd is the day (001-366),hh is the hour (00-23),mm is the minute (00-59), and ss is the second (00-59).

Note: The apostrophes in the above reply are optional.

If you included GMT in your reply, the time and date are Greenwich mean time. If you omitted GMT, the system assumes the values are the local time and date, converts them to Greenwich mean values, and sets the clock(s) with the Greenwich mean time. You can also specify the installation performance specification (IPS) at this time by including IPS=nn in your reply. nn is the two alphabetic characters indicating the SYS1.PARMLIB member containing the performance parameters to be used. Consult your system programmer for the IPS= specification. These performance parameters can also be changed, once the system is active by entering the SET command. See "Resetting Performance Specification," under the SET command later in this chapter for further discussion of the command.

When you have entered a valid reply to message IEA886A, the system issues message IEA903A which requests that you press the TOD clock security switch. Once you have successfully set the TOD clock, or if the TOD clock is already set, and you are allowed to alter it, the system displays the time and date and give you the option of accepting or changing them:

```
* id IEA888A GMT DATE=yy.ddd,CLOCK=hh.mm.ss
    IEA888A LOCAL DATE=yy.ddd,CLOCK=hh.mm.ss REPLY U, OR GMT/LOCAL TIME
```

If the values are acceptable, reply 'U'. If you want to change the value of the TOD clock, enter the date and/or clock with GMT. You may also specify the IPS parameter as described above. The reply format follows:

```
R id,'[DATE=yy.ddd][,CLOCK=hh.mm.ss][,GMT][,IPS=nn]'
```

Note: Message IEA888A is automatically issued at IPL time if TOD=PROMPT is included in the active PARMLIB member COMMNDxx. (See *OS/VS2 System Programming Library: Initialization and Tuning Guide*, for details.)

If you specified a different TOD clock setting, message IEA903A is issued asking you to press the switch as described above. If you omitted GMT, the local date and/or time are assumed. If you want to specify a high local date, specify 99.365 (not 99.366). Once you have set the new time and/or date, and/or IPS, the system re-issues message IEA888A with new values. Reply to the message as previously described.

Note: In MVS, the TOD clock should be set to a value based on zero being equivalent to 00 hours, 00 minutes, 00 seconds on January 1, 1900 GMT. During an IPL, the TOD clock may contain a value that, relative to this base, is not correct. This can happen, for example, when the processor has just completed running MVT, or when C.E. servicing left the clock in the error state. In such cases, to ensure the local time and date are correct, specify GMT prior to setting the local time and date.

Specifying SMF Options

If OPT=YES was specified in the SMFPRMxx member of SYS1.PARMLIB, the SMF options can be altered and you receive the following message:

```
IEE354I SMF PARAMETERS
```

The system lists the SMF parameters specified at system generation time and issues the following message:

```
* id IEE357A REPLY WITH SMF VALUES OR U
```

If you do not want to change the SMF values, enter:

```
R id,'U' or R id
```

Note: The apostrophes in the above message are optional.

If you want to change the SMF values, use the following form of the REPLY command:

```
R id,'options'
```

where 'options' is the list of SMF parameters to be changed. These options must be separated by commas and enclosed in apostrophes. The SMF options, generally provided by your system programmer, are listed here for your reference. A discussion of these parameters in greater detail can be found in *OS/VS2 System Programmer's Library: System Management Facilities (SMF).*

SMF Options

```
BUF=n

       {0}
DSV=   {1}
       {2}
       {3}

EXT=   {YES}
       {NO }

JWT=    n

       {NONE}
MAN=   {USER}
       {ALL }

OPI=   {YES}
       {NO }

OPT=   {1}
       {2}

REC=   {0}
       {2}

SID=xxxx
```

The specification of MAN=ALL allows SMF recording and MAN=NONE prevents it. MAN=USER allows the recording of user-written records.

Once specification of system options is complete, JES2 or JES3 is automatically invoked if your installation has provided for that capability.

Specifying Dump Options

After issuing a DUMP command, the system prompts you with message IEE094D for the options you want to specify. The system REPLY format is as follows:

```
         |     {   ,U                                 [,SDATA[=(option[,option]...)]] }
         |     {  {,ASID=(n[,n]...)              }    [,STOR=(beg,end[,beg,end]...) ] }
         |     {  {,JOBNAME=(name[,name]...)     }    [{,CONT}                      ] }
{REPLY}  |     {  {,TSONAME=(name[,name]...)     }    [{,END }                      ] }
{R    }  | id  {                                                                      }
         |     {  [,SDATA[=(option[,option]...)]]                                      }
         |     {  [,STOR=(beg,end[,beg,end]...) ]                                      }
         |     {  [{,CONT}                      ]                                      }
         |     {  [{,END }                      ]                                      }
         |
         |     Note: CONT or END, when specified, must be
         |           the last parameter on the input line.
```

id
The identification number (0-99), as specified in system message IEE094D. The leading zero can be omitted.

U
ASID=1 (master scheduler address space) and the SDATA default options are to be used.

ASID=(n)
The decimal address space identification number of the address space(s) that you want dumped. You can specify up to 15 address space identification numbers. If you specify only one number, parentheses are not required.

JOBNAME=(name)
The jobname of the background job whose address space you want dumped. You can specify a maximum of 15 jobnames (up to 120 characters including the keyword JOBNAME). If you specify only one name, parentheses are not required.

TSONAME=(name)
The user identification name of the time sharing user whose address space you want dumped. You can specify a maximum of 15 names (up to 120 characters including the keyword TSONAME). If you specify only one name, parentheses are not required.

Note 1: If neither ASID, JOBNAME, nor TSONAME is specified, the master address space is dumped.

Note 2: The dumping of several large address spaces such as the TCAM/VTAM, master, and job entry subsystem address spaces or large system areas such as the CSA and SQA results in significant system degradation. Only the address spaces that are likely to be involved in a problem should be dumped. For example, if a TSO terminal is having a problem, only the TSO user's address space and the TCAM or VTAM address space should be dumped.

SDATA=option
The specific areas to be dumped. The valid options and their definitions are:
PSA — prefixed storage area for all processors.
NUC — resident nucleus.
SQA — system queue area.
LSQA — local system queue area.
RGN — private area of address space being dumped, including LSQA and SWA.
TRT — GTF incore trace buffers (when GTF is active) and supervisor trace.
SWA — scheduler work area.
LPA — link pack area for the dumping task.
CSA — common service area.
SUM — summary dump.
NOSUM — no summary dump.
NOPSA — no prefixed storage area.
NOSQA — no system queue area.
If SDATA is specified without any options, PSA, SQA, LSQA, TRT, CSA, SUM, and SWA are assumed.

STOR=(heg,end)
The range of virtual storage to be dumped. These addresses can be specified as 3-byte hexadecimal numbers, such as 0BA040, or 5-digit decimal numbers followed by a K, such as 00508K.

CONT
Specifies that you want to continue the reply on another line. The system reissues message IEE094D in response to this parameter, after which you can continue your reply. The only parameters that can be specified on a continuation line are SDATA, STOR, CONT, and END.

END
Identifies the end of the command input and is only required if the reply line contains no other parameters. If there are no parameters specified on the same line as the END parameter and the previous line contains no parameters other than CONT, the system dumps the master address space without options.

Note: It is recommended that you have no more than one dump request in process at any time. If you have more than one request in process, you might have a problem determining the dump request that caused a particular occurrence of prompting message IEE094D.

Examples:

```
DUMP COMM=(DUMP FOR PAYROLL)
```

The system issues the following message:

```
* id IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND
R id,JOBNAME=PAYROLL,SDATA=(NUC,RGN,LSQA,TRT)
    ,STOR=(0CD450,0FF400,00012K,00060K)
```

The system dumps the resident nucleus, virtual address space, local system queue area, and GTF incore trace buffers for the job named PAYROLL. It also dumps the contents of locations 0CD450 to 0FF400 and locations 00012K to 00060K. Each page of the dump is titled DUMP FOR PAYROLL.

```
DUMP COMM=(MASTER SCHEDULER MEMORY)
* id IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND
R id,SDATA,STOR=(000000,FFFFFF)
```

The system dumps the prefixed storage area, system queue area, local system queue area, GTF incore trace buffers (if GTF is active), common service area, and the scheduler work area for the master scheduler address space. It also dumps the contents of location 000000 through FFFFFF. Each page of the dump is titled MASTER SCHEDULER MEMORY.

```
DUMP COMM=(LOOP IN ASID 6)
* id IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND
R id,ASID=(6,1,2,3,11,12),CONT
* id IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND
R id,SDATA=(NOSUM,PSA,CSA,SQA)
```

The system dumps the prefixed storage areas, common service area, and system queue area of the six ASIDs. The summary dump option is not invoked. Each page of the dump is titled LOOP IN ASID6.

```
DUMP COMM=(TSO USER TERMINAL IN LOOP)
* id IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND
R id,TSONAME=(TERMNAL,CONSOLE),CONT
* id IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND
R id,END
```

The system dumps the time sharing address spaces named TERMINAL and CONSOLE along with the default SDATA options. Each page of the dump is titled TSO USER TERMINAL IN LOOP.

RESET Command

Use the RESET command to change the performance group of a job currently in execution. This command affects the current job step and all subsequent job steps in this execution. This command should be used only at the direction of the system programmer.

The complete syntax of the RESET command is:

{RESET / E}	jobname,PERFORM=n

jobname
The name of the job whose performance group is to be changed.

PERFORM=n
The number, between 1 and 255, to be assigned as the new performance group value.

Example:

```
e tmasgj02,perform=1
```

Job TMASGJ02 has its performance group value changed to 1.

SEND Command

The SEND command is used for communication between operators in a multiple console support (MCS) environment. In a time sharing environment, it is also used to communicate between the system operator and a specific terminal user or all terminal users, and to manage the SYS1.BRODCAST data set.

The SEND command enables an operator to:

- Communicate with other operators in an MCS environment
- Send messages to specific users or to all time sharing users currently receiving messages
- Send messages to specific time sharing users or to all time sharing users logging on the system
- Save messages in the SYS1.BRODCAST data set
- List messages in the SYS1.BRODCAST data set
- Delete messages from the SYS1.BRODCAST data set

The complete syntax of the SEND command is:

```
{SEND}  {{'message'}  {,BRDCST                                  }
{ SE }  {{msgno    }  {,OPERATOR=routecode                      }
        {             {,CN=console                              }
        {             {,USER=(userid[,userid]...)  [,LOGON]     }
        {             {                            [,NOW  ]     }
        {{'message'}  [,LOGON]                                  }
        {{msgno    }  [,NOW  ]                                  }
        {             [,USER=(userid[,userid]...)]  ,SAVE       }
        {             [,ALL                      ]              }
        {  [msgno,]   {LIST  }                                  }
        {             {DELETE}                                  }
```

Communicating with Other Operators

Use the SEND command to communicate with other operators at MCS consoles.

```
{SEND}  {'message'}  {,BRDCST                 }
{ SE }  {msgno    }  {,OPERATOR=routecode     }
                     {,CN=console             }
```

message
The message to be sent. If the system log is set up to print only upper case, the message must be upper case in order for it to be recorded on the system log.

msgno
The number of the message in SYS1.BRODCAST to be sent. (See "Saving Messages in the Broadcast Data Set," under this command, for information on how messages can be saved and later sent by message number.)

BRDCST
The specified message is to be sent to all active consoles.

OPERATOR = routecode
The routing area to receive the message. Specified as a one or two digit number between 1 and 15 (see Figure 2-4, earlier in this manual).

CN=console
The console where the message is to be sent. This must be a one or two-digit value between 0 and 99.

When you are uncertain of the routing codes and console identifiers in effect, enter the DISPLAY CONSOLES command. See the section entitled "Displaying Console Configuration Information," under the DISPLAY command in this chapter for further discussion of the DISPLAY CONSOLES command.

Example 1: To send message number 46 to console 12, enter:

```
se 46,cn=12
```

Example 2: To send the following message to all active consoles, enter:

```
se 'CLOSE DOWN IN 15 MINUTES',BRDCST
```

Communicating with Specified Users

Use the following form of the SEND command to communicate with specific time sharing users.

```
{SEND}  {'message'}                                   [,LOGON]
{ SE }  {msgno    }  ,USER=(userid[,userid]...)       [,NOW  ]
```

'message'
The message to be sent to the terminal users.

msgno
The number of the message to be sent.

USER = (userid)
The identifications of those users to receive the message.

LOGON
If any specified user is currently logged on and is accepting messages, he receives the message. If he is logged on but not receiving messages, the message is stored in the mail section of the broadcast data set until he requests it. If the user is not logged on, the message is stored in the mail section of the broadcast data set until requested when he logs on.

NOW
Specifies that the message is to be sent immediately. If the recipient is not logged on, you are notified and the message is deleted. NOW is the default value if NOW and LOGON are omitted.

Note: When possible, use the LOGON parameter so you do not interrupt the user's terminal session unnecessarily.

Example:

```
se 'your listings are ready',user=(d58,d04),logon
```

This example results in the message being sent to users D58 and D04 immediately, if they are receiving messages, or when they request messages. If they are not logged on the system, they receive the message when they log on.

```
se 'getting I/O errors on your pack',user=(payroll)
```

This command sends the message to the specified user immediately if he is logged on.

Communicating with All Terminal Users

Use the following form of the SEND command to send a message to all terminal users currently logged on the system.

```
{SEND} | {'message'} [,LOGON]
{ SE } | {msgno   } [,NOW  ]
```

'message'
The message that is to be sent to all time-sharing terminal users.

msgno
The number of the message to be sent.

LOGON
All users logged on and accepting messages receive the message. Those users logged on but not receiving messages receive it upon requesting messages. The message is stored in the mail section of the broadcast data set and is sent to those users requesting messages when they log on. The message is retained until deleted by the operator.

NOW
Specifies that the message is to be sent immediately to all users currently logged on; the message is not retained for users not logged on. NOW is the default value if NOW and LOGON are omitted.

Note: When possible, use the LOGON parameter so you do not interrupt the user's terminal session unnecessarily.

Examples:

```
SE 'system going down in 5 minutes'
SE 'time-sharing will not be up next Thursday',LOGON
```

Since the first example is critical to the user, it is sent to all users immediately. However, the second example is of general interest and can be sent to the users when they request messages or at logon time.

Saving Messages in the Broadcast Data Set

Use the following form of the SEND command to store messages in the broadcast data set to be issued only at logon time, or when requested.

```
{SEND} | {'message'} [,USER=(userid[,userid]...)] ,SAVE
{ SE } | {msgno   } [,ALL                       ]
```

'message'
The message to be sent to the terminal users.

msgno
The number of the message to be sent.

USER=(userid)
The identifications of those users to receive the message. The message is stored in the mail sections for those users. (There is a mail section for each TSO user.)

ALL
All terminal users are to receive the message. Terminal users who are currently using the system receive the message immediately. In addition, the message is placed in the notice section and assigned a number. This number is printed when the message is stored and can be used as msgno in this and other forms of the SEND command. ALL is the default value if both USER= and ALL are omitted.

SAVE
The message is to be stored in the appropriate section of the broadcast data set until a user logs on or requests messages. If ALL is specified, the message is stored in the notice section of the broadcast data set and is retained until explicitly deleted. If userid is specified, the message is stored in the mail section of the broadcast data set and deleted after it is sent to the intended user. No attempt is made to send it immediately, even to those users logged on and receiving messages.

Example:

```
se· 'time-sharing will close down at 5:00 p.m. today.',save
```

An operator can submit messages to the broadcast data set before stopping the time sharing session for the day. When he starts time sharing the next day, the messages are available for users logging on. The above command does not affect those users currently logged on and receiving messages.

Listing the Notice Section of the Broadcast Data Set

Use the following form of the SEND command to keep track of accumulated messages in the notice section of the broadcast data set. You can list one or all of the messages present.

{SEND / SE}	[msgno,]LIST

msgno
The number of the message to be listed. If this operand is omitted, all messages in the notice section are listed, along with the message numbers assigned to them.

LIST
The requested messages in the notice section of the SYS1.BRODCAST data set are to be listed on the console.

Example:

```
se list
```

All messages in the notice section of the SYS1.BRODCAST data set are listed.

```
SEND 21,LIST
```

Message number 21 is listed.

Deleting a Message from the Notice Section

If you find, after listing the notice section of the broadcast data set, that a message is no longer needed, use the following form of the SEND command to delete it.

{SEND / SE}	msgno,DELETE

msgno
The number of the message to be deleted.

DELETE
The specified message is to be deleted.

Example:

```
SE 23,delete
```

Message number 23 is to be deleted.

SET Command

The SET command is used to change the installation performance (IPS) value and to set the local time and date.

The complete syntax of the SET command is:

```
{SET}   {IPS=nn                                  }
{ T }   {[DATE=yy.ddd][,CLOCK=hh.mm.ss]          }
        {RESET                                   }
```

Resetting the Performance Specification

Use the following form of the SET command to respecify the parameters the system resources manager uses to control job scheduling. This command should be issued only at the direction of your system programmer.

```
{SET}   IPS=nn
{ T }
```

Note: The local time and date can also be set at this time.

IPS=nn
The two alphameric characters indicating the IEAIPSnn member of SYS1.PARMLIB containing the new performance parameters to be used. The new parameters take effect for jobs in progress as well as for jobs read and scheduled after the command.

Example:

```
t ips=00
```

The installation performance parameters are changed according to the values found in the IEAIPS00 member of SYS1.PARMLIB.

Changing the Local Time and Date

After system initialization, use the following form of the SET command to change the local date and time.

```
{SET}   {[DATE=yy.ddd][,CLOCK=hh.mm.ss]}
{ T }   {RESET                         }
```

Note: IPS can also be changed at this time.

DATE=yy.ddd
The year (00-99) and the day (001-366). If the new time implies a change of date, the new date must be explicitly stated. Specify 99.365 when a high date is desired, because the system does not recognize 99.366 as a high date.

CLOCK=hh.mm.ss
The time in hours (00-23), minutes (00-59), and seconds (00-59). The system does not change the date when the new time implies change of date; if you want a new date, use the DATE parameter or wait for the time to pass midnight.

RESET
The local date and time are set to the values they would now contain had you not changed them at system initialization or through a previous SET command.

Examples:

If, when the displayed local time and date are 19.00.00 and 78.231, respectively, you want to set the local time ahead to 1:00 a.m., enter:

```
T DATE=78.232,CLOCK=01.00.00
```

It is necessary to enter DATE since, in this example, 1:00 a.m. implies a change of date.

If you want to reset the time and date to accurate values, enter:

```
t reset
```

SETDMN Command

The SETDM command is used to set the parameters in a domain. A domain is a user group that has characteristics defined by the installation performance specification (IPS) and has a multiprogramming level controlled by the system resource manager (SRM). This command should be used only at the direction of your system programmer.

The complete syntax of the SETDMN command is:

```
SETDMN
SD      domainnum   , { MIN=n1  [,MAX=n2]  [,WT=n3]  }
                      { MAX=n2  [,MIN=n1]  [,WT=n3]  }
                      { WT=n3   [,MAX=n2]  [,MIN=n1] }
```

domainnum
The domain table entry (1-128).

MIN=n1
The minimum multiprogramming level (0-255).

MAX=n2
The maximum multiprogramming level (0-255). This value must be greater than or equal to the value specified in MIN=n1.

WT=n3
The weighing factor (1-255).

Note: At least one keyword must be specified, but duplicate keywords are not accepted.

Examples:

```
setdmn 5,MAX=2
```

The maximum multiprogramming level in domain 5 is set to 2; MIN and WT values remain unchanged.

```
sd 6,WT=1,MIN=3,MAX=4
```

Values of a weighing factor of 1, minimum multiprogramming level of 3, and a maximum mutiprogramming level of 4 are set in domain 6.

```
sd 8,MAX=2,MAX=4
```

The system issues an error message because duplicate keywords are used.

SLIP Command

Use the SLIP command to set, modify, and delete SLIP definitions. A SLIP definition allows the user to specify an action to be taken when certain error conditions are met. See the *OS/VS2 System Programming Library: Supervisor* for more information regarding the use of SLIP to intercept software errors. For information about designing an effective SLIP trap, see *OS/VS2 System Programming Library: MVS Diagnostic Techniques*.

The complete syntax of the SLIP command is:

```
SLIP
SL      SET   [ , {ENABLE} ]
              [   {EN    }
              [   {DISABLE}
              [   {D     }
              [
              [ ,ID=xxxx
              [
              [ , {ACTION} =action
              [   {A     }
              [
              [ , {ERRTYP} =(type[,type...])
              [   {ER    }
              [
              [ , {MODE  } =(mode[,mode...])
              [   {M     }
              [
              [ , {COMP  } =option
              [   {C     }
              [
              [ , {JOBNAME} =jobname
              [   {J      }
              [
              [ , {JSPGM } =pgmname
              [   {JS    }
              [
              [ , {PVTMOD} =(name[,addr1[,addr2]])
              [   {P     }
              [
              [ , {LPAMOD} =(name[,addr1[,addr2]])
              [   {L     }
              [
              [ , {ADDRESS} =(addr[,addr])
              [   {AD     }
              [
              [ , {ASID  } =(id[,id...])
              [   {AS    }
              [
              [ , {END   }
              [   {E     }
        DEL   , {ALL   }
              {ID=xxx}
        MOD   , {ENABLE}
              {EN    }
              {DISABLE}
              {D     }
```

Setting a SLIP Definition

Use the SET parameter of the SLIP command to set SLIP definitions.

```
SLIP
SL      SET   [ , { ENABLE } ]
                  { EN     }
                  { DISABLE }
                  { D      }
              ,ID=xxxx
              , {ACTION} =action
                {A     }
              , {ERRTYP} =(type[,type...])
                {ER    }
              , {MODE} =(mode[, mode...])
                {M   }
              , {COMP} =option
                {C   }
              , {JOBNAME} =jobname
                {J      }
              , {JSPGM} =pgmname
                {JS   }
              , {PVTMOD} =(name[,addr1[,addr2]])
                {P     }
              , {LPAMOD} =(name[,addr1[,addr2]])
                {L     }
              , {ADDRESS} =(addr[,addr])
                {AD     }
              , {ASID} =(id[,id...])
                {AS  }
              , {END}
                {E  }
```

SET
This SLIP command creates a new SLIP definition.

ENABLE or EN
The SLIP definition is to be initially active.

DISABLE or D
The SLIP definition is to be initially inactive. Use the SLIP MOD command when you want to activate this SLIP definition.

ID=XXXX
One to four alphameric characters used to identify this SLIP definition. If ID is not specified, a unique identifier is generated by the system, starting with 0001. You are notified of the system-generated ID by message IEE727I.

ACTION or A=action
The action specifies what is to take place when the error conditions are met. The following actions can be specified:

SVCD
Indicates an SVC dump is to be scheduled for the address space in control.

WAIT
Indicates the system is to be placed in a wait state (code=01B). The system can be restarted by pressing the restart key.

NODUMP
Any ABEND or SVC dumps requested by a system or user program are to be suppressed.

Notes:

1. When specifying the NODUMP action, the SLIP event should be specific. A generalized SLIP description can cause the suppression of dumps for problems other than the one intended. For example, specifying only a system completion code suppresses all dumps for that code; however, if you specify both a system completion code and a jobname, other jobs that abend with that completion code still produce dumps. A specific SLIP description with a system completion code of 806 and a jobname of D10AXH1A is specified as follows:

```
SLIP SET,ACTION=NODUMP,COMP=806,JOBNAME=D10AXH1A,END
```

2. If a second error occurs during system processing for a problem that matches a SLIP trap with an action of NODUMP, any dump requested for the second error is also suppressed. To determine if a second error has occurred, check both the job output messages and the system LOGREC output. If either contains more than one abend, a second error has occurred. If you need a dump for the second error, disable the SLIP NODUMP action and rerun the failing job.

IGNORE
Indicates the system is not to perform any further SLIP processing for this error.

If ACTION is not specified, ACTION=SVCD is the default.

The following parameters define conditions that must exist for the action specified on the ACTION keyword to occur. These parameters are: ERRTYP, MODE, COMP, JOBNAME, JSPGM, PVTMOD, LPAMOD, ADDRESS, and ASID.

ERRTYP or ER=(type[,type...])
The types specify the error conditions that must exist to cause the action specified on the ACTION keyword to occur. One or more of the following error conditions can be specified:

PROG
Program check interrupt.

REST
Restart interrupt.

SVCERR
SVC error (issuing an SVC while locked, disabled, or in SRB mode).

PGIO
Paging I/O error.

DAT
Dynamic address translation error.

MACH
Software error caused by a machine check.

MEMTERM
Abnormal address space termination.

ABEND
Task issued SVC 13.

ALL
All of the above error conditions.

If ERRTYP is not specified, ERRTYP=ALL is the default. If more than one type is specified, the occurrence of any of the conditions causes the action specified on the ACTION keyword to be taken. If ALL is specified with one or more of the other types, ALL overrides the others.

MODE or M(mode[,mode...])
The modes specify the mode the system must be in for the action specified on the ACTION keyword to occur. One or more of the following modes can be specified:

SUPR
Supervisor control mode.

DIS
Physically disabled for I/O or external interrupts.

GLOC
Any global lock is held.

GLOCSP
A global spin lock is held.

GLOCSD
A global suspend lock is held.

LLOC
A local lock is held.

LOCK
Any lock is held.

TYP1
Type 1 SVC is in control.

SRB
SRB mode.

TCB
TCB mode.

PP
Problem program.

SUPER
Supervisor state.

SKEY
System key (7 or less).

PKEY
Problem program key (8 or more).

RECV
A recovery routine is in control.

ALL
All of the above modes.

The following options specify combinations of modes to satisfy error conditions:

EVERY
Every mode specified must be satisfied to meet error conditions.

ANY
Any mode specified meets error conditions.

If MODE is not specified, then MODE=(ALL,ANY) is the default. If ANY or EVERY is not one of the specified options, then ANY is the default. If ALL is specified with one or more of the other modes, ALL overrides the others. Message IEE728D is issued if any of the following combinations is specified:

MODE=EVERY
MODE=ANY
MODE=(ANY,EVERY)

COMP or C=option
The option specifies a system completion code or a user completion code. The system completion code must be in the following format:

XXX: Replace one or more of the positions with a hexadecimal number representing a system completion code, or a set of system completion codes. For example, X22 would represent completion codes 122, 222, 322, etc.

Note: If you specify any of the following abend codes as a COMP option, the requested SLIP action is not taken:

11A, 12E, 15D, 200, 212, 279, 282, 402, 42A, 57D, 6FC, 700, 72A, A00 and B00.

Each of these codes was originally a program check (completion code '0C4') that has been converted to a new value by the system. If you specify one of these codes as the COMP option, the SLIP action is not taken because SLIP requires the original completion code. Therefore, specify either COMP=0C4, or ERRTYP=PROG on the SLIP command. To avoid having the SLIP action occur for all program checks, you should also specify a program name, module name, or other limiting condition on the SLIP command.

The user completion code must be in the following format:

UXXXX: U indicates that this is a user completion code. Replace one or more of the last four positions with a decimal number representing a user completion code, or a set of user completion codes. For example, U102X represents completion codes U1021, U1022, U1023, etc.

Note: If any user completion code is changed by a user recovery routine with the SETRP macro, specify the original completion code on the SLIP command.

JOBNAME or J=jobname
The jobname indicated by the 1 to 8 character job name that must be in control for the action specified on the ACTION keyword to occur.

JSPGM or JS=pgmname
The job step program indicated by the 1 to 8 character program name that must be executing for the action specified on the ACTION keyword to occur.

PVTMOD or P =(name[,addr1[,addr2]]

name
The private area load module specified by the 1 to 8 character name that must be in control for the action specified on the ACTION keyword to occur.

addr1
Specifies the hexadecimal offset in the module where the error must occur.

addr2
Specifies the ending offset of a range beginning at addr1 within which the error must occur.

If addr1 and addr2 are not specified, then the range is assumed to be the entire load module.

LPAMOD or L=(name[,addr1[,addr2]])

name
The link pack area load module specified by the 1 to 8 character name that must be in control for the action specified on the ACTION keyword to occur.

addr1
Specifies the hexadecimal offset in the module where the error must occur.

addr2
Specifies the ending offset of a range beginning at addr1 within which the error must occur.

If addr1 and addr2 are not specified, then the range is assumed to be the entire load module.

ADDRESS or AD=(addr[,addr])
The addr specifies the hexadecimal virtual address, or range of addresses, where the error must occur for the action specified on the ACTION keyword to occur. The addresses can be 1 to 6 characters in length.

ASID or AS=(id[,id...])
The id specifies the address space identifier that must be in control for the action specified on the ACTION keyword to occur. The identifiers can be 1 to 4 hexadecimal characters. A maximum of 16 identifiers can be specified.

END or E
Indicates that there are no more parameters for this SLIP command. If END is not specified, the user is prompted for additional parameters.

Example:

```
SLIP SET,ENABLE,ID=0002,COMP=0C4,ERRTYP=PROG,JOBNAME=JOBXYZ,
LPAMOD=MOD01,END
```

The SLIP command establishes an active SLIP definition with an identifier of 0002. The default action to be taken is the scheduling of an SVC dump. The error conditions are a 0C4 program check interrupt while job JOBXYZ is executing and module MOD01 is in control.

Modifying a SLIP Definition

Use the following form of the SLIP command to modify an existing SLIP definition.

```
{SLIP}   MOD,{ENABLE  }  ,  {ALL     }
{SL  }       {EN     }      {ID=XXXX }
             {DISABLE}
             {D      }
```

MOD
This SLIP command modifies one or more existing SLIP definitions.

ENABLE or EN
The specified SLIP definition is to be made active.

DISABLE or D
The specified SLIP definition is to be made inactive.

ALL
Every SLIP definition present in the system is to be modified.

ID=XXXX
Only the SLIP definition associated with the identifier XXXX is to be modified.

Note: To modify more than one, but not all SLIP definitions, a separate SLIP MOD command must be entered for each ID.

Example:

```
SLIP MOD, DISABLE, ID=0024
```

The SLIP command is used to deactivate the SLIP definition associated with identifier 0024. This SLIP definition can be activated at a later time with the SLIP MOD command.

Deleting a SLIP Definition

Use the following form of the SLIP command to delete a SLIP definition.

```
{SLIP}   DEL ,   {ALL     }
{SL  }           {ID=XXXX }
```

DEL
This SLIP command removes one or all SLIP definitions from the system.

ALL
Every SLIP definition in the system is to be deleted.

ID=XXX
Only the SLIP definition associated with the identifier XXXX is to be deleted.

Note: To delete more than one, but not all SLIP definitions, a separate SLIP DEL command must be entered for each ID.

Example:

```
SLIP DEL, ID=0008
```

The SLIP command is used to delete the SLIP definition associated with identifier 0008. This SLIP definition cannot be activated by the SLIP MOD command.

{ADDRESS | AD} =(start[,end])

Each address is 1 to 8 hexadecimal digits. For an error detection trap, ADDRESS specifies the virtual address at which or range of addresses within which the error must occur.

For a storage alteration PER trap, ADDRESS specifies the virtual address at which or range of addresses within which the instruction that causes the storage alteration must reside.

ADDRESS is invalid for an instruction fetch or successful branch PER trap.

The starting address must be less than or equal to the ending address.

Example:

```
ADDRESS=(CD300,CD400)
```

{ASID | AS} =(id[,id...])

Each address space identifier is 1 to 4 hexadecimal digits and you can specify up to 16 identifiers. ASID specifies the address space that must be in control when the error or PER interruption occurs.

For non-IGNORE PER traps, specifying ASID indicates the address spaces in which PER monitoring is to be active.

If you do not specify ASID on non-IGNORE PER traps, PER monitoring is to be active in all address spaces unless you limit it by specifying JOBNAME.

If you specify both ASID and JOBNAME, one of the address space identifiers must be the one in which the job is running or the trap will not match.

Example:

```
ASID=(1,7,1A)
```

{COMP | C} =option

Specifies a system or user completion code to be associated with the error.

For a system completion code, the form must be ddd, three hexadecimal digits. You can indicate a set of codes by substituting x's for one or more of the digits. For example, x23 means 123, 223, 323, 423, etc. You can use x's in any position.

Note: If you specify any of the following abend completion codes, the SLIP action is not taken: 11A, 12E, 13E, 15D, 200, 212, 279, 282, 33E, 402, 42A, 57D, 6FC, 700, 72A, A00, B00, and x22.

Most of these codes were originally program checks (code 0C4) that the system has converted to a new value. If you want to specify a program check, use COMP=0C4 or ERRTYP=PROG. To avoid having the SLIP action occur for all program checks, you should also specify a program name, module name, or other condition.

The user completion code must be in the form Uxxxx where U indicates a user code and the x's represent decimal numbers. By leaving one or more of the x's in, you can specify a set of codes. For example, U102x represents U1021, U1022, U1023, etc.

Note: If any user completion code is changed by a user recovery routine with the SETRP macro instruction, specify the original completion code in the SLIP command.

Example:

```
COMP=U123x
```

{DATA | DA} =(target[(b)],operator,value[,target[(b)],operator,value]...)

Specifies that the contents of a target location are to be logically compared to a value.

target

Specifies the address of a storage location or a general purpose register (GPR) to be compared against the value supplied. The target can be:

- A virtual address (direct address) of 1 to 8 hexadecimal digits
- A GPR in the form xR where x is 0 – 15
- An indirect address (Refer to "Indirect Addressing Used With SLIP" later in this section)

b

If specified, b modifies the target address by indicating the bit position where a binary comparison is to start. For storage locations, b can be 0 – 7. For registers, b can be 0 – 31.

operator

Specifies the relationship that must exist between the contents of the target location and the value for the comparison to be successful.

EQ – equal
NE – not equal
GT – target greater than value
LT – target less than value
NG – target not greater than value
NL – target not less than value

value

Specifies the value to which the contents of the target is to be compared.

If (b) is specified, the value is binary digits.

If (b) is not specified, the value is hexadecimal digits. For example, 5R%,EQ,01 is hexadecimal; 5R(0),EQ,01 is binary. The length of the value establishes the length of the comparison. The maximum length for a hexadecimal comparison is 4 bytes. The maximum length for a binary comparison is 8 bits. The binary comparison cannot be across a register boundary.

Example:

```
DATA=(2R,EQ,80) where 2R is the target (general purpose register 2)
                EQ is the operator (equal)
                80 is the value in hexadecimal
```

DEBUG

Allows you to determine why a trap that you set is not working as you expected by indicating which of the conditions you established is not being met. GTF and its trace option for SLIP records must be active.

DEBUG indicates that trace records are to be created each time the SLIP trap is tested. Each DEBUG record contains SLIP information plus two bytes of match/no match bit indicators. Each bit corresponds to a possible test made to determine a match for a trap. If the bit is 1, the match failed. If the bit is 0, the match was successful or not performed. (After one failure, no further tests are made.)

For a description of the DEBUG trace record and the bit indicators, refer to *System Programming Library: Debugging Handbook*.

Example:

```
DEBUG
```

{DISABLE | D}

Indicates that the trap set is to be initially inactive (ineligible for checking). When neither DISABLE nor ENABLE is specified, ENABLE is the default.

Example:

```
SLIP SET,DISABLE,...........
```

{ENABLE | EN}

Indicates that the trap defined is to be initially active (eligible for checking). When neither ENABLE nor DISABLE is specified, ENABLE is the default.

Example:

```
SLIP SET,ENABLE,...
```

{ERRTYP | ER} =(type[,type]...)

Specifies one or more error events. If you specify ALL or more than one error type, the occurrence of any one of them satisfies the match test for this keyword. You can specify one or more of the following error events.

PROG	Program check interruption
REST	Restart interruption
SVCERR	SVC error (issuing an SVC while locked, disabled, or in SRB mode)
PGIO	Paging I/O error
DAT	Dynamic address translation error
MACH	Software error caused by machine check
MEMTERM	Abnormal address space termination
ABEND	Task issued SVC 13
ALL	A short method of entering all of the above error conditions

Example:

```
ERRTYP=(MACH,DAT)
```

ID=xxxx

Specifies 1 to 4 alphameric characters as a trap identifier. If ID is not indicated in SLIP SET commands, the system assigns a unique id.

Example:

```
ID=PER1
```

{JOBNAME / J}=jobname

Specifies the 1 to 8 character name of the job, TSO id, or started task that must be in control for the JOBNAME match test to be satisfied.

For PER traps, JOBNAME limits PER monitoring to the address space in which the specified job runs. If neither JOBNAME nor ASID is specified for a non-IGNORE PER trap, PER monitoring is active in all address spaces in the system. If both JOBNAME and ASID are specified, one of the address space identifiers must be the one in which the job is running, otherwise the trap will not match.

Example:

```
JOBNAME=D10AXX1
```

{JSPGM / JS} =jobstep-program-name

Specifies the 1 to 8 character name of the job step program that must be in control to satisfy the JSPGM match test. On non-PER traps, JSPGM causes a no-match if the error type being processed is MEMTERM.

Example:

```
JSPGM=IFOX00
```

{LPAMOD / L} =(name[,start[,end]])

Name specifies the 1 to 8 character name of the link pack area module within which the error must occur. Start and end are offsets within the module, and if specified, the error must occur within that offset range. The starting offset must be less than or equal to the ending offset. If just a start offset is specified, the error must occur at that offset. If offsets are not specified, the error can occur anywhere within the LPA module. For an IF or SB PER trap, LPAMOD establishes the range of addresses to be monitored.

For an SA PER trap, indicates that the storage alteration must be caused by an instruction within the LPAMOD or within the module in the range of offsets specified by start and end.

Example:

```
LPAMOD=(IEAVTXYZ,2C)
```

MATCHLIM=m

Specifies that the SLIP trap is to be disabled after m matches, where m is an integer from 1 to 65535.

Example:

```
MATCHLIM=50
```

{MODE / M} =(mode[,mode...], [ANY / EVERY] [mode...])

Specifies the mode the system must be in for the trap to match. You can specify more than one mode and you can indicate that any of them (ANY) will be sufficient to cause a match or you can indicate that every one specified (EVERY) must be present. The modes you can specify are:

DIS	Physically disabled for I/O and external interruptions
GLOC	Holding any global lock
GLOCSD	Holding a global suspend lock
GLOCSP	Holding a global spin lock
LLOC	Holding a local lock
LOCK	Holding any lock
PKEY	Problem program key (8 or more)
PP	Problem program state
RECV	Recovery routine in control
SKEY	System key (7 or less)
SRB	SRB mode
SUPER	Supervisor state
SUPR	Supervisor control mode (any bit set in PSASUPER)
TCB	TCB mode
TYP1	Type 1 SVC in control
ALL	All of above

Invalid Modes: The following mode specifications are invalid and will cause an error message to be issued:

MODE=(EVERY) – not possible because some modes are mutually exclusive. (ALL is assumed).

MODE=(ANY) – Must specify one or more modes to go with ANY.

MODE=(ANY,EVERY) – Contradictory.

The RECV mode cannot be chosen when MODE is specified on a PER trap.

Example:

```
MODE=( LLOC,SRB,EVERY )
```

PRCNTLIM=p

Specifies a software limit for PER processing by indicating a maximum percentage of the system that can be devoted to processing caused by PER interrupts. At least 33.55 seconds must have elapsed since the first PER interruption before a trap will be disabled because of this limit. The range of integers for p is 1 to 99. You should use caution in specifying 99 because it means that no percent limit checking is performed. The value computed to test PRCNTLIM is an approximation.

Example:

```
PRCNTLIM=20
```

{PVTMOD / P} =(name[,start[,end]])

For a non-PER trap, specifies the 1 to 8 character name of the private area load module within which the error must occur. If start and end offsets are specified, the error must occur within that offset range. If just a start offset is specified, the error must occur at that offset. If offsets are not specified, the error can occur anywhere within the private module. The starting offset must be less than or equal to the ending offset.

PVTMOD is invalid for an IF or SB PER trap.

For an SA PER trap, indicates that the storage alteration must be caused by an instruction within the private module or within the module in the range of offsets specified by start and end.

Example:

```
PVTMOD=(MYMOD,1C,1F)
```

{RANGE | RA} =(start[,end])

For PER traps, specifies the starting and ending hexadecimal addresses of the virtual storage area to be monitored. If only one address is specified, a one byte range is assumed. If the starting address is greater than the ending address, the addresses wrap around. Refer to *IBM System/370 Principles of Operation* for an explanation of storage addressing wrap around. RANGE is invalid for non-PER traps. RANGE cannot be specified on an ACTION=IGNORE storage alteration PER trap.

Example:

```
RANGE=(600,700)
```

{RBLEVEL | RB} = {ERROR | PREVIOUS | NOTSVRB}

Indicates which RB contains the registers for use in resolving indirect addresses and the PSW for use by LPAMOD, PVTMOD, ADDRESS, and MODE. This keyword applies only to unlocked task mode errors and is not valid for PER traps.

ERROR
The PSW used is obtained from the RB prior to the SVC 13 (ABEND) RB. The registers are obtained from the SVC 13 RB.

PREVIOUS
The RBs used are each one RB prior to the RBs used for ERROR.

NOTSVRB
The PSW is obtained from the most recent non-SVRB and the registers are obtained from the associated SVRB.

Example:

```
RBLEVEL=NOTSVRB
```

Example 1: This example establishes an enabled SLIP trap with an ID of 0002. It requests an SVC dump (by default) if there is an 0C4 program check interruption while module MOD01 and job JOBXYZ are in control.

```
SLIP SET,ENABLE,ID=0002,COMP=0C4,ERRTYP=PROG,JOBNAME=JOBXYZ,
LPAMOD=MOD01,END
```

Example 2: The following command sets an instruction fetch PER trap that will cause an SVC dump when the instruction at CD3100 is executed. PER monitoring will be active in all address spaces in the system because neither ASID nor JOBNAME was specified.

```
SL SET,IF,ENABLE,ACTION=SVCD,RANGE=CD3100,END
```

Example 3: This example sets up a trap for a successful branch trace of the path taken through the LPA module MOD01 starting at offset 108 (hex) through 4FC during the execution of JOBX. After 20 standard SLIP records have been written the trap is automatically disabled.

```
SLIP SET,SB,ENABLE,ID=PER1,LPAMOD=(MOD01,108,4FC),JOBNAME=JOBX,
ACTION=TRACE,MATCHLIM=20,END
```

Example 4: The following command defines a trap that will cause an SVC dump when storage location CD3010 is altered. MATCHLIM is 1 and PRCNTLIM is 10 by default.

```
SL SET,SA,ENABLE,ACTION=SVCD,RANGE=CD3010,END
```

Modifying a SLIP Trap

Use the following form of the SLIP command to modify an existing SLIP trap.

```
{SLIP}  MOD {,ENABLE}  {,ALL      }
{SL  }      {,EN    }  {,ID=xxxx  }
            {,DISABLE}
            {,D     }
```

MOD
This SLIP command modifies one or all existing SLIP traps.

ENABLE or EN
The specified SLIP trap is to be made active.

DISABLE or D
The specified SLIP trap is to be made inactive.

ALL
Every SLIP trap present in the system is to be modified.

ID=XXXX
Only the SLIP trap associated with the identifier XXXX is to be modified.

Note: To modify more than one, but not all SLIP traps, you must enter a separate SLIP MOD command for each ID.

Example 1: The following SLIP command is used to deactivate the SLIP trap associated with identifier 0024. This SLIP trap can be activated again with the SLIP MOD,ENABLE,ID=0024 command.

```
SLIP MOD,DISABLE,ID=0024
```

Example 2: This example sets up two SLIP traps, both initially disabled, and then enables them both with a SLIP MOD command. The second trap prevents the first trap from matching for the subset of instructions specified by LPAMOD.

```
SLIP SET,IF,DISABLE,LPAMOD=(MODX,110,,FB),JOBNAME=JOB1,MATCHLIM=500,
ACTION=TRACE,TRDATA=(STD,REGS),END
SLIP SET,IF,DISABLE,LPAMOD=(MODX,1C4,1D7),ACTION=IGNORE,END
SLIP MOD,EN,ALL
```

Note: The IGNORE trap is specified after the non-IGNORE trap because traps are processed for match tests in last-in-first-out order.

Note: Unless you know that you are the only user of SLIP in the system, do not issue

```
SLIP  MOD,ENABLE,ALL
```

You might enable someone else's disabled traps. To find out what SLIP traps are in the system, issue

```
DISPLAY  SLIP
```

Deleting a SLIP Trap

Use the following form of the SLIP command to delete a SLIP trap.

{SLIP / SL}	DEL {,ALL / ,ID=xxxx}

DEL
This SLIP command removes one or all SLIP traps from the system.

ALL
Every SLIP trap in the system is to be deleted.

ID=XXXX
Only the SLIP trap associated with the identifier XXXX is to be deleted.

Note: To delete more than one, but not all SLIP traps, you must enter a separate SLIP DEL command for each ID.

Example: The following SLIP command is used to delete the SLIP trap associated with identifier 0008. This SLIP trap cannot be reactivated by the SLIP MOD command.

```
SLIP DEL,ID=0008
```

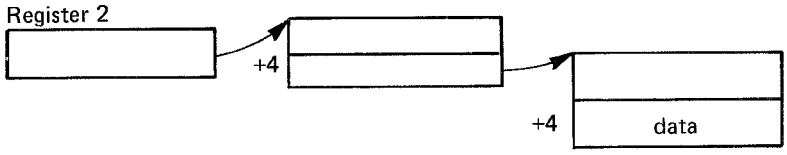
Note: Unless you know that you are the only user of SLIP in the system, do not issue SLIP DEL,ALL.

Base/Displacement Shorthand

Instead of entering the complete direct or indirect address, you can use a form of shorthand for any address after the first. The first address establishes a base address and is written out completely. Subsequent addresses are written as plus or minus displacements from the base.

```
2R%+4%+4
base   displacement
```

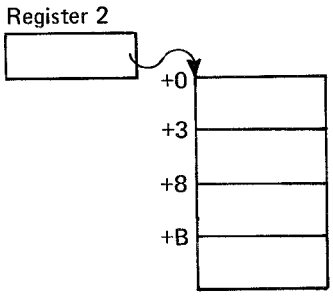
Graphically:



The base consists of everything in the indirect address except the last displacement. (If no displacement is specified, SLIP assumes a displacement of zero.)

```
Short Method                    Regular Method
2R%,+3,+8,+B                    2R%+0,2R%+3,2R%+8,2R%+B
base
                 displacement 3
            displacement 2
      displacement 1
```

Graphically:



Indirect Addressing and the DATA Parameter

The following discussion applies to the DATA parameter when you specify multiple targets and use shorthand.

The first direct or indirect address is written out completely and establishes the base. Subsequent addresses are written as plus or minus displacements from the base.

For example,

```
2R%+4,EQ,A24,2R%+8,NE,B66
target operator value  target operator value
```

is written using shorthand form as

```
2R%+4,EQ,A24,+8,NE,B66
target operator value target operator value
```

You must establish a direct or indirect base address before using displacements. In the following example, 2R specifies the contents of general purpose register 2, not an address. Therefore, the example using a +6 displacement is incorrect.

```
2R,EQ,C12,+6,NE,D01
```

Indirect Addressing and the LIST, SUMLIST, and TRDATA Parameters

When indirect addressing is used with the LIST, SUMLIST, or TRDATA parameters, the indirect addresses point to the beginning and end of a field of data.

The following example shows a starting and ending address:

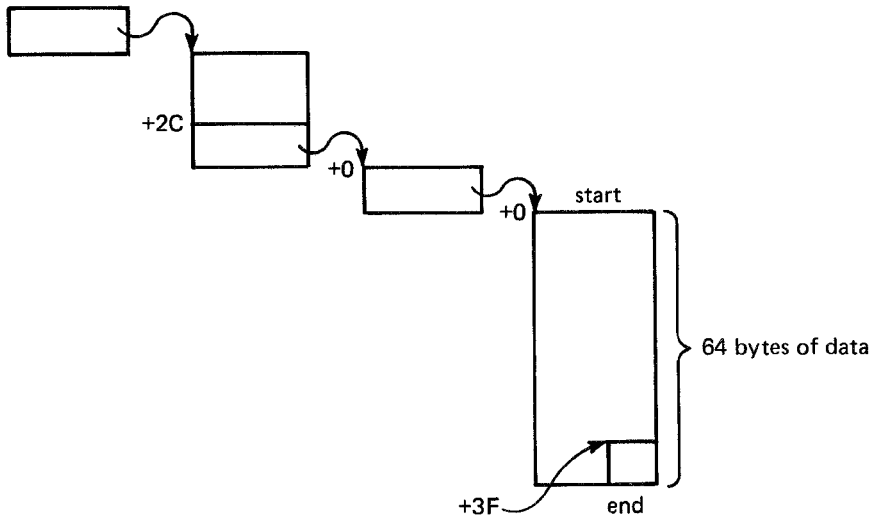
```
10%+2C%%,10%+2C%%+3F
 start      end
```

The same example in shorthand is:

```
10%+2C%%,+3F
 start end
```

Graphically:

Location 10 (hexadecimal)



START Command

The START command is used to:

- Start jobs from the console. The START command invokes a cataloged procedure residing in the procedure library. The cataloged procedure contains the JCL for the job.
- Start a reader or writer
- Start the generalized trace facility (GTF)
- Start the MF/1 recording facility
- Start TSO/VTAM

The complete syntax of the START command is:

```
{START}
{  S  }  {procname[.identifer][,keyword=option[,keyword=option]...]}
         {
         {procname[.identifier], [unitaddr ]
         {                       [devicetype]
         {  ,[volumeserial],[classes]
         {[,keyword=option[,keyword=option]...]
         {
         {procname[.identifier]
         {[,devicename][,volumeserial]
         {
         {[,([MEMBER= {name} ][,USERMAX=nnnnn])]
         {            {nn  }
         {
         {[,keyword=option[,keyword=option]...]
         {
         {{GTF      } [.identifier]
         {{procname}
         {[,[devicetype],[volumeserial]]
         {
         {[,([MODE= {INT} ][,BUF=nnn][,TIME=YES][,DEBUG=YES])]
         {           {EXT}
         {
         {[,MEMBER=xxxxxxxx][,REGION=nnnnK]
         {[,keyword=option[,keyword=option]...]
         {
         {{MF]      } [.identifier],[devicename],
         {{procname}
         {
         {     [volumeserial],[parmvalue]
         {     [,keyword=option[,keyword=option]...]
```

Starting a Job From the Console

Use the following form of the START command to start a job from the console.

```
{START}  procname[.identifier][,keyword=option [,keyword=option]...]
{  S  }
```

procname
The name of a cataloged procedure, residing in a procedure library, that defines the job to be started.

identifier
The user-determined name identifying the job to be started. This name can be up to eight characters long. The first character must be alphabetical.

keyword = option
Any appropriate keyword specified to override the corresponding parameter in the cataloged procedure.

Note: If option is specified within apostrophes, for example, 'DUMMY', the characters within the apostrophes must be in uppercase. Otherwise, the job abends with a JCL error.

Example:

```
s scratch
```

The job named SCRATCH is started.

Reading a Job From Direct Access or Tape

The START command can be used to read a JES2 job or a JES3 internal reader job from a magnetic tape or direct access device. The cataloged procedure named procname must be an internal reader procedure. For a description of the internal reader and how to catalog an internal reader procedure, refer to the subsection on the internal reader in the appropriate system programming library publication for JES2 or JES3.

Starting a Writer

Use the following form of the START command to start an external writer.

```
{START}   procname[.identifier],[unitaddr]
{  S  }    ,[volumeserial],[classes]
           [,keyword=option[,keyword=option]...]
```

procname
The name of a cataloged procedure, residing in SYS1.PROCLIB, that defines the writer to be started. The name can be either the IBM-assigned name, for an XWTR, or a user-assigned name.

identifier
The identifier of the writer to be started. This name consists of up to eight characters. The first one must be alphabetical. If you do not assign an identifier, the system uses the unit address of the device allocated to the writer as the identifier.

unitaddr or devicetype
The unit address, such as 280, or the device type, such as 2400, of the writer to be started.

volumeserial
The serial number, up to six characters, of the magnetic tape or direct access volume the writer is to use.

classes
The output classes, in priority sequence, the writer is to process. You can specify up to eight output classes, naming them in sequence without separating commas.

keyword = option
Any appropriate keyword specified to override the corresponding keyword in the cataloged procedure. The maximum length of each keyword=option is 66 characters. UNIT= cannot be used if the unitaddr or devicetype positional parameter was used; VOL= cannot be used if the volume serial positional parameter was used.

The commas preceding devicetype, volumeserial, and classes can be omitted when the positional parameters following them are not entered.

Example:

```
S WTR.A,282
```

An external writer, with the identifier A, is started to device 282.

Starting GTF

To start the generalized trace facility, use the following form of the START command and the following parameters.

```
{START}  {GTF     }
{  S  }  {procname} [.identifier]
          [,[devicetype],[volumeserial]]
          [,([MODE={INT}] [,BUF=nnn] [,TIME=YES] [,DEBUG=YES])]
                   {EXT}
          [,MEMBER=xxxxxxxx] [,REGION=nnnnK]
          [,keyword=option[,keyword=option]...]
```

GTF
The name of the IBM-supplied cataloged procedure that invokes GTF.

procname
The name of the user-written cataloged procedure that invokes GTF.

identifier
The user-determined name identifying this specific GTF session.

devicetype
An input/output device such as a tape or DASD device. The device type provided in the IBM-supplied cataloged procedure for GTF is to be used unless overridden by the START command.

volumeserial
The serial number of a magnetic tape or direct access volume.

MODE=INT
Trace data is to be maintained in GTF address space.

MODE=EXT
Trace data is to be maintained in an external data set.

BUF=nnn
The number (10-255) of buffers to be formatted if an abnormal termination occurs or if a dump is taken.

TIME=YES
Each logical trace record is to be time stamped. If this parameter is omitted, individual records are not to be time stamped.

DEBUG=YES
GTF terminates whenever an error is encountered while a trace record is being created. If this parameter is not specified GTF attempts to recover from the error condition but might not be able to record future events of the same type.

MEMBER=xxxxxxxx
The member of SYS1.PARMLIB to be accessed by this invocation of GTF. If this parameter is omitted, GTFPARM is used.

REGION=nnnnK
The maximum size (832 to 2880) of the GTF address space. K must be specified.

keyword = option
Any appropriate keyword specified to override the corresponding keyword in the cataloged procedure.

Examples:

```
S GTF.EXAMPLE,,,(MODE=INT),zyx=NULLFILE
```

This example starts a GTF session with the identifier EXAMPLE and with trace data maintained in the GTF address space. The zyx keyword is entered to override a symbolic parameter defined in the cataloged procedure.

```
s gtf,282,,(TIME=YES,BUF=20),LABEL=(,NL)
```

This example starts a GTF session with trace data recorded on the non-labeled tape on device 282. Each trace record is time stamped. Twenty buffers are formatted if a dump is taken.

```
s gtf,,,(mode=ext),region=1000k
```

This example starts a GTF session with the trace data recorded on an external device. Since it is not apparent which is the GTF recording device, you have to display active jobs with the D A,LIST command before you can stop GTF. The GTF session started in this example could run in an address space of a maximum of one megabyte.

Starting System Activity Measurement Facility (MF/1)

MF/1 is used to gather measurement data, write SMF records and to print measurement reports. Use the following form of the START command to start MF/1.

```
{START}  {MF1     } [.identifier],[devicename],
{  S  }  {procname}
                 [volumeserial],[parmvalue]
                 [,keyword=option[,keyword=option]...]
```

Note: The system activity measurement facility (MF/1) is not supported with MVS System Extensions (program number 5740-XE1).

MF1
The name of the IBM-supplied cataloged procedure that invokes MF/1.

procname
The name of the user-supplied procedure that invokes MF/1.

identifier
The user-determined name identifying this specific MF/1 session.

devicename
An input/output device. The devicename provided in the IEFRDER DD statement in the IBM-supplied cataloged procedure for MF/1 is to be used if this parameter is omitted.

volumeserial
The serial number of a magnetic tape or direct access volume.

parmvalue
The options desired for this specific MF/1 session. They can appear in any order and can be separated by commas, blanks, or comments. If you use a non-alphameric character, as when separating the parameters with commas, the parmvalues must be enclosed in parentheses. All options available are listed here for your reference and are described in greater detail in *OS/VS2 System Programming Library: Initialization and Tuning Guide.*

keyword = option
This parameter is only used to specify any special keyword syntax desired for the IEFRDER DD statement, or to specify any symbolic parameter keyword defined when a user-written cataloged procedure replaces the IBM-supplied one.

The MF/1 keywords and options include:

{ CHAN / NOCHAN }	Specifies whether or not system channel activity is to be monitored by MF/1.
{ CPU / NOCPU }	Specifies whether or not system processor activity is to be monitored by MF/1.
CYCLE	Specifies the frequency at which sampling observations are made of channel and device data.
{ DEVICE (list) / NODEVICE }	Specifies whether or not system device activity is to be monitored by MF/1. If DEVICE is specified, a device list must indicate the classes of devices that are monitored.
{ CHRDR / NOCHRDR }	A device list choice of character reader devices.
{ COMM / NOCOMM }	A device list choice of communications equipment.
{ DASD / NODASD }	A device list choice of direct access storage devices.
{ GRAPH / NOGRAPH }	A device list choice of graphic devices.
{ TAPE / NOTAPE }	A device list choice of magnetic tape devices.
{ UNITR / NOUNITR }	A device list choice of unit record devices.
INTERVAL { value / value M }	Specifies the interval at which all data is to be gathered for report formatting and/or SMF record writing.

MEMBER (nn)	The value specified by this parameter is appended to IRBMF1 to form the name of the partitioned data set that contains the MF/1 options.
{OPTIONS or OPTN / NOOPTIONS or NOOPTN}	Specifies whether or not a list of the keyword options to be used is to be printed at the operator's console at MF/1 initialization.
{PAGING / NOPAGING}	Specifies whether or not the system paging activity is to be monitored by MF/1.
{REPORT (REALTIME / DEFER) / NOREPORT}	Specifies whether or not printed reports of the monitored data are to be produced.
{STOP (value / value M / value H) / NOSTOP}	Specifies the desired time duration of MF/1 activity in minutes or hours.
SYSOUT (class)	Specifies the SYSOUT class to which formatted reports are directed.
{RECORD / NORECORD}	Specifies whether or not the monitored data is to be written to the SMF data set.
{WKLD (PERIOD / GROUP / SYSTEM) / NOWKLD}	Specifies whether or not system workload activity is to be monitored by MF/1.

Example:

```
s mf1.example,,,(wkld(system),cycle(100),device(nocomm))
```

This example initiates an MF/1 session with the identifier EXAMPLE and the WKLD, CYCLE, and DEVICE options specified to override the default options. It is advantageous to specify an identifier so that, if necessary, you can cancel the MF/1 session.

Starting TSO/VTAM Time Sharing

Use the following form of the START command to start TSO once VTAM is active. This command creates the terminal control address space (TCAS), which accepts requests by terminal users for logon to TSO.

```
{START}
{  S  }   procname[.identifier]
          [,devicename][,volumeserial]
          [,([MEMBER={name}][,USERMAX=nnnnn])]
                     {nn  }
          [,keyword=option[,keyword=option]...]
```

procname
The name of the cataloged procedure that starts TSO/VTAM time sharing.

identifier
The user-determined name identifying this specific time-sharing session.

devicename
The unit address or the device type of an output device where time-sharing parameters from SYS1.PARMLIB are listed. This operand is useful only if the PRINTOUT DD statement of the cataloged procedure invoked by the START command specifies DDNAME=IEFRDER. If the PRINTOUT DD statement specifies a SYSOUT device, the parameter values are listed on that device.

volumeserial
If devicename is a tape or direct access device, the volume serial number of the volume mounted on the device.

MEMBER =

name
The name of the member in SYS1.PARMLIB that contains TSO/VTAM time-sharing parameters.

nn
A two-digit decimal number that forms the suffix for specifying the name of the member in SYS1.PARMLIB that contains TSO/VTAM time-sharing parameters, where the name has the form TSOKEYnn.

USERMAX=nnnnn
The maximum number (0 to 32,767) of users that can be logged on to TSO/VTAM time sharing at any one time. If USERMAX is not specified here or in the parmlib member that contains TSO/VTAM time-sharing parameters, a value of 40 is used.

keyword = option
Any appropriate keyword specified to override the corresponding keyword in the cataloged procedure.

If more than one SYS1.PARMLIB name is specified, or if no name is specified, the order of priorities that determines which time-sharing parameters are used is:

1. The member name coded on the PARMLIB DD statement.
2. The MEMBER operand of the START command.
3. The keyword operand of the START command.
4. The default member TSOKEY00 if a member is not specified but a SYS1.PARMLIB (that contains TSOKEY00) is.
5. The default values in the TCAS program if neither a member nor a SYS1.PARMLIB is specified.

STOP Command

The STOP command is used to stop:

- A job in execution
- A writer
- The generalized trace facility
- MF/1
- TSO/VTAM

The complete syntax of the STOP command is:

```
{STOP}
{   P}  ( jobname                          )
        ( [procname.]identifier            )
        ( identifier                       )
        ( {MF1.identifier        }         )
        ( {procname.identifier   }         )
        ( {identifier            }         )
        ( [procname.]identifier            )
```

jobname
The name of the job to be stopped.

Note: The STOP command stops a job only if the programmer has coded a stop routine in his program.

procname
The name of a cataloged procedure used in the START writer command, for example, XWTR.

identifier
The identifier of the writer to be stopped. You can also use the device address of the writer.

Note: If the writer is waiting, it stops immediately; if it is working, it stops after finishing the current data set.

identifier
The identifier or device type that was specified when GTF was started. If GTF was started in internal mode and no identifier was specified, GTF is the identifier. If you are uncertain of the identifier or devicetype, use the D A,LIST command to display active jobs.

MF/1
The MF/1 session is to be stopped.

procname
The user-supplied MF/1 session is to be stopped.

identifier
The MF/1 session started with this identifier is to be stopped. If an identifier was not used with the START command, MF/1 cannot be stopped with the STOP command.

procname
The name of the cataloged procedure that was used in the START command to start TSO/VTAM time sharing.

identifier
The identifier that was specified in the START command that started TSO/VTAM time sharing. (Note that if you did not use an identifier with the START command, the system automatically used the procname you supplied in the START command as the identifier.)

If you enter the STOP command and one or more terminals are still active, you are asked to respond to the following message:

```
* id IKT010D nn USERS ACTIVE - REPLY 'SIC' or 'FSTOP'
```

Reply 'SIC' to cancel the active users normally. This allows them to receive any messages queued for them, and it allows TSO/VTAM to perform its normal termination processing. Reply 'FSTOP' to force immediate cancelation of the active users. The users do not receive any messages queued for them, and TSO/VTAM does not peform its normal termination processing (that is, task resource manager processing is by-passed). This reply should be used only if 'SIC' is ineffective.

Examples:

```
p sysda
```

The job SYSDA terminates, provided it has been written to accept a STOP command. Otherwise, the command has no effect.

```
p XWTR.00E
```

00E stops after it processes the current data set.

```
p EXAMPLE
```

This command stops the GTF session started with an identifier of EXAMPLE.

```
p MF1.example or p example
```

Either of these commands can stop the MF/1 sessions started with an identifier or example.

STOPMN Command

The STOPMN command stops the continual display of job status, data set status, or time-sharing user session activity requested by the MONITOR command.

The complete syntax of the STOPMN command is:

```
{STOPMN}   {JOBNAMES}
{   PM }   {DSNAME  }
           {SPACE   }
           {STATUS  }
           {SESS    }
```

JOBNAMES
The jobname display specified in the MONITOR JOBNAMES command is to be stopped.

DSNAME
The display of nontemporary data set names specified in MONITOR DSNAME command is to be stopped.

SPACE
The display of available space on direct access volumes specified in the MONITOR SPACE command is to be stopped.

STATUS
The display of data set names, volume serial numbers, and status specified in the MONITOR STATUS command is to be stopped.

SESS
The display of time sharing user identifiers is to be stopped.

Example:

```
pm jobnames
```

The display of job name information, appearing when a job is initiated or terminated, is to be discontinued.

STOPTR Command

Use the STOPTR command to halt or reduce the information periodically displayed as a result of the TRACK command.

The complete syntax of the TRACK command is:

```
{STOPTR} {TS   } [,L={a  }]
{ PT   } {JOBS } [    {cc }]
         {J    }      {cca}
         {A    }
```

TS
The display of active time sharing users is to be terminated.

JOBS or J
The display of active jobs, MOUNT commands in execution, and active tasks is to be terminated.

A
A display of active jobs, active time sharing users, MOUNT commands in execution, and active tasks is to be discontinued.

L=a, cc, or cca
The display area (a), console identifier (cc), or both (cca) of the active MCS console where the requested display is to appear. If you omit this operand, the command affects the dynamic display on the console through which the command is entered unless routing instructions are in effect; see the MSGRT command in this chapter.

Note 1: TRACK A followed by STOPTR TS or STOPTR JOBS has the same effect as TRACK JOBS or TRACK TS.

Note 2: The command sequence TR A, L; PT TS; TR TS, results in a display of the counts of STARTS, MOUNTS, jobs, time sharing users, and a list of jobs (but not time sharing users).

Example:

```
pt a,l=22a
```

The periodic display of system status information directed to area A of console 22 is to be discontinued.

SWAP Command

The SWAP command is used to initiate an operator request for dynamic device reconfiguration (DDR) and to activate or deactivate system initiated DDR.

The complete syntax of the SWAP command is:

```
{SWAP}  {OFF      }
{ G  }  {ON       }
        {xxx,yyy  }
```

Operator-Requested DDR

Use the following form of the SWAP command when it is necessary to move a volume to another device, for example, when the device requires maintenance.

```
{SWAP}  xxx,yyy
{ G  }
```

xxx
The address of the device from which the volume is to be swapped.

yyy
The address of the device to which the volume is to be swapped.

xxx and yyy must be the same device type and have compatible features. Wait for system message IGF500D or IGF509D and reply with one of the following:

YES — the system is to proceed as indicated.
NO — the swap request is to be canceled.
zzz — an alternate "to" device is to be used.

When replying YES or zzz, wait for the following message before proceeding with the swap:

```
* id IGF502E PROCEED WITH SWAP OF xxx TO yyy
```

While the system never requests DDR for readers, printers, or punches, you can request a swap of these devices. Unit record devices must be in a not-ready state when you enter the SWAP command.

DDR does not manage all the swap of 3851 devices (MSC). Information on the switching from a failing 3851 to the backup 3851 is described in *Operator's Library: IBM 3850 Mass Storage System (MSS) Under OS/VS*.

System-Initiated DDR

If a permanent I/O error occurs on a DDR-supported device, and it is an error supported by DDR, the system requests that the volume be moved.

Use the following form of the SWAP command to activate or deactivate system-initiated DDR.

{SWAP} {G}	{OFF} {ON}

OFF

System-initiated DDR is to be deactivated. Any further swapping of devices must be done by operator-initiated DDR.

ON

System-initiated DDR is to be activated for any further swapping of devices.

You receive a message (IGF500D or IGF509D) that gives you the choice of using DDR, not using DDR, or specifying an alternate swap device. Reply one of the following:

YES — the system is to proceed as indicated.
NO — the swap is to be canceled and the error should be posted.
zzz — an alternate "to" device is to be used.

When replying YES or zzz, wait for the following message before proceeding with the swap:

```
* id IGF502E PROCEED WITH SWAP OF xxx TO yyy
```

Note 1: If you are using a 3348 Model 70F Data Module, make sure that the "to" 3340 device has the fixed-head feature installed.

Note 2: If a system-initiated DDR is in progress, the swap must be satisfied before system-initiated DDR can be deactivated. If you are unable to satisfy the requirements, it might be necessary to re-IPL.

SWITCH Command

Use the SWITCH command to manually switch the recording of SMF data from one data set to the other (SYS1.MANX and SYS.MANY).

The complete syntax of the SWITCH command is:

```
{SWITCH}
{   I  }        SMF
```

SMF
The recording of SMF data is transferred from one SMF data set to the other. All SMF data in storage is to be written out before the transfer is made.

TRACE Command

Use the TRACE command to keep system tracing active between the time the system starts JES2 or JES3 and the time you start the generalized trace facility (GTF).

The complete syntax of the TRACE command is:

TRACE	{ ON OFF STATUS }

ON
After subsystem initialization, system tracing continues and *cannot* be stopped by TRACE OFF. Whenever GTF is not active, system tracing is active. TRACE ON imposes a load on the system and could cause some performance degradation. TRACE ON is ignored unless it is issued before the system starts JES2 or JES3. Therefore, it is recommended that, if TRACE ON is required, the system programmer issues it by including it in the COMMNDxx member of SYS1.PARMLIB.

OFF
After subsystem initialization, system tracing terminates. This operand is used to revoke TRACE ON before subsystem initialization has been completed.

STATUS
The system displays the current status of system tracing. If the TRACE command is issued with no operand, STATUS is assumed.

TRACK Command

Use the TRACK command to request a periodic display of job information on display consoles. This command is rejected when a non-display console is specified. The information requested is displayed at 30-second intervals unless you specify a different time interval with the CONTROL command.

The complete syntax of the TRACK command is:

```
{TRACK} {TS   } [{,LIST}] [,L= {a  }]
{ TR  } {JOBS }  {,L   }       {cc }
        {J    }                {cca}
        {A    }
```

TS
The system is to periodically display the number of active time sharing users and the maximum number of TSO/VTAM users.

JOBS or J
The system is to periodically display the number of:

- Active batch jobs (MOUNT commands in execution are treated as active batch jobs).
- Started tasks
- Active initiators

A
The system is to periodically display the information listed for both TS and JOBS, above.

Note: If both TRACK JOBS and TRACK TS are issued, the result is the same as if TRACK A had been issued.

LIST or L
For TR JOBS or TR A, the system also periodically displays the jobnames and V=R region boundaries. For TR TS or TR A, the system also periodically displays the user identification number of each active terminal.

L=a, cc, or cca
The display area (a), console identifier (cc), or both (cca) of the active MCS console where the requested display is to appear. If you omit this operand, the display is presented in the first available display area or message area of the console through which you enter the command (unless routing instructions are in effect; see the MSGRT commmand in this chapter).

Example:

```
tr a
```

The system is to periodically display the information listed for both TS and JOBS, above. Since the L= operand is not specified, one of two situations must exist:

- The console issuing the command is a display console and is to receive the specified output.
- A MSGRT command has been previously entered specifying the routing defaults for TRACK command output.

UNLOAD Command

Use the UNLOAD command to remove tape or DASD volumes that are mounted on the system.

The complete syntax of the UNLOAD command is:

{UNLOAD / U}	unitaddr

unitaddr
The unit address of the input/output device containing the volume to be unloaded. The device to be unloaded cannot be a system resident or permanently resident volume. If the volume is in use, it is unloaded when the steps using it have terminated. If the volume is a tape volume, it is rewound and unloaded.

Note: The UNLOAD command is not recommended for use in a JES3 complex and must not be used for JES3-managed devices and volumes.

VARY Command

The VARY command is used to control the assignment and configuration of devices, channels, and processor(s).

Through the use of the VARY command, the operator has the capability to:

- Assign and control MCS (multiple console support) consoles.
- Change MCS assignments
  - –Change the master console
  - –Control the hardcopy log
- Place resources online or offline
  - –Change the status of a secondary console
  - –Place an IO device online or offline
  - –Place a range of I/O devices online or offline
  - –Place a path online or offline
  - –Place real storage online or offline
  - –Place a channel online or offline
  - –Place a processor online or offline in an MP (multiprocessing) or AP (attached processor) environment.

The complete syntax of the VARY command is:

```
{VARY}
{ V }

{ CN(consoleid[,consoleid]...) ,AUTH= {ALL                    }
                                       {INFO                   }
                                       {([SYS][,IO][,CONS])    }

  ( {unitaddress     } [,unitaddress     ]...)
    {(I-unit,O-unit)}  [,(I-unit,O-unit) ]
    {O-unit          }  [,O-unit          ]

      , {ONLINE                                          }
        {OFFLINE                                         }
        {CONSOLE  [,AUTH= {ALL                    }]     }
                          {INFO                   }
                          {([SYS][,IO][,CONS])    }
                  [,ROUT= {ALL                    }
                          {NONE                   }
                          {rtcode[,rtcode]...     }
                  [,ALTCONS= {unitaddress     }
                             {(I-unit,O-unit) }
                             {O-unit          }]

  {unitaddress     }
  {(I-unit,O-unit) }  ,MSTCONS

  [unitaddress] ,HARDCOPY [,NOCMDS]
  [SYSLOG     ]           [,INCMDS]
                          [,STCMDS]
                          [,CMDS  ]
                          [,OFF   ]

      [,ROUT= {ALL                    }]
              {NONE                   }
              {(rtcode[,rtcode]...)   }

  STOR ( {dddddk,dddddk} )      {,ONLINE  }
         {xxxxxx,xxxxxx}        {,OFFLINE }
         {ddM,ddM      }

  {CH(y,x)      }  {,ONLINE                 }
  {PATH(ddd,x)  }  {,OFFLINE[,UNCOND]       }
  {CPU(x)       }  {,ONLINE   [,KEEPCHAN]   }
                   {,OFFLINE  [,UNCOND  ]   }
```

Assigning and Controlling MCS Consoles

Use the VARY command to:

- Assign a device as an MCS console.
- Change the type of system commands a console is authorized to issue.
- Assign or change the system message routing codes sent to a console.
- Assign an alternate MCS console.

The devices specified in this command must have been designated as consoles during system generation and initialization.

Details of allocating JES3 consoles and changing JES3 console authority, JES3 message routing, and JES3 console processor association are provided in *Operator's Library: OS/VS2 MVS JES3 Commands*.

```
{VARY}  {CN(consoleid[,consoleid]...) ,AUTH={ALL                    }                                  }
{ V  }  {                                    {INFO                   }                                  }
                                             {([SYS][,IO][,CONS])    }                                  }
                                                                                                        }
        {(unitaddr        }[,unitaddr        ]...) ,CONSOLE[ ,AUTH={ALL                    }        ]  }
        {(O-unit          }[,O-unit          ]                     [ {INFO                   }        ]  }
        {(I-unit,O-unit)  }[,(I-unit,O-unit)  ]                     [ {([SYS][,IO][,CONS])    }        ]  }
                                                                    [ ,ROUT={ALL                  }    ]  }
                                                                    [        {NONE                 }    ]  }
                                                                    [        {(routecode[,routecode]...)}]  }
                                                                    [ ,ALTCONS={unitaddr          }    ]  }
                                                                    [          {O-unit             }    ]  }
                                                                    [          {(I-unit,O-unit)    }    ]  }
```

Note: A single unit address, a single AUTH=operand, or a single routecode need not be enclosed in parentheses. When CN is specified a single console must be enclosed in parentheses.

CN(consoleid[,consoleid]...)
Change the indicated authority for the specified consoles. When you are uncertain of the console identifiers in effect, enter a DISPLAY CONSOLES command; see "Displaying Console Configuration Information", under the DISPLAY command, earlier in this chapter, for further information on this command.

unitaddr
The address of an input/output console device.

O-unit
The address of a device with output capability.

(I-unit,O-unit)
The addresses of the input and output units that make up a composite console.

CONSOLE
The unit is to be an active console. The authorized commands, message routing, and alternate console remain the same as when the device was last active unless changed by the AUTH, ROUT, or ALTCONS operands.

AUTH=
The operator command groups that the console is authorized to enter. If AUTH is omitted, the command group assignment remains unchanged. This parameter is accepted only from the master console. Figure 2.3 lists the commands and their associated groups. (The use of command group numbers is explained in the description of the CONSOLE macro instruction *OS/VS2 System Programming Library: System Generation Reference*.

ALL
The specified console(s) are authorized to enter INFO, SYS, IO, and CONS commands.

INFO
The specified console(s) are authorized to enter only INFO commands.

(SYS, IO, CONS)
The specified console(s) are authorized to enter INFO commands as well as those other types specified. These parameters can be entered in any order. The commands associated with these command groups are listed in Figure 2.3.

ROUT=
The routing codes the console(s) are assigned to receive. These codes replace those previously assigned. The master console always receives routing code 1 messages, even when NONE is specified. See Figure 2.4 for a list of routing codes.

ALL
The console receives all system-to-operator messages.

NONE
The console does not receive any system-to-operator messages.

routecode
The console receives all messages with the specified routing codes.

ALTCONS=
The unit address of the alternate console. If this parameter is omitted, the alternate console is unchanged.

unitaddr
The console specified is to be assigned as an alternate to the console specified in the first operand.

O-unit
The output device specified is to be assigned as the alternate to the output-only console specified in the first operand.

(I-unit, O-unit)
The composite console specified is to be assigned as the alternate to the console specified in the first operand.

Examples:

```
v ((i-00c,o-00e),009),console,altcons=01f
```

The composite console, consisting of 00C and 00E, and the console 009 are to become active consoles with 01F as their alternate console.

```
V 01f,console,auth=(cons,sys),rout=(1,3,5)
```

01F is to become an active console that is authorized to enter informational, system control, and console control commands and that receives messages for the master console, the tape pool, and the tape library.

Changing MCS Console Assignments

Use the VARY command to alter an MCS console assignment. This command can be used to change the master console.

If you are uncertain of the console assignments currently in effect, enter the DISPLAY CONSOLES command (see "Displaying Console Configuration Information," under the DISPLAY command, earlier in this chapter).

Changing the Master Console

Use the following form of the VARY command to switch the master console to another device.

```
{VARY}  {unitaddr          }  ,MSTCONS
{ V  }  {(I-unit,O-unit)   }
```

unitaddr
The address of the device is to be assigned as the master console. This unit must have input and output capabilities.

(I-unit,O-unit)
The addresses of the input (I-unit) and output (O-unit) devices forming a composite console to be assigned as the master console.

MSTCONS
The master console is to be switched.

Example:

```
V (I-00C,O-00E),MSTCONS
```

The composite console consisting of devices 00C and 00E is assigned as the new master console.

Controlling the Hardcopy Log

Use the following form of the VARY command to:

- Assign a unit as the hardcopy log device
- Discontinue the hardcopy log function
- Change the routing of messages or commands to the hardcopy log

```
{VARY}  [unitaddr]  ,HARDCPY  [,NOCMDS                                  ]
{ V  }  [SYSLOG  ]            [,INCMDS                                  ]
                              [,STCMDS                                  ]
                              [,CMDS                                    ]
                              [{,OFF                                    }]
                              [{,ROUT= {ALL                            }}]
                              [{       {NONE                           }}]
                              [{       {(routecode[,routecode]...)     }}]
```

unitaddr
The specified device is to be the hardcopy log. unitaddr must be the address of an active, nondisplay console. If a composite console is to be used, unitaddr specifies the address of the output device.

SYSLOG
The system log is to be the hardcopy log device.

When the unitaddr and SYSLOG operands are omitted, the specified modifications are performed on the current hardcopy log device. When the system log is the hardcopy log device, you may want to refer to "Recording System Information" in chapter 1 for a discussion of scheduling system log output.

HARDCPY
The status of the hardcopy log is to be changed.

NOCMDS
No operator commands or responses are recorded on the hardcopy log. On systems with multiple consoles active or at least one active display console, a hardcopy log is mandatory, this operand is ignored and CMDS is assumed.

INCMDS
Operator commands and responses are written on the hardcopy log. Status displays are not included in this output.

STCMDS
Operator commands, responses, and status displays (except time-interval updated status displays) are to be written to the hardcopy log.

CMDS
All operator commands, responses, and status displays (static and time-interval updated displays) are to be written to the hardcopy log.

Note 1: JES2 commands entered from a console are hardcopied if CMDS or INCMDS is in effect. JES2 commands are skipped if they are entered through the input stream.

Note 2: VTAM messages can be routed by routing codes or command repsonse.

OFF
The hardcopy log is to be terminated. This operand is rejected when the hardcopy log is mandatory (see NOCMDS operand) or when JES3 is active. If specified, OFF must be the last parameter specified. OFF and ROUT are mutually exclusive.

ROUT=
The routing codes that the hardcopy log is assigned to receive.

ALL
All system and operator communications are to be routed to the hardcopy log.

NONE
No system or operator communications are to be routed to the hardcopy log. This operand is ignored by systems with graphic consoles and minimum route codes are assigned, as described under "routecode."

routecode
All communications with the specified routing codes are recorded on the hardcopy log. The minimum routing codes for systems with mandatory hardcopy log are 1,2,3,4,7,8 and 10. Refer to Figure 2.4 for a list of routing code definitions.

Examples:

```
v ,hardcpy,off
```

The hardcopy log function is discontinued, provided that it is not mandatory and that JES3 is not active.

```
v syslog,hardcpy
```

The hardcopy log recording to be done on the system log.

```
v 01f,hardcpy,nocmds,rout=none
```

If 01F is an active console, it becomes the hardcopy device. If 01F is the only system console and it is not a graphic device, operator and system communications are not recorded on the hardcopy log.

```
v 01F,hardcpy,rout=13
```

If the hardcopy log is not mandatory, 01F receives only those messages with a routing code of 13. However, if the hardcopy log is mandatory, messages with route codes 1,2,3,4,7,8,10 and 13 are directed to the hardcopy log.

Placing Resources Online and Offline

Use the VARY command to make a system resource available or unavailable for system use. The following forms of the VARY command can:

- Change the status of a secondary console
- Place a device (or a range of devices) online or offline
- Place a path (or paths) online or offline
- Place storage online or offline
- Place a channel (or a range of channels) online or offline
- Place a processor online or offline

Changing the Status of a Secondary Console

Use the following form of the VARY command to make a secondary console available or unavailable for system use. You cannot VARY the master console or the hardcopy log ONLINE or OFFLINE. These functions must be assigned to another console before the VARY ONLINE or VARY OFFLINE command is issued.

```
{VARY}  ({unitaddr          } [,unitaddr          ] ...){,OFFLINE}
{ V  }   {O-unit            } [,O-unit            ]     {,ONLINE }
         {(I-unit,O-unit)   } [,(I-unit,O-unit)   ]
```

unitaddr
The address of the console(s) to be modified.

O-unit
The address of an output device to be modified.

If you specify only one device address in unitaddr or O-unit, the parentheses are not required.

(I-unit,O-unit)
The addresses of the input and output devices that make up a composite console.

ONLINE
The device(s) is made available to the system.

OFFLINE
The specified device(s) is made unavailable for system use.

Example:

```
v (003,001),offline
```

Consoles 003 and 001 are no longer available for system use.

Placing an I/O Device or Range of I/O Devices Online or Offline

Use the VARY command to change the status of an input/output device or devices or range of input/output devices. VARY should be used with caution in a JES3 environment. A corresponding JES3 command, *VARY, must be issued to inform JES3 of the status of JES3-managed devices.

{VARY / V}	{unitaddr / (unitaddr[,unitaddr]...) / xxx-yyy / (xxx-yyy[,aaa-bbb]...)}	, {ONLINE / OFFLINE}

unitaddr
The unit address of a specific input/output device that is to be put online or offline. You do not need parentheses when specifying only one unit address.

xxx-yyy
The lower unit address (xxx) and the upper unit address (yyy) of a range of devices to be put online or offline.

aaa-bbb
The lower unit address (aaa) and the upper unit address (bbb) of a second range of devices to be put online or offline.

ONLINE
The specified devices or range of devices are made available for allocation to problem programs and system tasks if there is an online path to the device. If you change a device from offline to online status and you want the system to recognize a volume mounted while the device was offline, enter a MOUNT command for the device.

Notes:

1. If a device that is not physically attached to its control unit is specified, the system might consider the device as being operational and online. If an attempt is later made to allocate the device to a job, the attempt might fail, in which case you would have to cancel the job.
2. Varying a device online cannot bring online paths that have been varied offline with a VARY PATH commmand.
3. If you issue a VARY device ONLINE command for a device whose last path has been varied offline with a VARY PATH OFFLINE command, the system issues the following message:

```
IEE025I UNIT ddd HAS NO LOGICAL PATHS
```

OFFLINE
The specified devices or range of devices are no longer available for allocation to problem programs or system tasks. Devices currently in use are put offline when all tasks to which the devices are allocated have terminated.

Note: If any jobs are allocated to the specified device when the VARY OFFLINE command is issued, the device assumes a pending-offline status. The system does not allocate a device in pending-offline status to any other job unless the job specifically requests the volume that is associated with the device.

If the specified devices are not currently in use or scheduled for use, the VARY command does not take effect until a system task has started or terminated. If you are running a long job or waiting for work, enter a START command specifying the deallocation procedure, DEALLOC. This causes the system task to start and terminate, allowing the devices to be put offline.

When the specified devices are offline, you receive a system message and all tape drives specified are rewound and unloaded. All devices placed offline remain so until you enter another VARY command or specify them in response to a system request for devices.

Example 1: To make devices 282, 283, and 287 available for system use, enter:

```
vary (282,283,287),online
```

Example 2: To make any valid devices, that is, any device with a unit control block (UCB) assigned at system generation, in the range 283 through 287 and the range 130 through 135, unavailable for system use, enter:

```
vary (283-287,130-135),offline
```

In response to the command, one of the following occurs:

- If some or all of the devices are valid, you receive a status display of those devices in each range.
- If some or all of the devices are valid and are alternate path addresses, or do not have device names assigned to their UCBs, you receive message IEE712I, stating the VARY processing has completed.
- If any or all of the devices are not valid, you receive message IEE313I indicating the unit addresses that are invalid.

Note: This form of unit specification cannot be used with the CONSOLE operand. The range should never include any console addresses.

Placing a Path or Paths Online or Offline

Use the following form of the VARY command to make a path to a device available or unavailable to a uniprocessing or multiprocessing system. A path is the logical route between a processor and a device, consisting of a processor, a channel, and a control unit. A path can be unavailable if:

- One or more of the components is unavailable.
- A VARY PATH command has been entered.

Notes:

1. Paths taken offline with the VARY PATH command can be brought back online only with the VARY PATH command.
2. VARY PATH should be used with caution in a JES3 environment. A corresponding JES3 command, *VARY, must be issued to inform JES3 of the status of JES3-managed devices.

```
{VARY}   PATH {(unitaddr[,unitaddr]...[,z])} {ONLINE              }
{ V  }        {(xxx-yyy [,aaa-bbb ]...[,z])} {OFFLINE[,UNCOND]}
```

PATH
The specified path or paths are to become available or unavailable for system use. If the specified path is the last available path to an *allocated* device, offline processing for the device is bypassed.

Note: Single devices and device ranges can be entered together.

unitaddr
The unit address associated with the path to be changed.

xxx-yyy
The range of unit addresses - the lower unit address (xxx) and the upper unit address (yyy) associated with the paths to be changed.

aaa-bbb

The second range of unit addresses - the lower unit address (aaa) and the upper unit address (bbb) associated with the paths to be changed.

z
The processor identification (0 or 1) associated with the path to be changed. The processor identification can be omitted in a uniprocessing environment. If you are uncertain of the processor identifier, enter the DISPLAY M=CPU command to display it. In an attached processor system, you must specify z as 1. In this system, 0 refers to the attached processing unit (APU) and 1 refers to the processor itself; the APU has no associated path. For more information, refer to *OS/VS2 MVS Multiprocessing: An Introduction and Guide to Writing Operating and Recovery Procedures.*

ONLINE
The path is to be made available to the system.

OFFLINE
The path is to be made unavailable to the system.

UNCOND
The path is to be made unavailable to the system even though it is the last path to the specified device. This command is rejected if the specified device is allocated for system use.

Example:

```
v path(130,133-135,140,0),offline,uncond
```

The paths through processor 0, leading to devices 130,133,134,135, and 140 become unavailable for the system, even though there are no other paths to the devices available. A message describes the outcome for each device processed. For example, the path to device 130 can be taken offline successfully while processing for device 133 could be rejected if it is allocated to the system.

Placing Storage Online or Offline

Use the following form of the VARY command to make sections of real storage available or unavailable for system use. This command can be entered only from the master console, a reader with master console authority, or as an internal command.

```
{VARY}  STOR( {dddddK,dddddK} ), {ONLINE }
{ V  }        {xxxxxx,xxxxxx}    {OFFLINE}
              {ddM,ddM       }
```

STOR
A section of real storage is to be made available or unavailable for system use.

dddddK,dddddK
One to five decimal digits, followed by K, which are the starting and ending addresses of the section. Each address represents a multiple of 1024 bytes. The system rounds the lower address down to the next lower 4K and the high address up to the next higher 4K.

xxxxxx,xxxxxx
One to six hexadecimal digits that address the first and last bytes of the section. The system rounds the lower address down to the next lower 4K and the higher address (if not on a 4K boundary) up to the next higher 4K.

ddM,ddM
One or two decimal digits, followed by M, which are the starting and ending addresses of the section. Each address represents a multiple of one megabyte (1,048,576 bytes).

ONLINE
The specified real storage area is to be available for system use. When entering the VARY STOR ONLINE command, you cannot specify an address higher than that specified at system generation or initialization or storage that is not physically available to the system.

OFFLINE
The specified real storage area is to be unavailable for system use. This command is rejected if the storage being taken offline contains any portion of the control program's nucleus. There can be a delay between the time you enter this command and the time you receive a message indicating that the storage is offline because all activity in the specified storage area must stop before the command can take effect. There is also a delay if the storage is occupied by a program the system is unable to page out; the program must complete or be canceled before the storage can be freed and released in response to the VARY STOR OFFLINE command. Also, the command is rejected if the storage is permanently assigned to the control program. Generally, preferred storage (non-reconfigurable) cannot be taken offline while reconfigurable storage can be taken offline. The *OS/VS2 System Programming Library: Supervisor* publication describes some situations that can prevent storage from being taken offline and tells how to correct them using the program properties table (PPT).

Example:

```
v stor(1024K,2048K),online
v stor(100000,200000),online
v stor(1M,2M),online
```

The above commands make the same section of real storage, location 1,048,576 through location 2,097,151, available to the system.

Placing a Channel Online or Offline

Use the following form of the VARY command to make a channel available or unavailable for system use.

```
{VARY}  | CH {(x)    } , {ONLINE                 }
{ V  }  |    {(x,y)  }   {OFFLINE[,UNCOND]       }
```

CH
The status of a channel is to be changed.

x
The number (0-F) of the channel to be placed online or offline.

y
The processor identification (0 or 1) of the processor associated with the channel to be placed online or offline. y is not required for a uniprocessing system.

Note: In an Attached Processor System, you must specify y as 1. In this system 0 refers to the attached processing unit (APU) and 1 refers to the processor itself; the APU has no associated channel.

ONLINE
The channel is to be available for system use.

OFFLINE
The channel is to be unavailable for system use. This command is rejected if it affects the console when another active console is not available.

UNCOND

The channel is to be unavailable for system use even though it is the last path to a device. The command is rejected if an allocated device on that channel is not available through another channel path.

If you remove the master console with a V CH(x,y),OFFLINE command, a later V CH(x,y),ONLINE command for the same channel places the device online. However, the alternate console assigned when the master console became unavailable remains the master console. The previous master console does not become the alternate console unless a VARY CONSOLE command is entered. When you enter a V CH(x,y),OFFLINE command, the devices currently available through that channel also become unavailable to the system unless an alternate path exists. When you enter a V CH(x,y),ONLINE command, the devices on that channel also become available to the system unless a V xxx,OFFLINE command was issued while the channel was unavailable.

Example:

```
v ch(0),offline,uncond
```

Channel 0 is to be unavailable to the uniprocessing system issuing the above command even though it might be the last path to an unallocated device.

The channel reconfiguration hardware (CRH) of the Model 168 MP can be activated by varying a channel attached to an offline processor online. This first VARY command activates CRH and subsequent VARY commands of channels attached to offline processors permit the use of those channels through CRH. The CRH facility is deactivated when the last channel on the offline processor has been varied offline. To permit CRH operation, the MP/UP mode switch on the configuration control panel must be set to MP mode, and the offline processor must be powered up.

Placing a Processor Online or Offline

Use the following form of the VARY command to make a processor available or unavailable in a multiprocessing system. Refer to the section "Tightly Coupled Multiprocessing Guidelines" in Chapter 1 for further information on reconfiguring the system.

```
{VARY}  CPU(n),{ONLINE                          }
{  V }         {OFFLINE[{,UNCOND  }]}
               {       [{,KEEPCHAN}]}
```

CPU(n)

The processor (0 or 1) to be made available or unavailable for use.

Note: In an attached processor system, you must specify processor (0). In this system, processor (0) refers to the attached processing unit (APU) and processor (1) refers to the processor itself; VARY CPU(1) does not work because it would remove the last path to system allocated devices.

ONLINE

The specified processor is to be available for system use. All operational channels and paths utilizing the new processor are brought online. Any devices taken offline because of a previous VARY CPU or CHANNEL offline command are brought online. The TOD clock is synchronized if required, and the CRH facility is deactivated if active.

OFFLINE

The specified processor is to be unavailable for use. This command is rejected if the processor controls the last path to any device and neither UNCOND nor KEEPCHAN is specified.

UNCOND

The processor is to be placed offline even though it makes unavailable the last path to an unallocated device. If the device has been allocated for system use, the command is rejected. All I/O devices attached to the specified processor are placed offline.

KEEPCHAN

The processor is to be placed offline but with continual access to its devices. This function is supported only on the 168MP with (CRH) facility active. The channel reconfiguration hardware (CRH) on the 168MP is activated by the KEEPCHAN keyword. The KEEPCHAN keyword is mutually exclusive with the UNCOND keyword. When KEEPCHAN is specified, devices do not go offline with the processor.

Example:

```
v cpu(1),offline,uncond
```

Processor (1) is made unavailable for use even though, as a result of this command, certain unallocated I/O devices are made unavailable for use. If there are any jobs scheduled requiring the specified processor, the system does not allow the processor to be put offline and issues a message giving you the option of restricting the scheduling of any new jobs that require that processor.

WRITELOG Command

The WRITELOG command is used to control the system log. Through the use of the WRITELOG command, you can start, stop, or print the system log, or modify the output class of the system log.

The complete syntax of the WRITELOG command is:

```
{WRITELOG}  [class]
{   W    }  [CLOSE]
            [START]
```

WRITELOG by itself schedules the printing of the system log.

class
The one-character output class (A-Z, 0-9) to be used when printing the contents of the system log. This command is in effect only for the current scheduling of the system log output. All subsequent scheduling is to the default output class unless the class parameter is again entered.

CLOSE
The system log is closed and the log function is discontinued. This command is rejected if the system log is the hardcopy log and no other hardcopy device is available.

START
The system log is to be restarted.

Example:

```
writelog d
```

This command would result in the system log being scheduled to the class D output queue.

Appendix A: System Command Summary Index

Command (Abbr)	Function	Page	Acceptable From (Note 1)		Command Group
			Job Stream (Note 2)	TS Terminal	
ASSIGN	Assign primary host for MSS multi-host environment	–			(Note 3)
CANCEL (C)	Canceling a MOUNT command	81	X		SYS
	Canceling a job in execution	81	X		
	Canceling an external writer allocation	81	X		
	Canceling the writing of a SYSOUT data set by an external writer	81	X		
	Canceling a time sharing terminal session	81	X	X	
CHNGDUMP (CD)	Override dump options specified in SYS1.PARMLIB, on the ABEND, CALLRTM and SETRP macros and in the SDUMP parameter list	83	X		SYS
CONTROL (K)	Halt printing of a status display on a printer or typewriter that is not the hardcopy device	93			INFO (Note 4)
DISPLAY (D)	Display console configuration information	102	X		(Note 4)
	Display CONTROL command functions	103	X		(Note 4)
	Display configuration information	101	X		(Note 4)
	Display device allocation	100	X		(Note 5)
	Display current system status	103	X	X	INFO
	Display system information request	104	X	X	
	Display local time and date	105	X	X	
	Display domain descriptor table	105			
	Display effective dump options	106	X		

Notes:

1. The command is acceptable from the job stream or a time sharing terminal as indicated by an X. If neither job stream nor TS terminal is indicated, the command may be entered only from an operator console.
2. If you enter a system command through a card reader in a JES2 environment and the command is placed between jobs, enter $VS, 'sys command'; if the command is placed within a job, enter //ϕ . In a JES3 environment enter //**T,sysname, sys command.
3. For information concerning MSS commands, see *Operator's Library: IBM 3850 Mass Storage System (MSS) Under OS/VS.*
4. CONS command group when message routine is specified.
5. For information concerning the VTAM commands, see *Operator's Library: VTAM Network Operating Procedures.*

Command (Abbr)	Function	Page	Acceptable From (Note 1)		Command Group
			Job Stream (Note 2)	TS Terminal	
DISPLAY (D) (cont.)	Display SLIP definitions	106	X		(Note 3)
	Display commands associated with PFK keys	104			
DUMP	Request a dump of virtual storage to be stored in SYS1.DUMP data set	107			MC
FORCE	Forcing termination of:				MC
	• A MOUNT command	109			
	• A job in execution	109			
	• An external writer allocation	109			
	• The writing of a SYSOUT data set by an external writer	109			
	• A time-sharing terminal session	109			
HALT (Z)	Halt the system while preserving statistics and data records. (Must stop subsystem (JES2 or JES3) processing through use of a subsystem command first)	111			SYS (Notes 4 and 5)
HOLD (H)	Suspend transmission to a station	–	X		(Note 6)
LOG (L)	Enter comments into the system log	113	X		INFO
MODE	Control recording of system recovery and degradation machine check interrupts on the SYS1.LOGREC data set Control the enabling of the high speed buffer and TLB Control the monitoring of hard machine checks	115			SYS
MODIFY (F)	Change characteristics of a job through modification of the job parameters	121	X		SYS
	Specify criteria used by external writer in data set selection for processing	124	X		SYS
	Cause external writer to pause for operator intervention	125	X		SYS
	Start time sharing once TCAM is active	122	X		SYS
	Stop time sharing	122	X		SYS
	Modify TSO/VTAM time sharing	123	X		SYS

Notes:

1. The command is acceptable from the job stream or a time sharing terminal as indicated by an X. If neither job stream nor TS terminal is indicated, the command may be entered only from an operator console.
2. If you enter a system command through a card reader in a JES2 environment and the command is placed between jobs, enter $VS, 'sys command'; if the command is placed within a job, enter //ϕ. In a JES3 environment enter //**T,sysname, sys command.
3. For information on using SLIP, see *OS/VS2 System Programming Library: Supervisor.*
4. For information concerning MSS commands, see *Operator's Library: IBM 3850 Mass Storage System (MSS) Under OS/VS.*
5. For information concerning VTAM commands, see *Operator's Library: VTAM Network Operating Procedures.*
6. For information concerning TCAM commands, see *OS/VS TCAM.*

Command (Abbr)	Function	Page	Acceptable From (Note 1)		Command Group
			Job Stream (Note 2)	TS Terminal	
MONITOR (MN)	Continuously display data set status	127	X	X	INFO
	Continuously display job status		X	X	INFO
	Monitor time sharing terminal users logging on and off the system		X	X	
MOUNT (M)	Mount volumes	129	X		I/O
MSGRT (MR)	Establish message routing defaults for certain options of DISPLAY, TRACK, STOPTR, and CONTROL commands	131			INFO
	Stop message routing	132			
PAGEADD (PA)	Adding PAGE or SWAP data sets	133			I/O
PURGE	Demount all 3330V volumes from specified host	–			(Note 3)
QUIESCE	Put system in MANUAL state without affecting step timing	135			MC
RELEASE (A)	Remove station from interrupted status	–	X		(Note 4)
REPLY (R)	Reply to system information requests	137	X		INFO
	Specifying system parameters	138			
	Setting the time-of-day clock and specifying the installation performance specifications	139			
	Specifying SMF options	140			
	Specifying DUMP options	141			
RESET (E)	Change performance group of a job currently in execution	145	X		SYS
SEND (SE)	Communicate with other operators	147	X	X	INFO
	Communicate with specific time sharing terminal users	148	X	X	(Note 5)
	Communicate with all time sharing users	149	X	X	

Notes:

1. The command is acceptable from the job stream or a time sharing terminal as indicated by an X. If neither job stream nor TS terminal is indicated, the command may be entered only from an operator console.
2. If you enter a system command through a card reader in a JES2 environment and the command is placed between jobs, enter $VS, 'sys command'; if the command is placed within a job, enter //ϕ. In a JES3 environment enter //**T,sysname, sys command.
3. For information concerning MSS commands, see *Operator's Library: IBM 3850 Mass Storage System (MSS) Under OS/VS.*
4. For information concerning TCAM commands, see *Operator's Library: OS/VS TCAM.*
5. See *OS/VS2 System Programming Library: TSO.* Refer to the OPERATOR command and SEND subcommand of the OPERATOR command for further detail.

Command (Abbr)	Function	Page	Acceptable From (Note 1)		Command Group
			Job Stream (Note 2)	TS Terminal	
SEND (SE) (cont.)	Save messages in the broadcast data set for issuance at logon time or when requested	149	X	X	
	List messages accumulated in the notice section of the broadcast data set	150	X	X	
	Delete a message from the notice section of the broadcast data set	151	X	X	
SET (T)	Respecify parameters used by the system resources manager to control distribution of service to jobs	153	X		SYS
	Change local time and date	153	X		SYS
SETDMN (SD)	Respecify domain parameters	155	X		SYS
SLIP (SL)	Set SLIP definitions	157	X		SYS (Note 3)
	Modify SLIP definitions	163	X		
	Delete SLIP definitions	163	X		
START (S)	Start a job from a console	165	X		
	Start an external writer	166			
	Start the system activity measurement facility (MF/1)	168	X		
	Start the generalized trace facility (GTF)	167	X		
	Start TSO/VTAM time sharing	170	X		SYS
STOP (P)	Stop a job in execution	173	X		
	Stop an external writer	173	X		
	Stop a system activity measurement facility (MF/1) session	173	X		
	Stop the generalized trace facility (GTF)	173	X		
	Stop TSO/VTAM time sharing	173	X		SYS
STOPMN (PM)	Stop continual display of data set status	175	X	X	INFO
	Stop continual display of job status	175	X	X	INFO
	Stop monitoring time sharing terminal activity	175	X	X	
STOPTR (PT)	Halt or reduce information displayed using TRACK command	177			INFO

Notes:

1. The command is acceptable from the job stream or a time sharing terminal as indicated by an X. If neither job stream nor TS terminal is indicated, the command may be entered only from an operator console.
2. If you enter a system command through a card reader in a JES2 environment and the command is placed between jobs, enter $VS, 'sys command'; if the command is placed within a job, enter //ϕ. In a JES3 environment enter //**T,sysname, sys command.
3. For information on using SLIP, see *OS/VS2 System Programming Library: Supervisor.*

Command (Abbr)	Function	Page	Acceptable From (Note 1)		Command Group
			Job Stream (Note 2)	TS Terminal	
SWAP (G)	Move a volume from one device to another	179			I/O
SWITCH (I)	Manually switch recording of SMF data from one data set to another	181			SYS
TRACE	Keep system tracing active between time subsystem starts and you start the generalized trace facility	183			SYS
TRACK (TR)	Periodically display job information on display consoles	185			INFO
UNLOAD (U)	Remove a volume from system use	187	X		I/O
VARY (V)	Assigning and controlling consoles	190	X		MC
	Changing the master console	193			(Note 3)
	Controlling the hardcopy log	193			(Note 3)
	Changing the status of a secondary console	195			(Note 3)
	Placing an I/O device online or offline	196			(Note 3)
	Placing a range of I/O devices online or offline	196			
	Placing a path online or offline	197	X		(Note 3)
	Placing a path or paths online or offline	197			
	Placing storage online or offline	198	X		MC
	Placing a channel online or offline	199	X		MC
	Placing a processor online or offline	200	X		MC
WRITELOG (W)	Schedule printing of a system log	203	X		SYS
	Change system log output class	203	X		
	Close system log and discontinue log function	203	X		
	Restart system log after closing	203	X		

Notes:

1. The command is acceptable from the job stream or a time sharing terminal as indicated by an X. If neither job stream nor TS terminal is indicated, the command may be entered only from an operator console.
2. If you enter a system command through a card reader in a JES2 environment and the command is placed between jobs, enter $VS, 'sys command'; if the command is placed within a job, enter //ϕ. In a JES3 environment enter //**T,sysname, sys command.
3. I/O command group when specifying a nonconsole device. CONS when specifying a console.

Appendix B: 3066 (Model 168 Display Console)

This appendix describes the physical characteristics of the 3066 (model 168) display console. Chapter 3 describes the functions and operation of the 3066 and other display consoles.

The 3066 operator console operates only in full-capability (input/output) mode; you cannot use it as an output-only display console. Other operator consoles (display and non-display) can be active in the same system with a 3066 display console; you can use the 3066 console to control output-only consoles.

Console Characteristics

The 3066 display console is composed of a display screen and a typewriter keyboard. The screen and keyboard are normally part of the processor control board as shown in Figure B.1. The display screen is a cathode ray tube with 2,800 character positions.

Screen Format

The display screen format is shown in Figure B.1. The screen is divided into four functional areas:

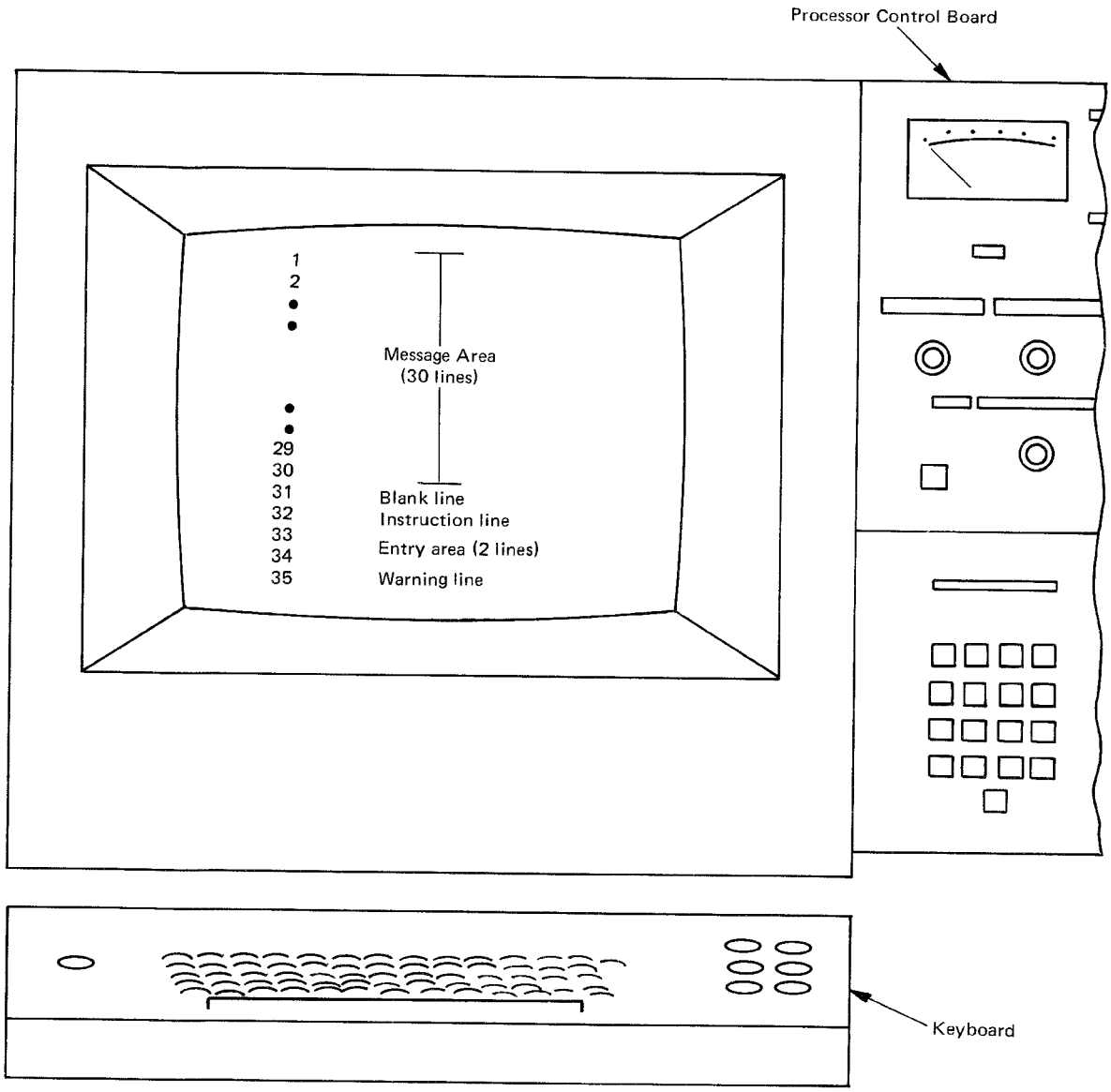
- Message Area: This area has 30 lines of 80 character positions each. The first two character positions in each line contain the number assigned to the message line (if message numbering is in effect), or the number of messages waiting to be displayed (if roll mode is in effect). Positions three, four, and five contain the message-type indicator (*,-,|,+,@). Positions 5-76 contain the text of system messages, operator commands (except the CONTROL command), and system status displays.

The following is a sample message area line.

```
06  IEE334 HALT EOD SUCCESSFUL
```

- Instruction Line: This line is used to display system messages relating to control of the console.
- Entry Area: You use these two lines, in conjunction with the typewriter keyboard, for entering commands and replying to system messages. Each entry area line has 80 character positions; commands that are too large to fit the first line are split at the 80th character and continued in the second line (no special continuation indicator is required). The maximum command length is 126 characters.
- Warning Line: This line is used to display messages that warn you of conditions that may require action.

Note: The PFK display line (line 31) is blank on the 3066 display console, because these consoles do not support light pen command entry.



Note: The labels do not appear on the screen.

Figure B.1. Model 168 Display Console

The Cursor

The position of the cursor on the console screen indicates to the system:

- Where to position the next character that you type.
- What predefined action you want the system to take.
- What messages you want to delete from the screen.

On the 3066, cursor movement is not restricted (that is, the cursor can be positioned anywhere on the screen), but its position is meaningful only in the following locations:

- Any position on a nonaction message line for deleting all messages above that line.
- The asterisk or at sign (@) on an action message for deleting action messages.
- The indicators in a status display control line for framing, updating, and erasing the display.
- Any position within the entry area for entering a command.

The four cursor control keys are advance →, backspace ←, up ↑, and down ↓. Pressing a control key once moves the cursor one character position in the indicated direction; holding the key down moves the cursor continuously in the indicated direction.

The Audible Alarm

The audible alarm is an optional feature of the 3066. It is sounded for one second when:

- You make an error entering a CONTROL or MSGRT command.
- The screen is full and another message is waiting to be displayed.
- An action message appears on the screen.
- The console is placed in roll mode for message deletion.

Response: If the alarm sounds while you are entering a CONTROL or MSGRT command, check the instruction line to see if an error message was issued. If so, follow the procedures given in the message explanation in the message library publication *VS2 System Messages* for the system that you are using.

If the alarm sounds and the following message appears in the warning line:

```
IEE159E MESSAGE WAITING
```

delete some messages from the message area using the procedures described under "Message Deletion" in Chapter 3.

The Visual Alarm

The visual alarm is located on the control panel. It is activated when the audible alarm is sounded. The visual alarm must be turned off manually whenever it is activated. If you leave the visual alarm on, the audible alarm sounds again if one of the conditions listed above occurs. Respond to the visual alarm the same way that you respond to the audible alarm.

How to Enter Commands

To enter a command, you must arrange the information in the proper format and then signal the system that information is ready to be passed.

Entering Commands With the Keyboard

1. Move the cursor to the first position in the entry area.
2. Type in the command.
3. Press the END key.

As you type each character of the command, the corresponding character is displayed in the entry area, and the cursor advances to the next character position. When the end of the first entry area line is reached, the cursor advances automatically to the first character position of the next line, permitting continuation of the command. The maximum number of characters that can be entered is 126, but only one command can be entered at a time.

Note: Refer to chapter 3 for entering commands by PFK keys or the selector pen.

Appendix C: 2250 Display Device

This appendix describes the physical characteristics of the 2250 display console. Chapter 3 describes the functions and operation of the 2250 and other display consoles.

The 2250 display device operates only in full-capability (input/output) mode; you cannot use it as an output-only console. Other operator consoles (display and non-display) can be used in the same system with the 2250 display console; you can use the 2250 to control output-only consoles.

Console Characteristics

The 2250 display device is composed of a display screen and a typewriter keyboard. The display screen is a cathode ray tube with 3848 character positions. The keyboard has 53 keys, including 4 cursor control keys. The light pen, the program function keyboard, and the audible alarm are optional features.

Screen Format

The display screen format is shown in Figure C.1. The screen is divided into five functional areas:

- *Message Area:* This area has 47 lines of 74 character positions each. The first two positions in each line contain the number assigned to the message line, if message numbering is in effect, or the number of messages waiting to be displayed, if roll mode is in effect. Positions three, four, and five contain the message-type indicators (*, | ,-,+,@). Positions 5-74 contain the text of system messages, operator commands (except the CONTROL command), and system status displays. The following is a sample message area line:

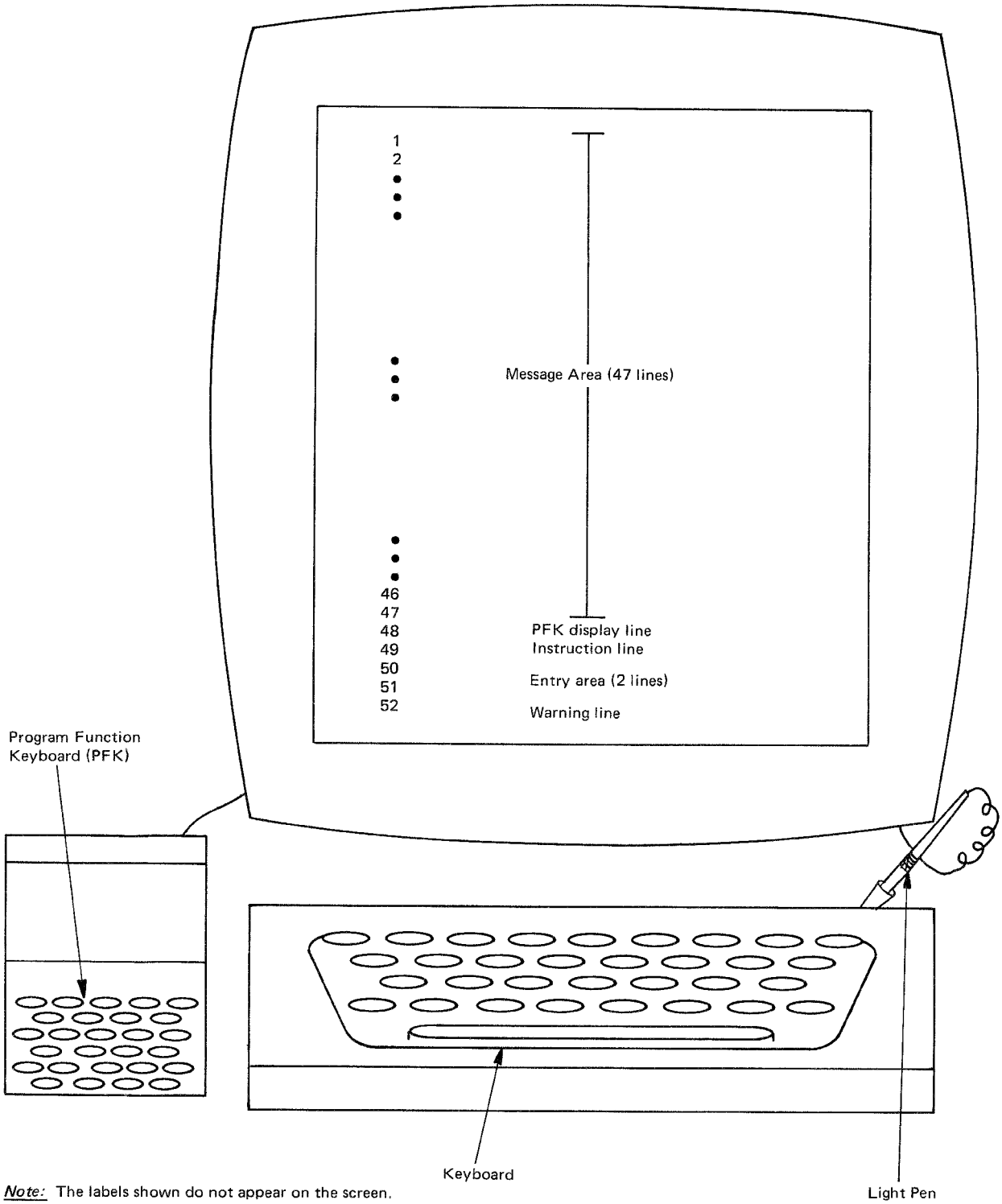
  ```
  0 6 IEE334I HALT EOD SUCCESSFUL
  ```

- *PFK Display Line:* This line is used for displaying program function key (PFK) numbers that you use to enter commands with the selector pen. When the display is requested, the line appears as follows:

  ```
  0 1 2 3 4 5 6 7 8 9 10 11
  ```

This line is blank and separates the message area and the instruction line when the light pen is not in use.

- *Instruction Line:* This line is used for system messages pertaining to control of the console.
- *Entry Area:* You use these two lines, in conjunction with the typewriter keyboard, to enter commands and to reply to messages. Each entry area line has 70 character positions; commands that are too large to fit in the first in are split at the 70th character and continued on the second line (no special continuation indicator is required). Maximum command length is 126 characters.
- *Warning Line:* This line contains messages that warn you of conditions that could require action.



Note: The labels shown do not appear on the screen.

Figure C.1. 2250 Display Console

The Cursor

The position of the cursor on the console screen tells the system where the next character that you type is to be positioned; the cursor cannot be moved out of the entry area.

Cursor movement is controlled by the ADVANCE, BACKSPACE and CONTINUE keys, that are located on the right side of the typewriter keyboard, and by the JUMP key, which is located on the left side of the typewriter keyboard:

- The ADVANCE key moves the cursor one character position to the right.
- The BACKSPACE key moves the cursor one character position to the left.
- The CONTINUE key, when pressed at the same time as either the BACKSPACE or ADVANCE KEY, moves the cursor continuously to the left or to the right.
- The JUMP key moves the cursor to the first position in the entry area.

The Light Pen

The light pen is an optional feature of the 2250 display consoles. The light pen is a light-sensing device that you can use to identify a portion of the screen for system action. To use the pen, position it on the screen over the area that you are identifying, and activate the light-sensing mechanism. The pen is activated either by pressing a foot switch or by pressing the pen against the screen (depending on the type of light pen that you are using).

The light pen can be positioned anywhere on the screen, but its position is recognized as valid only in the following areas:

- On a nonaction message line, to delete all messages above that line.
- On the asterisk in an action message line, to delete that message.
- On *F, *H, *U, *E, or *PT in the control line of a status display, to frame, hold, update, or erase the display.
- On *ENTER*, *CANCEL*, or *DC,K* in the instruction line, to enter a command, cancel a command, or to request a display of the CONTROL command operands.
- On a number displayed in the PFK display line, to enter a command.

If the light pen is positioned in an invalid location, the following message is displayed in the instruction line:

```
IEE164I ILLEGAL LIGHT PEN-CURSOR DETECT
```

The Program Function Keyboard

The program function keyboard is an optional feature available on the 2250 display consoles. It consists of a board of keys (called PFKs) that send a signal to the system whenever one of the keys is pressed. The system responds to the signal according to the way it has been programmed. In the operator console mode, up to twelve PFKs can be associated with operator commands.

The Audible Alarm

The audible alarm is an optional feature of the 2250 display consoles. It is sounded for one second under the following conditions:

- When you make an error entering a CONTROL or MSGRT command.
- When the screen is full and another message is waiting to be displayed.
- When an action message appears on the screen.
- When the console is placed in roll mode for message deletion.

Response: If the alarm sounds while you are entering a CONTROL or MSGRT command, check the instruction line to see if an error message was issued. If so, follow the procedures given in the message explanation in the message library publication *VS2 System Messages* for the system you are using.

If the alarm sounds and the following message appears in the warning line:

```
IEE159E MESSAGE WAITING
```

delete some messages from the message area using the procedures described under "How to Delete Messages" in Chapter 3.

How to Enter Commands

Commands can be entered using the keyboard, the program function keyboard, or the light pen. Refer to Chapter 3 for entering commands by PFK keys or the light pen.

Entering Commands With the Keyboard

1. Move the cursor to the first position in the entry area.
2. Type in the command.
3. Enter the command by holding down the ALT key (located on the left side of the keyboard) and pressing the END key (numeric 5). You can also enter the command by positioning the light pen over the *ENTER* indicator in the instruction line.

As you type each character of the command, the corresponding character is displayed in the entry area, and the cursor is advanced to the next character position. When the end of the first entry area line is reached, the cursor advances automatically to the first character position of the next line, permitting continuation of the command. The maximum number of characters that can be entered is 126; but only one command can be entered at a time.

Note: Refer to Chapter 3 for entering commands by PFK keys or the light pen.

Appendix D: 2260 Display Station

This appendix describes the physical characteristics of the 2260 display station when used as an operator's console. Chapter 3 describes the functions and operation of the 2260 and other display consoles.

Console Characteristics

The 2260 display console is composed of a display screen and a typewriter keyboard. The display screen is a cathode-ray tube with 960 character positions. The typewriter keyboard is a 50-key keyboard including special keys to control cursor movement. Figure D.1 shows a 2260 display console.

Screen Format

The screen is divided into four functional areas (see Figure D.1):

- *Message Area:* This area has eight lines of 80 character positions each. The first two character positions contain the message number (if message numbering is in effect). Positions three, four, and five contain message-type indicators (*, |, -, +, @). Positions five through 80 contain the text of system messages, operator commands, message replies, and system status displays. The following is a sample message area line:

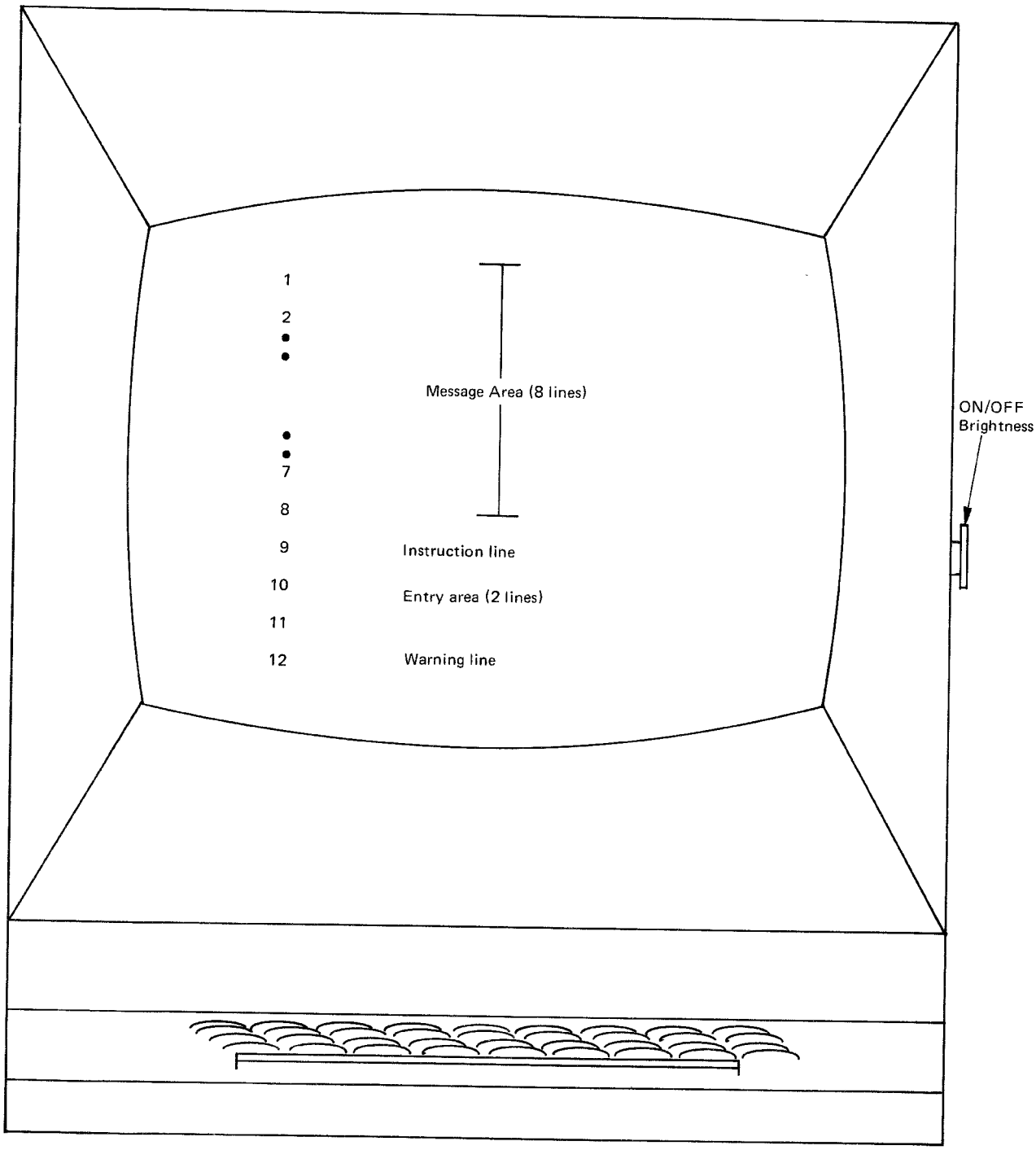
  ```
  04 IEE334I HALT EOD SUCCESSFUL
  ```

- *Instruction Line:* This line is used to display system messages relating to control of the console.
- *Entry Area:* These two lines are used to enter commands and to reply to messages.
- *Warning Line:* This line is used to display messages that warn you of conditions that can require action.

Special Screen Characters

The 2260 display console uses two special characters to assist you in entering commands:

- The start manual input (START MI) indicator (▶) appears in the first data entry position (in the entry area) when the system is ready for you to type in a command or a message reply.
- The end of message (EOM) indicator (■) appears after the last character in the entry area when you signal that the information in the entry area is to be read into the system.



Note: The labels shown do not appear on the screen.

Figure D.1. 2260 Display Console

The Cursor

The cursor appears as a vertical bar. It is located below the character line and to the left of the position that it is designating. The cursor can be moved freely about the console screen. The position of the cursor indicates to the system:

- Where to position the next character that you type.
- What predefined action you want the system to take.
- What messages you want deleted from the screen.

On the 2260 display console, the cursor can be positioned anywhere on the screen, but its position is meaningful only in the following locations:

- Any position on a nonaction message line for deleting all messages above that line.
- The asterisk in an action message for deleting action messages.
- The *F, *E, *U, *H, or *PT in a status display control line for controlling the display.
- Any position within the entry area for entering a command.

The cursor is controlled by the advance (space bar), backspace, up, and down keys. Pressing these keys moves the cursor one space to the right (advance), one space to the left (backspace), one line up, and one line down.

Note: The ERASE DISPLAY key erases the screen and moves the cursor to the upper left hand corner of the screen. This key is not intended for use in operator console mode. To restore the screen (if ERASE DISPLAY is pressed accidentally):

1. Position the cursor to the first position in line ten (the first line of the entry area).
2. Hold down the SHIFT key and press the START key (this displays the START MI symbol (▶) in the entry area.
3. Hold down the SHIFT key and press the ENTER key twice.

How to Enter Commands with the Keyboard

1. Position the cursor in the entry area to the right of the START MI symbol (▶).
2. Put the console in hold mode by holding down the SHIFT key and pressing the ENTER key; wait for the following message to appear in the second line of the entry area:

   ```
   IEE167E OUTPUT IN HOLD MODE
   ```

3. Type in the command.
4. Move the cursor to the next character position after the last character in the command.
5. Enter the command by holding down the SHIFT key and pressing the ENTER key.

You can position the cursor to the first data entry position by pressing the START key. As you type each character of the command, the corresponding character is displayed in the entry area, and the cursor is advanced to the next character position. When the end of the first entry area line is reached, the cursor advances automatically to the first character position of the second entry area line, permitting continuation of the command. The maximum number of characters that can be entered is 126; but only one command can be entered at a time.

Warning: While message IEE167E is visible, no additional messages can be written to the screen. The system must save messages until the operator presses the ENTER key, indicating the end of the command. If the console is left in hold mode for an extended period, the saved messages use too much storage and the system enters a wait state. Press the ENTER key to release hold mode and allow the system to continue.

Entering the Command: Position the cursor after the last character in the command before you press the SHIFT and ENTER keys. When you press the SHIFT and ENTER keys, the EOM indicator (■) momentarily replaces the cursor; its appearance indicates that the system is receiving the command. After the command is accepted by the system, the command disappears from the entry area (unless it is a CONTROL command), and the cursor is repositioned to the character position immediately following the START MI symbol. Commands other than the CONTROL command reappear in the message area when screen space is available. The CONTROL command remains in the entry area until the requested action takes place.

Appendix E: 3277, 3056, 3036, and 3158 Display Consoles

This appendix describes the physical characteristics of the 3277 display devices, Models 1 and 2, the 3056 display console, the 3036 display console, and the 3158 (Model 158) display console. Chapter 3 describes the functions and operation of the 3036, 3056, 3158, 3277, and other display consoles.

For the operator, the 3056 display console and the 3036 display console are similar to the 3158 display console. Unless differences are noted, the 3158 console information applies to the 3056 display console and the 3036 display console.

The 3277 Model 2 display console can be used in both full-capability and output-only mode of operation: the 3277 Model 1 functions only in output-only mode.

Console Characteristics

The 3277 Model 2, the 3056 display console, the 3036 display console and the 3158 display console are composed of a display screen and usually a typewriter keyboard. If a console configuration does not include a keyboard, it can function only as an output-only device (see "Console Mode" in chapter 3). There are differences between the 3277 Model 2 and the others, several of which are noted in Figure 3.1 in Chapter 3, Summary of Display Console Features. Other differences are explained throughout this appendix.

Screen Format

Figure E.1 shows the display screen format. The screen is divided into five functional areas:

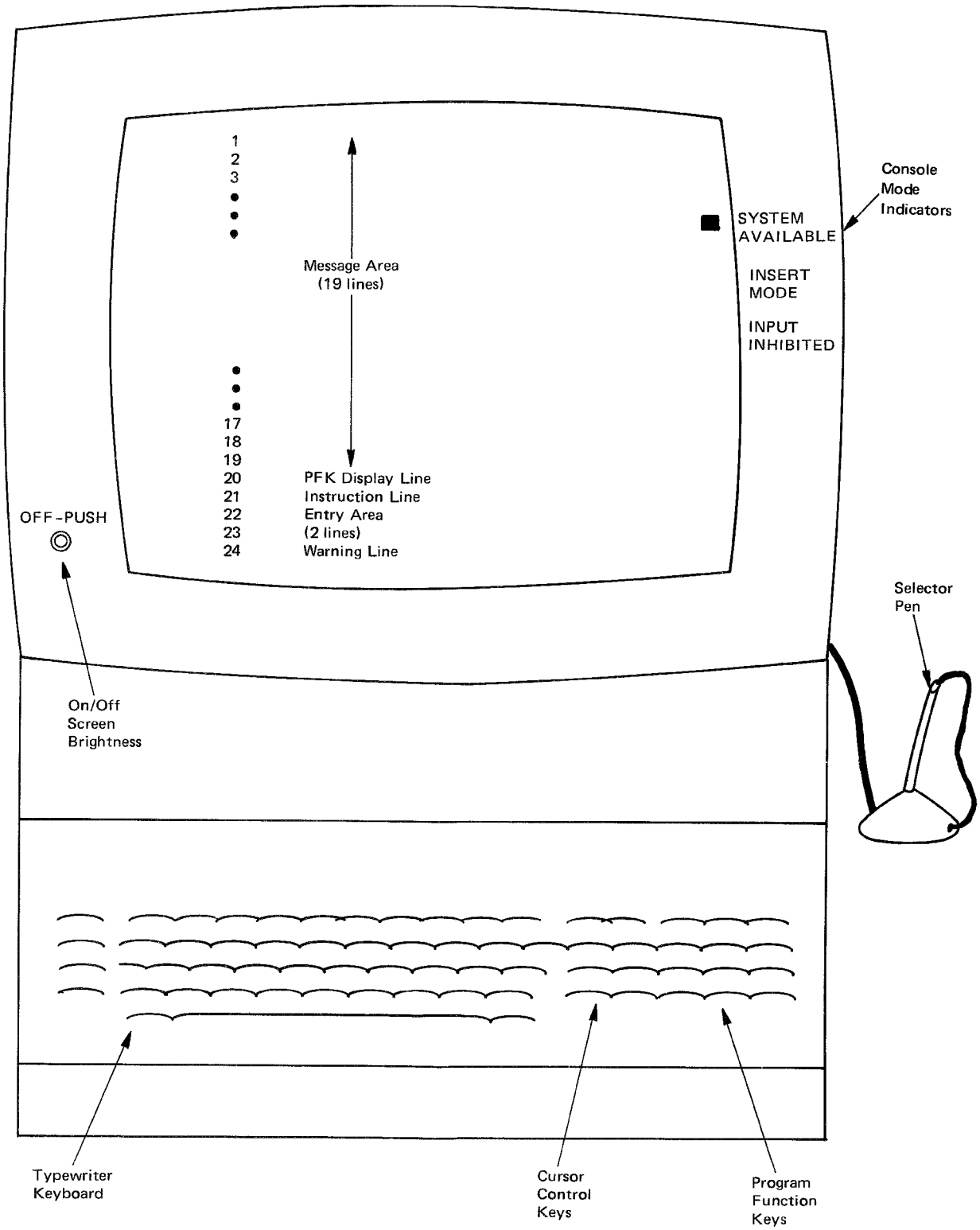
- *Message Area:* The message area is 19 lines long; it contains system messages, operator commands, and status displays. Each message area line contains 78 visible character positions. (There are two system indicator positions preceding the first visible character position; these always appear blank on the screen). The first two visible positions contain the message line number (if message numbering is in effect). Positions three, four, and five contain the message-type indicator (*, | ,-,+,@). The message text is contained in positions 5 through 78. Position 5 can contain either a message type indicator or text. Messages longer than 78 characters are continued on the next screen line, beginning in position six. The following is a sample message area line:

```
01 IEE334I HALT EOD SUCCESSFUL
```

Note: If you have requested that the device display a time stamp or job id, multi-line messages are truncated at screen position 78; they are not continued onto the next screen line. You may choose not to display the time stamp and job id to prevent truncation.

- *PFK Display Line:* This line is used for displaying PFK numbers (for selection with the selector pen). When the display is requested, the line appears as follows:

```
1 2 3 4 5 6 7 8 9 10 11 12
```



Note: The labels shown do not appear on the screen.

Figure E.1. 3277 Model 2 Display Console with Optional Features Attached

- *Instruction Line:* This line is used for system messages relating to control of the console. This example shows the appearance of the instruction line with system message IEE152I displayed:

  ```
  IEE152I ENTER CANCEL D C,K
  ```

  Messages appearing in this line, other than message IEE152I, are intensified (made brighter than the other messages on the screen).
- *Entry Area:* This area occupies two screen lines. The operator uses this line, in conjunction with the cursor and the typewriter keyboard, to enter commands to the system. The first line of the entry area contains 78 character positions (the first two positions are used for system indicators and always appear blank on the screen). The second line contains 80 character positions. the following example shows a DISPLAY UNIT command in the entry area:

  ```
  DU,L=12A
  ```

- *Warning Line:* This line contains warning messages that usually require operator action. Messages appearing in this line are intensified (made brighter than the other messages on the screen). The following is an example of a warning line containing message IEE163I:

  ```
  IEE163I MODE=R
  ```

Special Screen Characters

The 3277 model 2, the 3158, and the 3036 use special screen characters to indicate the console's mode of operation and the status of certain screen messages.

Console Mode: The 3277 model 2 display console and the 3158 display console differ in that they show console mode in different ways. For the 3277 model 2 display console, the mode marker (■) appears on the right side of the screen adjacent to one of the printed mode indicators: INSERT MODE, INPUT INHIBITED or SYSTEM AVAILABLE. Insert mode, which is requested by means of the INS MODE key, allows the operator to correct typing errors by inserting extra letters within a line of text. After correcting the error, the operator restores normal operating mode by pressing the RESET key. INPUT INHIBITED indicates that the system is temporarily unable to receive input. Normal system operation can sometimes be restored by pressing the RESET key, although this key is not honored if certain operations are in process. The system restores normal operating mode after most processes that result in inhibited input. SYSTEM AVAILABLE indicates that the unit is in communication with the system and is able to send or receive messages.

The 3158 and 3036 display consoles do not have console mode indicators. There is an additional line on the screen, line 25, which is not shown in Figure E.1. An appropriate message and a PSW are displayed in line 25, and the operator references these for console mode.

The Cursor

The cursor appears on the screen as an underscore. Its location indicates to the system:

- Where to position the next character that you type.
- What predefined action you want the system to take.
- What messages you want to delete from the screen.

The cursor can be positioned anywhere on the screen, but its position is meaningful only in the following locations:

- Any position on a nonaction message line for deleting action messages.
- The asterisk on an action message for deleting action messages.
- The special indicators in a status display control line for holding, framing, updating, or erasing the display.
- The special indicators in the instruction line for entering or canceling commands.
- Any position in the entry area for entering commands.

Cursor Control Keys: The four basic cursor control keys are located on the right side of the keyboard. Pressing one of these keys moves the cursor continuously in the indicated direction: Up ↑, down ↓, left ←, or right →. Cursor movement stops when the key is released. The other cursor control keys move the cursor as follows:

- ←The backspace key, located on the right side of the top line of keys, moves the cursor to the left.
- The tab key moves the cursor to the next unprotected field (an unprotected field is one in which you can type data). Because the only unprotected field on the display console screen is the entry area, pressing this key moves the cursor to the first position in the entry area.
- |←The back-tab key moves the cursor back to the first position of an unprotected field. It can be used to return the cursor to the first data position in the entry area.
- ↵ The new line key moves the cursor to the first unprotected field in any line below the line in which the cursor is located. It can be used to move the cursor to the first position in the entry area.

Special Keys

CLEAR Key: The CLEAR key is not intended for use in operator console mode. This key clears the screen and moves the cursor to the first character position (upper left corner of the screen). If you accidentally press the CLEAR key, press the CANCEL key to restore the screen. After pressing the CANCEL key, check the screen to see that no information has been lost. Possible problem areas include:

- *Status Displays:* The CLEAR key removes status displays from the screen but does not restore display areas for use by other messages. To restore a display area, either request another display for the area or issue a CONTROL E,D,L=cca command specifying the area in the L=cca operand. Dynamic status displays resume normal operation when the time interval in progress elapses. (See "System Status Displays" in chapter 3.)
- *PFK Display Line:* To restore this line, first issue a CONTROL E, PFK command and then issue a CONTROL D,PFK command (see "Selector Pen Command Entry" in chapter 3).

TEST REQ Key: This key is used to test the console; it is not used in normal operator console functions.

PA1 Key: This key has no function in operator console mode.

The Selector Pen

The selector pen is an optional feature of the 3277 Model 2 and standard on the 3158. (It is not available on the 3036 or the 3056 display console.) The selector pen is a light-sensitive device used to identify an area of the screen for system action. To use the pen, position it on the screen over the area that you are identifying and exert slight pressure to activate the light-sensing mechanism.

You can position the selector pen anywhere on the screen, but its position is meaningful only in the following locations:

- On a nonaction message line for deleting all messages above that line.
- On an indicator in the control line of a status display for framing and updating control, or to erase the display.
- On ENTER, CANCEL, or D C,K in the instruction line for entering a command, canceling a command, or requesting a display of CONTROL command operands.
- On a key number displayed in the PFK display line for entering commands.

If the selector pen is positioned in any other location, the following message appears in the instruction line:

```
IEE164I ILLEGAL LIGHT PEN - CURSOR DETECT
```

To respond to the message, move the selector pen to a valid location.

Program Function Keyboard

The program function keyboard is an optional feature available on the 3277 Model 2 display console. The typewriter keyboard of the 3158 and 3036 can serve as a program function keyboard when used in a special way; refer to "Entering Commands With the PFKs" in Chapter 3.

The program function keyboard on the 3277 Model 2 consists of a group of 12 keys called PFKs. The keys are numbered PF1 through PF12. Pressing a PFK sends a signal to the system; the system responds to the signal according to the way it has been programmed. In the operator console mode, up to twelve PFKs can be associated with operator commands. The number of PFKs available for command entry is determined during system generation, and the operator associates commands with the PFKs after IPL. Refer to "Defining Commands for Each PFK", in Chapter 3.

The Alarms

The audible alarm is an optional feature of the 3277 Models 1 and 2 display console and standard on the 3158. With the 3158, a visual alarm is also standard. These alarms signal for one second when:

- You make an error entering a CONTROL or MSGRT command.
- The console is placed in roll mode for message deletion.
- The screen is full and another message is waiting to be displayed.
- An action message appears on the screen.

Note: The 3036 display console also has a visual alarm. However, it signals only when the power and cooling alarm sounds. It signals until you press the reset key.

Response: If an alarm signals while you are entering a CONTROL or MSGRT command, check the instruction line to see if an error message was issued. If so, follow the procedures given in the message explanation *VS/2 System Messages*.

If an alarm signals and the following messages appears in the warning line:

```
IEE159E MESSAGE WAITING
```

delete some screen messages using the procedures described under "How to Delete Messages" in Chapter 3.

Message Intensity

The 3277 can present messages with two degrees of intensity or contrast. The intensity is controlled by the brightness controls on the on/off switch.

The following message types are intensified:

- Error messages in the instruction line.
- Roll and roll-deletable mode messages in the warning line.

How to Enter Commands - 3277 Models 1 and 2, 3056, 3036, 3158

You can enter commands with the typewriter keyboard, the program function keyboard, or the selector pen (in conjunction with the PFK display line). Also, on 3158, commands associated with PFK numbers are entered by using the typewriter keyboard in a special way.

Use the following procedure to enter a command or to reply to a WTOR message:

1. Move the cursor to the first position in the entry area.
2. Type in the command.
3. The command by *one* of the following methods:
   - Press the ENTER key.
   - Position the selector pen over the ENTER indicator in the instruction line.
   - Position the cursor under the ENTER indicator in the instruction line and press the ENTER key.

3277 Model 1 Display Console

The 3277 Model 1 display console operates only as an output-only console in message stream mode. The console screen format is shown in Figure E.2.

Each message area line is 38 characters in length. The first two character positions contain the number assigned to the message line. Positions three, four, and five contain the message-type indicator (*, , ,+,@). Messages longer than 38 characters are continued in position three of subsequent lines. Warning messages are displayed in positions 1-37 of the warning line.

Note: If you have requested that the device display a time stamp or job id, multi-line messages are truncated at screen position 37; they are not continued on the next line. You can choose not to display the time stamp and job id to prevent truncation.

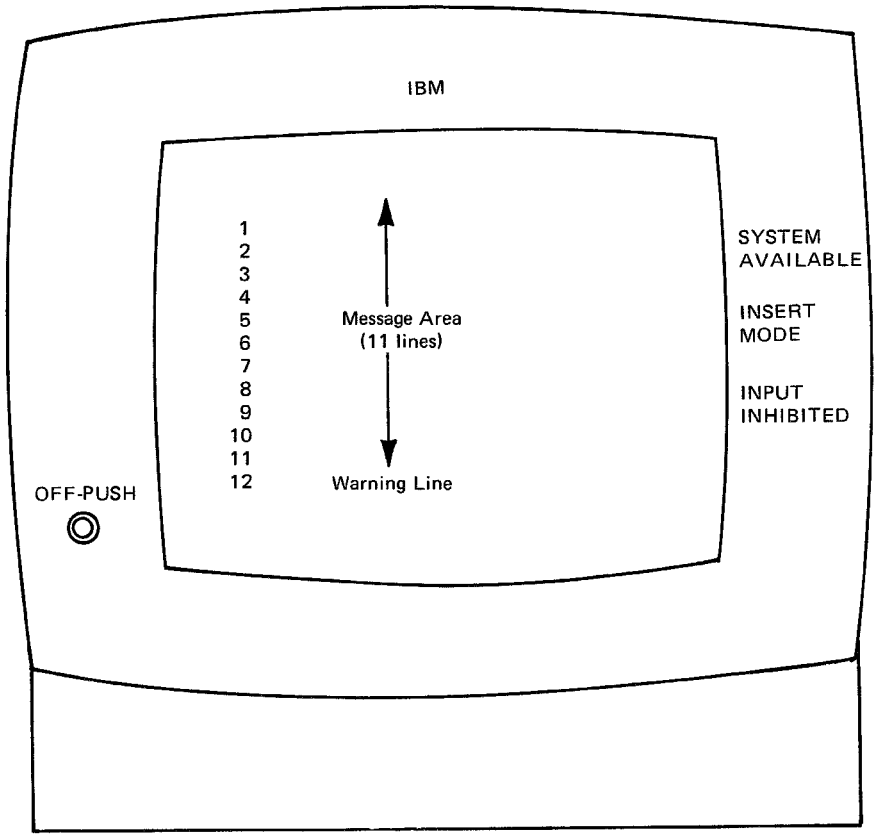


Figure E.2. 3277 Model 1 Display Console

Appendix F: IBM Printer-Keyboard Consoles (1052, 3210, 3215)

The IBM printer-keyboard consoles can be attached directly to the processor or be used as a stand-alone console with an IBM 2150 console. The maximum line length is 126 characters for both input and output messages.

How to Enter a Command

If the PROCEED light is on:

- Type the command
- Press the END key

If the PROCEED light is not on:

- Press the REQUEST key on the device
- Wait for the PROCEED light to go on
- Type the command
- Press the END key or, on the 1052 console, signal EOB by holding down the alternate coding key and pressing the numeric 5 key.

To correct errors, cancel the entire line by pressing the CANCEL key. On the 1052 printer-keyboard console, hold down the alternate coding key and press the numeric 0 key. After performing this cancel action, the command can be retyped. If your device can be backspaced, simple errors can be corrected by backspacing and retyping from the point of error.

Appendix G: IBM 2740 Communications Terminal

The IBM 2740 Communications Terminal may be used as an operator's console. The 2740 operator's console is connected to either an IBM 2701 Data Adapter Unit or an IBM 2702 Transmission Control Unit.

How to Enter a Command or Reply

- Set the Local/Communicate switch to COM (Communicate Mode). The standby light (S light) must come on before you proceed.
- Press the Bid key. Once your bid has been placed on the communications line, the transmit light (T light) turns on, the S light turns off, and the typewriter motor turns on.
- Enter your command or reply and press the RETURN key. If your console does not have the automatic EOB feature on the RETURN key, press the EOB (End-of-Block) key. Your console momentarily switches to receive status (R light on) in order to accept the checking reply from the operating system (Receive and Restart lights on). If the command or reply was received correctly, your console automatically switches back to standby status (Standby light on and Restart light off).

How to Answer a Negative System Response

If the operating system receives the command or reply incorrectly, you get a negative response that causes a dash or underscore symbol to be printed as the first character in the next line. A negative response also causes the keyboard to lock, the Restart light to remain on, the Transmit light to turn on, and the alarm to sound at your terminal. Operating the Restart key unlocks the keyboard, turns off the alarm, and turns off the Restart light. You can then either resend the command or reply, or press EOB to return the 2740 to standby status.

How to Correct an Error

If you make an error when entering your command or reply, correct the error by backspacing to the characters in error and type over them with the correct characters; start at or before the first character in error and continue to the end of the command or reply. If you wish to cancel a command or reply, backspace once and then press the Return key.

How to Receive Messages

The procedure for receiving messages from the operating system differs slightly from entering commands and replies. The operating system must ready your console by entering a call. For your terminal to receive a call, the following conditions must be satisfied:

- Local/Communicate switch set to COM (if not, alarm sounds when call is received).
- Paper inserted (if not, alarm sounds when call is received).
- Standby light on.

When your terminal receives a call, the Standby light turns off, the Receive light turns on, and the typewriter motor is turned on. At this time your keyboard is locked and the message is printed. At the end of the message, the Standby light is turned on, the keyboard remains locked except for the Bid key and EOT key, and the console motors are turned off.

If the operating system prints a dash or underscore as the first character of the line following the message, your 2740 might have received a part of the message incorrectly. The message is resent. If after five retries your console is still receiving the message incorrectly, the system switches automatically to the alternate console.

If your console runs out of paper while operating in receive mode, it switches to a no-status standby condition. All status lights are turned off and the keyboard locks. The message being sent is lost.

Additional information on the 2740 Communications Terminal can be found in *IBM 2740/2741 Communications Terminal-Operator's Guide*.

Glossary

This glossary includes definitions developed by the American National Institute (ANSI). This material is reproduced from the American National Dictionary for Information Processing, copyright 1977 by the Computer and Business Equipment Manufacturers Association, copies of which may be purchased form the American National Standards Institute, 1430 Broadway, New York 10018.

If the term you are seeking does not appear in this glossary, refer to *Data Processing Glossary*, GC20-1699.

ABEND. Abnormal end of task.

abnormal end of task (ABEND). Termination of a task prior to its completion because of an error condition that cannot be resolved by recovery facilities while the task is executing.

accepting. The process by which a terminal obtains a message transmitted to it from the computer. Contrast with *entering*. See also *receiving, sending*.

activate. In VTAM, to connect a node to another node, or to make the node available for connection.

active line. A line that is currently available for transmission of data.

address. (1) * An identification, as represented by a name, label, or number, for a register, location in storage, or any other data source or destination such as the location of a station in a communication network. (2) * Loosely, any part of an instruction that specifies the location of an operand for the instruction.

address space. The complete range of addresses that is available to a programmer. See also *virtual address space*.

address stop. A capability to specify at the system console an address which when encountered causes a halt in processing.

allocate. To assign a resource for use in performing a specific task.

* **allocation.** See *storage allocation*.

allocation of data sets. The process of assigning auxiliary storage space to a data set. See also *dynamic data set definition*.

alphabetic character. A letter or other symbol, excluding digits, used in a language.

* **alphameric.** Same as *alphanumeric*.

alphameric characters. In programming, usually the characters A through Z, digits 0 through 9, and #, $, and @.

* **alphanumeric.** Pertaining to a character set that contains letters, digits, and usually other characters, such as punctuation marks. Synonymous with *alphameric*.

alternate CPU recovery (ACR). The component of MVS that automatically removes a processor from a tightly-coupled multiprocessing system for some types of catastrophic machine checks.

alternate routing. A secondary or backup communications path; used if normal routing is not possible.

American National Standard Labels. Magnetic tape labels that conform to the conventions established by the American National Standards Institute. Synonymous with ASCII label.

attribute. A characteristic; for example, attributes of data include record length, record format, data set name, associated device type and volume identification, use, and creation date.

automatic mode. The setup and selection of jobs on a printer is to be controlled by JES as opposed to being controlled by the operator through the use of operator commands.

automatic restart. A restart that takes place during the current run, that is, without resubmitting the job. An automatic restart can occur within a job step or at the beginning of a job step. Contrast with *deferred restart*.

automatic volume recognition (AVR). A feature that allows the operator to mount labeled volumes on available I/O devices before the volumes are needed by a job step.

auxiliary storage. (1) Data storage other than main storage; for example, storage on magnetic tape or direct access devices. Synonymous with external storage, secondary storage. (2) * A storage that supplements another storage. Contrast with *main storage*.

availability. The degree to which a system or resource is ready when needed to process data.

AVR. Automatic volume recognition.

background. (1) In multiprogramming, the environment in which low-priority programs are executed. (2) Under TSO, the environment in which jobs submitted through the SUBMIT command or SYSIN are executed. One job step at a time is assigned to a region of main storage, and remains in main storage to completion. Contrast with *foreground*.

background job. (1) A low-priority job, usually a batched or noninteractive job. (2) Under TSO, a job entered through the SUBMIT command or through SYSIN. Contrast with *foreground job*.

* **backspace.** To move back the reading or display position according to a prescribed format. Contrast with *space* (4).

* **backspace character.** A format effector that causes the location of the printing or display position to be moved backward one printing or display space.

batch processing. (1) * Pertaining to the technique of executing a set of computer programs such that each is completed before the next program of the set is started. (2) * Pertaining to the sequential input of computer programs or data. (3) * Loosely, the execution of computer programs serially. (4) Under TSO, the processing of one job step in a region, so called because jobs are submitted in a group or batch. (5) See also *execution batch processing*.

batched job. (1) A job that is grouped with other jobs as input to a computing system. (2) A job whose job control statements are grouped with job control statements of other jobs as input to a computing system.

* **beginning-of-tape marker.** A marker on a magnetic tape used to indicate the beginning of the permissible recording area, for example, a photo-reflective strip, a transparent section of tape.

blank. (1) * A part of a data medium in which no characters are recorded. (2) On a CRT display device, to turn off the electron beam so that no glow is produced on the face of the screen.

* **blank character.** Same as *space character*.

broadcast. The simultaneous dissemination of information to a number of stations.

broadcast data set. Under TSO, a system data set containing messages and notices from the system operator, administrators, and other users. Its contents are displayed to each terminal user when he logs on the system, unless suppressed by the user.

buffer. (1) * A routine or storage used to compensate for a difference in rate of flow of data, or time of occurrence of events, when transmitting data from one device to another. (2) An area of storage that is temporarily reserved for use in performing an input/output operation, into which data is read or from which data is written.

byte. (1) * A sequence of adjacent binary digits operated upon as a unit and usually shorter than a computer word. (2) The representation of a character. (3) In System/360 and System/370, a sequence of eight adjacent binary digits that are operated upon as a unit and that constitute the smallest addressable unit in the system.

card punch. A device to record information in cards by punching holes in the cards to represent letters, digits, and special characters.

card reader. A device that senses and translates into machine code the holes in punched cards.

* **carriage control tape.** A tape that contains line feed control data for a printing device.

catalog. (1) * An ordered compilation of item descriptions and sufficient information to afford access to the items. (2) The collection of all data set indexes that are used by the control program to locate a volume containing a specific data set. (3) To include the volume identification of a data set in the catalog.

cataloged data set. A data set that is represented in an index, or hierarchy of indexes, that provide the means for locating it.

cataloged procedure. A set of job control statements that has been placed in a partitioned data set called the procedure library, and can be retrieved by naming it in an execute (EXEC) statement or started by the START command.

cathode ray tube. An electronic vacuum tube, such as a television picture tube, that can be used to display graphic images.

* **central processing unit (CPU).** A unit of a computer that includes the circuits controlling the interpretation and execution of instructions. Synonymous with *main frame*.

* **chain printer.** A printer in which the type slugs are carried by the links of a revolving chain. Contrast with *train printer*.

channel. (1) A path along which signals can be sent, for example, data channel, output channel. (2) The portion of a storage medium that is accessible to a given reading or writing station, for example, track, band. (3) In communication, a means of one-way transmission. Several channels may share common equipment. For example, in frequency multiplexing carrier systems, each channel uses a particular frequency band that is reserved for it. (4) A hardware device that connects the CPU and main storage with the I/O control units.

Channel Reconfiguration Hardware (CRH). That function of MVS that allows one processor to access paths attached to the other processor. This is a recovery feature designed to allow access to all I/O devices of a 168 MP system even though one of the processors is offline either through a machine check or as a result of a VARY CPU offline command.

channel-to-channel adapter (CTC). A hardware device that can be used to connect two channels on the same computing system or on different systems.

* **character.** A letter, digit, or other symbol that is used as a part of the organization, control, or representation of data.

character position. Same as *display position*.

character row. Same as *display line*.

* **character string.** A string consisting solely of characters.

checkpoint. (1) * A place in a routine where a check, or a recording of data for restart purposes, is performed. (2) A point at which information about the status of a job and the system can be recorded so that the job step can be later restarted. (3) To record such information.

checkpoint data set. A sequential or partitioned data set containing a collection of checkpoint entries. If a checkpoint data set is a partitioned data set, each checkpoint entry is a member.

checkpoint restart. The process of resuming a job at a checkpoint within the job step that caused abnormal termination. The restart may be automatic or deferred, where deferred restart involves resubmitting the job. See also *automatic restart, deferred restart*. Contrast with *step restart*.

checkpoint/restart facility. (1) A facility for restarting execution of a program at some point other than at the beginning, after the program was terminated due to a program or system failure. A restart can begin at a checkpoint or from the beginning of a job step, and uses checkpoint records to reinitialize the system. (2) Under TCAM, a facility that records the status of the teleprocessing network at designated intervals or following certain events. Following system failure, the system can be restarted and continue without loss of messages.

* **clear.** To place one or more storage locations into a prescribed state, usually zero or the space character. Contrast with *set*.

* **clock.** (1) A device that generates periodic signals used for synchronization. (2) A device that measures and indicates time. (3) A register or storage area whose contents change at regular intervals in such a way as to measure time.

cold start. Same as *initial program load*.

command name. The first term in a command, usually followed by operands.

command processing. The reading, analyzing, and performing of commands issued via a console or through an input stream.

communication. Transmission of intelligence between points of origin and reception without alteration of sequence or structure of the information content.

communication line. Any medium, such as a wire or a telephone circuit, that connects a remote station with a computer.

component. (1) In teleprocessing, one or more input/output devices attached to a single control unit, and together making up one remote terminal or station. (2) A point in a communications system at which data can enter or leave; an input/output device. A component is always attached to a terminal control unit.

* **computer.** (1) A data processor that can perform substantial computation, including numerous arithmetic or logic operations, without intervention by a human operator during the run.

* **computer network.** A complex consisting of two or more interconnected computing units.

* **computer program.** A series of instructions or statements, in a form acceptable to a computer, prepared in order to achieve a certain result.

computer word. (1) * A sequence of bits or characters treated as a unit and capable of being stored in one computer location. (2) In System/360 and System/370, 32 bits or 4 bytes.

computing system. A central processing unit, with main storage, input/output channels, control units, direct access storage devices, and input/output devices connected to it.

configuration. The group of machines, devices, and programs that make up a data processing system.

* **console.** That part of a computer used for communication between the operator or maintenance engineer and the computer.

* **control panel.** A part of a computer console that contains manual controls.

control program. A program that is designed to schedule and supervise the performance of data processing work by a computing system.

control unit. A device that controls input/output operations at one or more devices.

conversational. Pertaining to a program or a system that carries on a dialog with a terminal user, alternately accepting input and then responding to the input quickly enough for the user to maintain his train of thought.

* **CPU.** *Central processing unit*.

* **CRT display.** Cathode ray tube display.

CRT display device. A display device on which images are produced on a cathode ray tube.

CSA. Common Service Area.

cursor. A movable spot of light on a cathode ray tube unit that indicates where the next character will be entered.

DASD. *Direct access storage device*.

* **data.** (1) A representation of facts, concepts, or instructions in a formalized manner suitable for communication, interpretation, or processing by humans or automatic means. (2) Any representations such as characters or analog quantities to which meaning is, or might be, assigned.

data attribute. See *attribute*.

data definition name. The name of a data definition (DD) statement, which corresponds to a data control block that contains the same name. Abbreviated ddname.

data definition (DD) statement. A job control statement that describes a data set associated with a particular job step.

data set. The major unit of data storage and retrieval in the operating system, consisting of a collection of data in one of several prescribed arrangements and described by control information to which the system has access.

data set label. (1) A collection of information that describes the attributes of a data set and is normally stored on the same volume as the data set. (2) A general term for data set control blocks and tape data set labels.

data set name. The term or phrase used to identify a data set.

ddname. *Data definition name.*

DD statement. *Data definition statement.*

DDR. *Dynamic device reconfiguration.*

deallocate. To release a resource that is assigned to a specific task.

* **decimal.** (1) Pertaining to a characteristic or property involving a selection, choice, or condition in which there are ten possibilities. (2) Pertaining to the number representation system with a radix of ten.

* **decimal digit.** In decimal notation, one of the characters 0 through 9.

default value. The choice among exclusive alternatives made by the system when no explicit choice is specified by the user.

deferred restart. A restart performed by the system on resubmission of a job by the programmer. The operator submits the restart deck to the system through a system input reader. Contrast with *automatic restart.*

degradation machine check (DG). Indicates a continuous degradation of system performance. Degradation machine checks can be reported when system-recovery conditions exceed a machine pre-defined threshold or when unit deletion occurs.

demount. To remove a volume from a tape unit or a direct access device.

destination station. A station to which a message is directed.

device. A mechanical, electrical, or electronic contrivance with a specific purpose.

device control unit. A hardware device that controls the reading, writing, or display of data at one or more input/output devices or terminals.

device number. (1) The reference number assigned to any external device. (2) A part of an external page address that refers to a particular paging device. In OS/VS2, together with a group number and a slot number, it identifies the location of a page in external page storage.

device type. The general name for a kind of device; for example, 2311, 2400, 2400-1.

* **diagnostic.** Pertaining to the detection and isolation of a malfunction or mistake.

digit. A symbol that represents one of the non-negative integers smaller than the radix. For example, in decimal notation, a digit is one of the characters from 0 through 9. Synonymous with *numeric character.*

direct access. (1) Retrieval or storage of data by a reference to its location on a volume, rather than relative to the previously retrieved or stored data. (2) * Pertaining to the process of obtaining data from, or placing data into, storage where the time required for such access is independent of the location of the data most recently obtained or placed in storage. (3) * Pertaining to a storage device in which the access time is effectively independent of the location of the data.

direct access storage device (DASD). A device in which the access time is effectively independent of the location of the data.

disk. (1) Loosely, a *disk storage device.* (2) See *magnetic disk.*

disk pack. A removable direct access storage volume containing magnetic disks on which data is stored. Disk packs are mounted on a disk storage drive, such as the IBM 2314 Disk Storage Drive.

disk storage. Storage on direct access devices that record data magnetically on rotating disks.

dismount. See *demount.*

* **display.** A visual presentation of data.

display line. On a display device, the series of character locations that constitute a horizontal line on the display surface. Synonymous with *character row.*

display position. On a display device, the series of character locations that constitute a vertical line on the display surface.

* **display tube.** A tube, usually a cathode ray tube, used to display data.

domain. A group of users whose characteristics are defined by the IPS and for which the SRM controls the multiprogramming level (i.e. the number of swappable incore users within the group). There may be 1 to 128 domains in the system.

* **drive.** See *tape drive.*

* **drum.** See *magnetic drum.*

drum storage. A direct access storage device that records data magnetically on a rotating cylinder. A type of addressable auxiliary storage associated with some computers.

* **dump.** (1) To copy the contents of all or part of a storage, usually from an internal storage into an external storage. (2) A process as in (1). (3) The data resulting from the process as in (1).

duplex channel. A channel providing simultaneous transmission in both directions.

duplexed system. In communications, a system with two distinct and separate sets of facilities, each of which is capable of assuming the system function while the other assumes a standby status. Usually, the sets are identical.

dynamic allocation. Assignment of system resources to a program at the time the program is executed rather than at the time it is loaded into main storage.

dynamic data set definition. The process of defining a data set and allocating auxiliary storage space for it during job step execution rather than before job step execution.

dynamic device reconfiguration (DDR). A facility that allows a demountable volume to be moved, and repositioned if necessary, without abnormally terminating the job or repeating the initial program load procedure.

* **dynamic dump.** A dump that is performed during the execution of a computer program.

* **dynamic storage allocation.** A storage allocation technique in which the location of computer programs and data is determined by criteria applied at the moment of need.

EBCDIC. *Extended binary coded decimal interchange code.*

end of block (EOB). A code that marks the end of a block of data.

end-of-file mark (EOF). A code which signals that the last record of a file has been read.

* **end of tape marker.** A marker on a magnetic tape used to indicate the end of the permissible recording area, for example, a photo-reflective strip, a transparent section of tape, or a particular bit pattern.

entering. The process in which a terminal places on the line a message to be transmitted to the computer. Contrast with *accepting*. See also *receiving, sending*.

environmental recording, editing, and printing. See *EREP*.

EOB. *End of block.*

EOF. *End of file mark.*

EOV. End of volume.

* **erase.** To obliterate information from a storage medium, for example, to clear, to overwrite.

EREP. The environmental recording, editing, and printing program that makes the data contained on the system recorder file available for further analysis.

error condition. The state that results from an attempt to execute instructions in a computer program that are invalid or that operate on invalid data.

* **error message.** An indication that an error has been detected.

execute (EXEC) statement. A job control language (JCL) statement that marks the beginning of a job step and identifies the program to be executed or the cataloged or in-stream procedure to be used.

execution batch processing. The concatenation of jobs or transactions that come from possibly different input sources and have possibly different output routings, and the execution of these jobs or transactions as if they were one continuous input stream. The purpose is to improve system performance by reducing the overhead of bringing different batch processors in and out of the system.

execution priority. A rank assigned to a task that determines its precedence in being selected for execution.

extended binary coded decimal interchange code (EBCDIC). A set of 256 characters, each represented by eight bits.

external interruption. An interruption caused by a signal from the interruption key on the system console panel, from the timer, or from another computing system.

external writer. In OS/VS2, a program that supports the ability to write SYSOUT data in ways and to devices not supported by the job entry subsystem.

facility. (1) A feature of an operating system, designed to serve a particular purpose, for example, the checkpoint/restart facility. (2) A measure of how easy it is to use a data processing system. Together with system performance, a major factor on which the total productivity of an installation depends. (3) Anything used or available for use in furnishing communication service. Commonly, a general term for communications paths.

* **field.** In a record, a specified area used for a particular category of data, for example, a group of card columns used to represent a wage rate or a set of bit locations in a computer word used to express the address of the operand.

fixed page. In System/370 virtual storage systems, a page in real storage that is not to be paged-out.

* **fixed storage.** Storage whose contents are not alterable by computer instructions, for example, magnetic core storage with a lockout feature, photographic disk. Synonymous with nonerasable storage, *permanent storage*, read-only storage.

* **flag.** (1) Any of various types of indicators used for identification, for example, a wordmark. (2) A character that signals the occurrence of some condition, such as the end of a word.

foreground. (1) In multiprogramming, the environment in which high-priority programs are executed. (2) Under TSO, the environment in which programs are swapped in and out of main storage to allow CPU time to be shared among terminal users. All command processor programs execute in the foreground. Contrast with *background*.

foreground job. (1) A high-priority job, usually a real-time job. (2) A teleprocessing or graphic display job that has an indefinite running time during which communication is established with one or more users at local or remote terminals. (3) Under TSO, any job executing in a swapped region of main storage, such as a command processor or a terminal user's program. Contrast with *background job*.

* **format.** The arrangement of data.

fullword. See *computer word*.

generalized trace facility (GTF). An optional OS/VS service program that records significant system events (such as supervisor calls and start I/O operations) for the purpose of problem determination. Abbreviated GTF.

general register. A register used for operations such as binary addition, subtraction, multiplication, and division. General registers are used primarily to compute and modify addresses in a program.

* **graphic.** A symbol produced by a process such as handwriting, drawing, or printing.

* **graphic character.** A character normally represented by a graphic.

Greenwich mean time (GMT). The mean solar time of the meridian of Greenwich used as the prime basis of standard time throughout the world. See also *TOD clock*.

GTF. *Generalized trace facility*.

* **halfword.** A contiguous sequence of bits or characters which comprises half a computer word and is capable of being addressed as a unit.

hard copy. A printed copy of machine output in a visually readable form; for example, printed reports, listings, documents, and summaries. See also *display (1)*.

hardcopy log. In systems with multiple console support or a graphic console, a permanent record of system activity.

* **hardware.** Physical equipment, as opposed to the computer program or method of use, for example, mechanical, magnetic, electrical, or electronic devices. Contrast with *software*.

HASP. An extension to the System/360 Operating System that provides supplementary job management, data management, and task management functions such as control of job flow, ordering of tasks, and spooling.

hexadecimal. Pertaining to a number system with a base of 16; valid digits range from 0 through F, where F represents the highest units position (15).

high local date. The highest valid local date that can be specified. (On most systems, 99.365, that is, December 31 1999.)

host-id. The unique ten-digit CPU identification made up of the six-digit CPU serial number followed by a four-digit model number.

IBM System/370. An upward-compatible extension of the IBM System/360, offering new function and improved price/performance.

* **identifier.** A symbol whose purpose is to identify, indicate or name a body of data.

independent mode. A means of isolating a processor for testing purposes. A processor so designated will only process jobs that are both routed to it and are themselves designated to execute on a processor in independent mode.

* **initialize.** To set counters, switches, and addresses to zero or other starting values at the beginning of, or at prescribed points in, a computer routine. Synonymous with *prestore*.

initial program load (IPL). The initialization procedure that causes an operating system to commence operation.

* **initial program loader (IPL).** The procedure that causes the initial part of an operating system or other program to be loaded such that the program can then proceed under its own control.

initiating task. The job management task that controls the selection of a job and the preparation of the steps of that job for execution.

initiator/terminator. The job scheduler function that selects jobs and job steps to be executed, allocates input/output devices for them, places them under task control, and at completion of the job, supplies control information for writing job output on a system output unit.

* **input.** (1) Pertaining to a device, process, or channel involved in the insertion of data or states, or to the data or states involved. (2) One, or a sequence of, input states. (3) Same as *input device*.

* **input device.** The device or collective set of devices used for conveying data into another device. Synonymous with *input (3)*.

input job queue. Same as *input work queue*.

input/output. (1) * Pertaining to either input or output, or both. (2) A general term for the equipment used to communicate with a computer, commonly called I/O. (3) The data involved in such communication. (4) The media carrying the data for input/output.

input/output interruption. See *I/O interruption*.

input queue. Same as *input work queue*.

input reader. See *reader (2)*.

input stream. The sequence of job control statements and data submitted to an operating system on an input unit especially activated for this purpose by the operator. Synonymous with input job stream, *job input stream*.

input stream control. Same as *JES reader*.

input work queue. In OS/360 and OS/VS, a queue (waiting list) of job definitions in direct access storage assigned to a job class and arranged in order of assigned priority. Synonymous with *input queue, input job queue, job queue*.

installation. A particular computing system, in terms of the work it does and the people who manage it, operate it, apply it to problems, service it, and use the results it produces.

* **instruction counter.** A counter that indicates the location of the next computer instruction to be interpreted.

internal reader. A facility that allows HASP, JES2, or JES3 to accept commands, control cards, and jobs from a program as though they were submitted through a card reader supported by HASP or JES.

interrupt. (1) * To stop a process in such a way that it can be resumed. (2) In data transmission, to take an action at a receiving station that causes the transmitting station to terminate a transmission.

interruption. A break in the normal sequence of instruction execution. It causes an automatic transfer to a preset storage location where appropriate action is taken. See also *external interruption, I/O interruption, machine check interruption, program check interruption, SVC interruption*.

* **I/O.** *Input/output*.

I/O interruption. An interruption caused by the termination of an I/O operation or by operator intervention at the I/O device.

IPL. (1) * *Initial program loader*. (2) *Initial program load*.

JCL. *Job control language*.

JES. *Job entry subsystem*.

JES reader. In OS/VS, the part of the job entry subsystem that controls the input stream and its associated job control statements. Synonymous with *input stream control*.

JES writer. In OS/VS, the part of the job entry subsystem that controls the output of specified data sets. Synonymous with *output stream control*.

job. (1) * A specified group of tasks prescribed as a unit of work for a computer. By extension, a job usually includes all necessary computer programs, linkages, files, and instructions to the operating system. (2) A collection of related problem programs, identified in the input stream by a JOB statement followed by one or more EXEC and DD statements. (3) See also *batched job, background job, foreground job*.

job class. Any one of a number of job categories that can be defined under an MFT or MVT control program configuration. By classifying jobs and directing initiator/terminators to initiate specific classes of jobs, it is possible to control the mixture of jobs that are performed concurrently.

job control language (JCL). A programming language used to code job control statements.

* **job control statement.** A statement in a job that is used in identifying the job or describing its requirements to the operating system.

job entry subsystem. In OS/VS, a system facility for spooling, job queueing, and managing the scheduler work area. Abbreviated JES.

job input device. A device assigned by the operator to read job definitions and any accompanying input data.

job input stream. Same as *input stream*.

jobname. The name assigned to a JOB statement; it identifies the job to the system.

job output device. A device assigned by the operator for common use in recording output data for a series of jobs.

job output element. Information that describes a unit of work for the HASP or JES2 output processor and represents that unit of work for queueing purposes.

job output stream. Same as *output stream*.

job pack area. Under MVT and OS/VS2, the two subpools in a region into which executable programs are loaded.

job priority. Under MVT and OS/VS2, a value assigned to a job that, together with an assigned job class, determines the priority to be used in scheduling the job and allocating resources to it.

job processing. The reading of job control statements and data from an input stream, the initiating of job steps defined in the statements, and the writing of system output messages.

job queue. Same as *input work queue*.

job (JOB) statement. The job control statement that identifies the beginning of a job. It contains such information as the name of the job, an account number, and the class and priority assigned to the job.

job step. (1) * The execution of a computer program explicitly identified by a job control statement. A job may specify that several job steps be executed. (2) A unit of work associated with one processing program or one cataloged procedure and related data. A job consists of one or more job steps.

job step initiation. The process of selecting job steps for execution and allocating input/output devices for them.

job step restart. Same as *step restart*.

job step task. A task that is initiated by an initiator/terminator in the job scheduler in accordance with specifications in an execute (EXEC) statement.

job stream. See *input stream, output stream*.

JOE. *Job output element*.

JPA. *Region job pack area*.

K. 1024 bytes; used in referring to storage capacity.

keyboard. A device for the encoding of data by key depression, which causes the generation of the selected code element.

keyword. A part of a command operand that consists of a specific character string (such as DSNAME=).

keyword parameter. A parameter that consists of a keyword, followed by one or more values. Contrast with *positional parameter*.

label. (1) * One or more characters used to identify a statement or an item of data in a computer program. (2) An identification record for a tape or disk file.

library. (1) * A collection of organized information used for study and reference. (2) * A collection of related files. For example, one line of an invoice may form an item, a complete invoice may form a record, the complete set of such records may form a file, the collection of inventory control files may form a library, and the libraries used by an organization are known as its data bank. (3) In OS/360 and OS/VS, any partitioned data set.

light pen attention. An interruption generated by a light pen when it senses light on the screen of a CRT display device. Synonymous with *selector pen attention*.

line. (1) On a terminal, one or more characters entered before a return to the first printing or display position. (2) A string of characters accepted by the system as a single block of input from a terminal, for example, all characters entered before a carriage return or all characters entered before the terminal user hits the attention key. (3) In communications, same as channel, circuit.

line number. (1) A number associated with a line in a printout or display. (2) In systems with time sharing, a number associated with a line in a line data set.

link pack area. In OS/VS2, an area of virtual storage containing reenterable routines that are loaded at IPL time and can be used concurrently by all tasks in the system. Abbreviated LPA.

listing. A printout, usually prepared by a language translator, that lists the source language statements and contents of a program.

* **load.** In programming, to enter data into storage or working registers.

local date. The local year and day of the year. See *high local date*. Contrast with *TOD clock*.

local system queue area (LSQA). In OS/VS2, one or more segments associated with each virtual storage region that contain job-related system control blocks.

* **location.** Any place in which data may be stored.

logical unit. The combination of programming and hardware of a teleprocessing subsystem that functions like a terminal to VTAM.

logon. The procedure by which a user begins a terminal session.

logoff. The procedure by which a user ends a terminal session.

* **loop.** A sequence of instructions that is executed repeatedly until a terminal condition prevails.

LPA. *Link pack area*.

LSQA. *Local system queue area*.

machine check interruption. An interruption that occurs as a result of an equipment malfunction or error. A machine check interrupt can be either hardware-recoverable, software-recoverable, or non-recoverable.

* **magnetic disk.** A flat circular plate with a magnetic surface on which data can be stored by selective magnetization of portions of the flat surface.

* **magnetic drum.** A right circular cylinder with a magnetic surface on which data can be stored by selective magnetization of portions of the curved surface.

* **magnetic tape.** A tape with a magnetic surface on which data can be stored by selective polarization of portions of the surface.

magnetic tape label. One or more records at the beginning of a magnetic tape that identifies and describes the data recorded on the tape and contains other information, such as the serial number of the tape reel.

* **main frame.** Same as *central processing unit*.

main storage. (1) * The general purpose storage of a computer. Usually, main storage can be accessed directly by the operating registers. Contrast with *auxiliary storage*. (2) All program-addressable storage from which instructions may be executed and from which data can be loaded directly into registers. (3) See also *real storage, virtual storage*.

master console. In a system with multiple consoles, the basic console used for communication between the operator and the system.

master scheduler. A control program routine that responds to operator commands and initiates the requested actions.

MCS. *Multiple console support*.

memory. * Same as *storage*.

message. (1) * An arbitrary amount of information whose beginning and end are defined or implied. (2) In telecommunications, a combination of characters and symbols transmitted from one point to another on a network. (3) * See *error message, operator message*.

message queue. A queue of messages that are awaiting processing or waiting to be sent to a terminal.

message routing. The process of selecting the correct circuit path for a message.

message text. The part of a message consisting of the actual information that is routed to a user at a terminal or to a program.

* **monitor.** Software or hardware that observes, supervises, controls or verifies the operations of a system.

MSS. Mass Storage System.

MSC. Mass Storage Control.

Multi-Access Spool complex. Two to seven systems sharing the JES2 input, job and output queues through the use of shared DASD.

Multi-Access Spool multiprocessing. Two or more computing systems interconnected by an I/O channel-to-channel adapter. The CPUs can be different types and have their own unique configurations.

multiple console support. A feature of OS/VS2 that permits selective message routing to up to 99 operator's consoles. Abbreviated MCS.

multiprocessing. (1) * Pertaining to the simultaneous execution of two or more computer programs or sequences of instructions by a computer network. (2) * Loosely, parallel processing. (3) Simultaneous execution of two or more sequences of instructions by a multiprocessor.

multiprocessing system. A computing system employing two or more interconnected processing units to execute programs simultaneously.

multiprocessor. (1) * A computer employing two or more processing units under integrated control. (2) A system consisting of two or more CPUs (or ALUs, or processors) that can communicate without manual intervention.

multiprogramming level. The number of swappable of incore users in a domain or in the system. The sum of the domain multiprogramming levels is the system multiprogramming level.

name. A 1-to-8 character alphameric term that identifies a data set, a control statement, or program, or a cataloged procedure. The first character of the name must be alphabetic.

network. In teleprocessing, a number of communication lines connecting a computer with remote terminals.

network operator. (1) The person responsible for controlling the operation of a telecommunication network. (2) A VTAM application program authorized to issue network operator commands.

NIP. *Nucleus initialization program.*

no-consoles condition. In systems with MCS, a condition in which the system is unable to access any full-capability console device.

node. An addressable point in a telecommunication system defined to VTAM by a symbolic name.

* **nonpageable dynamic area.** In OS/VS, an area of virtual storage whose virtual addresses are identical to real addresses; it is used for programs or parts of programs that are not to be paged during execution. Synonymous with *V = R dynamic area.*

nonpageable region. In OS/VS2, a subdivision of the nonpageable dynamic area that is allocated to a job step or system task that is not to be paged during execution. In a nonpageable region, each virtual address is identical to its real address. Synonymous with V=R region.

nonstandard labels. Labels that do not conform to American National Standard or IBM System/360 and System/370 standard label conventions.

nontemporary data set. A data set that exists after the job that created it terminates. Contrast with *temporary data set.*

nucleus. That portion of a control program that always remains in main storage.

nucleus initialization program (NIP). The program that initializes the resident control program; it allows the operator to request last minute changes to certain options specified during system generation.

* **null character.** A control character that serves to accomplish media fill or time fill, for example, in ASCII the all zeroes character (not numeric zero). Null characters may be inserted into or removed from a sequence of characters without affecting the meaning of the sequence, but control of equipment or the format may be affected. Abbreviated NUL. Contrast with *space character.*

* **number.** (1) A mathematical entity that may indicate quantity or amount of units. (2) Loosely, a numeral.

* **numeric.** Pertaining to numerals or to representation by means of numerals.

* **numeric character.** Same as *digit.*

* **offline.** Pertaining to equipment or devices not under control of the central processing unit.

* **offline storage.** Storage not under control of the central processing unit.

* **online.** (1) Pertaining to equipment or devices under control of the central processing unit. (2) Pertaining to a user's ability to interact with a computer.

* **online storage.** Storage under the control of the central processing unit.

operand. (1) * That which is operated upon. An operand is usually identified by an address part of an instruction. (2) Information entered with a command name to define the data on which a command processor operates and to control the execution of the command processor.

* **operating system (OS).** Software which controls the execution of computer programs and which may provide scheduling, debugging, input/output control, accounting, compilation, storage assignment, data management, and related services.

Operating System/Virtual Storage (OS/VS). A compatible extension of the IBM System/360 Operating System that supports relocation hardware and the extended control facilities of System/370.

operator. (1) * In the description of a process, that which indicates the action to be performed on operands. (2) * A person who operates a machine.

operator command. A statement to the control program, issued via a console device, or control terminal, that causes the control program to provide requested information, alter normal operations, initiate new operations, or terminate existing operations.

operator control station. Under TCAM, any station that is eligible to enter the operator commands.

operator message. A message from the operating system or a problem program directing the operator to perform a specific function, such as mounting a tape reel, or informing him of specific conditions within the system, such as an error condition.

OS/VS. Operating System/Virtual Storage.

* **output.** (1) Pertaining to a device, process, or channel involved in an output process, or to the data or states involved. (2) One, or a sequence of, output states. (3) Same as *output device. (4) Same as output process.*

output class. Under MFT, MVT, and OS/VS, any one of up to 36 different categories, defined at an installation, to which output data produced during a job step can be assigned. When an output writer is started, it can be directed to process from one to eight different classes of output data.

* **output device.** The device or collective set of devices used for conveying data out of another device. Synonymous with *output (3).*

* **output process.** The process of delivering data by a system, subsystem, or device. Synonymous with *output (4).*

output queue. See *output work queue.*

output stream. Diagnostic messages and other output data issued by an operating system or a processing program on output devices especially activated for this purpose by the operator. Synonymous with *job output stream.*

output stream control. Same as *JES writer.*

output work queue. A queue of control information describing system output data sets, that specifies to an output writer the location and disposition of system output.

output writer. A part of the job scheduler that transcribes specified output data sets onto a system output device independently of the program that produced the data sets.

page. (1) In virtual storage systems, a fixed-length block of instructions, data, or both, that can be transferred between real storage and external page storage. (2) To transfer instructions, data, or both, between real storage and external page storage.

pageable region. In OS/VS2, a subdivision of the pageable dynamic area that is allocated to a job or a system task that can be paged during execution. Synonymous with V=V region.

page data set. In System/370 virtual storage systems, a data set in external page storage in which pages are stored.

page fault. In System/370 virtual storage systems, a program interruption that occurs when a page that is marked "not in real storage" is referred to by an active page.

paging. In System/370 virtual storage systems, the process of transferring pages between real storage and external page storage.

paging device. In System/370 virtual storage systems, a direct access storage device on which pages (and possibly other data) are stored.

parameter. (1) * A variable that is given a constant value for a specific purpose or process. (2) See *keyword parameter, positional parameter.*

password. (1) A unique string of characters that a program, computer operator, or user must supply to meet security requirements before gaining access to data. (2) In systems with time sharing, a one-to-eight-character symbol that the user may be required to supply at the time the user logs on the system. The password is confidential, as opposed to the user identification.

path. In VTAM, the intervening nodes and lines connecting a terminal and an application program in the host CPU.

* **permanent storage.** Same as *fixed storage.*

physical unit. (1) The control unit or cluster controller of an SNA terminal. (2) The part of the control unit or cluster controller that fulfills the role of a physical unit as defined by systems network architecture.

PFK capability. On a display console, indicates that program function keys are supported and were specified at system generation.

positional parameter. A parameter that must appear in a specified location, relative to other parameters. Contrast with *keyword parameter.*

* **postmortem dump.** A static dump, used for debugging purposes, performed at the end of a machine run.

* **prestore.** Same as *initialize.*

* **preventive maintenance.** Maintenance specifically intended to prevent faults from occurring during subsequent operation.

printer. (1) A device that writes output data from a system on paper or other media. (2) * See *chain printer, train printer.*

priority. A rank assigned to a task that determines its precedence in receiving system resources. See also *job priority.*

private library. A user-owned library that is separate and distinct from the system library.

private volume. In OS/360 and OS/VS, a mounted volume that the system can allocate only to an output data set for which a specific volume request is made. A private volume is demounted after its last use in a job step. Contrast with *public volume*.

problem program. Any program that is executed when the central processing unit is in the problem state; that is, any program that does not contain privileged instructions. This includes IBM-distributed programs, such as language translators and service programs, as well as programs written by a user.

procedure. (1) * The course of action taken for the solution of a problem. (2) In a procedure oriented language, an independent, named block of statements that defines a specific portion of a program.

procedure library. A program library in direct access storage containing job definitions. The reader/interpreter can be directed to read and interpret a particular job definition by an execute statement in the input stream.

processing unit. See *central processing unit*.

processor storage. General purpose storage that is part of a central processing unit. Synonymous with *real storage*.

PROC statement. A job control statement used in cataloged or in-stream procedures. It can be used to assign default values for symbolic parameters contained in a procedure. For in-stream procedures, it is used to mark the beginning of the procedure.

* **program.** (1) A series of actions proposed in order to achieve a certain result. (2) Loosely, a routine. (3) To design, write and test a program as in (1). (4) Loosely, to write a routine. (5) See *computer program*.

program check interruption. An interruption caused by unusual conditions encountered in a program, such as incorrect operands.

program interruption. See *interruption, program check interruption*.

* **program library.** A collection of available computer programs and routines.

* **programmed check.** A check procedure designed by the programmer and implemented specifically as a part of his program.

program status word (PSW). A doubleword in main storage used to control the order in which instructions are executed, and to hold and indicate the status of the computing system in relation to a particular program.

PSA. Prefixed Save Area.

PSW. *Program status word*.

public volume. In OS/360 and OS/VS, a mounted volume that the system can allocate to an output data set for which a nonspecific volume request is made. A public volume remains mounted until the device on which it is mounted is required for another volume. Contrast with *private volume*.

* **punch.** A perforation, as in a punched card or paper tape.

queue. (1) A waiting line or list formed by items in a system waiting for service; for example, tasks to be performed or messages to be transmitted in a message switching system. (2) To arrange in, or form, a queue.

* **quiescing.** (1) The process of bringing a device or a system to a halt by rejection of new requests for work. (2) The process of bringing a system to a halt by rejection of new jobs.

* **range.** (1) The set of values that a quantity or function may assume. (2) The difference between the highest and lowest value that a quantity or function may assume.

* **read.** To acquire or interpret data from a storage device, a data medium, or any other source.

reader. (1) A device that converts information in one form of storage to information in another form of storage. (2) A part of the scheduler that reads an input stream into the system.

reader/interpreter. A part of job management that reads and interprets a series of job definitions from an input stream.

real address. In virtual storage systems, the address of a location in real storage.

real storage. (1) In System/370 virtual storage systems, the storage of a System/370 computing system from which the central processing unit can directly obtain instructions and data, and to which it can directly return results. (2) Same as *processor storage*.

receiving. The process by which a computer obtains a message from a line. Contrast with *sending*. See also *accepting, entering*.

recoverable ABEND. An error in which control is passed to a specified routine that allows continued execution of the program. Contrast with *unrecoverable ABEND*.

recoverable error. An error condition that allows continued execution of a program.

reel. A mounting for a roll of tape.

region job pack area (JPA). In OS/VS2, an area in a virtual storage region that contains modules that are not in the link pack area but are needed for the execution of jobs. Abbreviated JPA.

* **register.** A device capable of storing a specified amount of data such as one word.

remote device. A device attached to a remote station for sending input and receiving output.

remote job entry. Submission of job control statements and data from a remote terminal, causing the jobs described to be scheduled and executed as though encountered in the input stream.

* **remote station.** Data terminal equipment for communicating with a data processing system from a location that is time, space, or electrically distant.

remote terminal. An input/output control unit and one or more input/output devices attached to a system through a transmission control unit.

* **rerun.** A repeat of a machine run, usually because of a correction, an interruption, or a false start.

* **reset.** (1) To restore a storage device to a prescribed initial state, not necessarily that denoting zero. (2) To place a binary cell into the state denoting zero.

resident. Pertaining to a program that is permanently located in storage. For example, the nucleus in main storage or a system library on direct access storage.

resource. Any facility of the computing system or operating system required by a job or task, and including main storage, input/output devices, the central processing unit, data sets, and control or processing programs.

* **restart.** To reestablish the execution of a routine, using the data recorded at a checkpoint.

rewind. To return a magnetic or paper tape to its beginning.

* **routine.** An ordered set of instructions that may have some general or frequent use.

routing. The assignment of the communications path by which a message or telephone call will reach its destination.

routing code. A code assigned to an operator message and used, in systems with multiple console support (MCS), to route the message to the proper console.

* **scheduled maintenance.** Maintenance carried out in accordance with an established plan.

secondary console. In a system with multiple consoles, any console except the master console. The secondary console handles one or more assigned functions on the multiple console system.

secondary storage. Same as *auxiliary storage*.

security. Prevention of access to or use of data or programs without authorization.

* **selective dump.** A dump of one or more specified storage locations.

selector pen attention. Same as *light pen attention*.

sending. The process by which the central computer places a message on a line for transmission to a terminal. Contrast with *receiving*. See also *accepting, entering*.

sequence. (1) * An arrangement of items according to a specified set of rules. (2) In sorting, a group of records whose control fields are in ascending or descending order, according to the collating sequence.

* **sequential.** Pertaining to the occurrence of events in time sequence, with little or no simultaneity or overlap of events.

session. The period of time during which a user engages in a dialog with a conversational time sharing system; the elapsed time from when a terminal user logs on the system until he logs off the system; in VTAM, the period of time during which a node is connected to an application program.

* **set.** (1) A collection. (2) To place a storage device into a specified state, usually other than that denoting zero or space character. Contrast with *clear*.

setup. The preparation of a computing system to perform a job or job step. Setup is usually performed by an operator and often involves performing routine functions, such as mounting tape reels and loading card decks.

shared DASD option. An OS/360 and OS/VS option that enables independently operating computing systems to jointly use common data residing on shared direct access storage devices.

shared file. A direct access device that may be used by two systems at the same time; a shared file may link two systems.

shared main storage multiprocessing. A mode of operation in which two processing units have access to all of main storage.

* **sharing.** See *time sharing*.

SMF. *System management facilities*.

SNA. *Systems network architecture*.

* **software.** A set of programs, procedures, and possibly associated documentation concerned with the operation of a data processing system. For example, compilers, library routines, manuals, circuit diagrams. Contrast with *hardware*.

space. (1) * A site intended for the storage of data, for example, a site on a printed page or location in a storage medium. (2) * A basic unit of area, usually the size of a single character. (3) * One or more space characters. (4) * To advance the reading or display position according to a prescribed format, for example, to advance the printing or display position horizontally to the right or vertically down. Contrast with *backspace*.

* **space character.** A normally nonprinting graphic character used to separate words. The space character is also a format effector which controls the movement of the printing or display position, one position forward. The space character may also be considered in the hierarchy of information separators. Abbreviated SP. Synonymous with *blank character*. Contrast with *null character*.

* **special character.** A graphic character that is neither a letter, nor a digit, nor a space character.

spooled data set. A data set written on an auxiliary storage device.

spooling. The reading and writing of input and output streams on auxiliary storage devices, concurrently with job execution, in a format convenient for later processing or output operations.

SQA. *System queue area.*

standard job. A JES3 job that consists of input service, main service, output service, and purge performed in that order.

* **step.** (1) One operation in a computer routine. (2) To cause a computer to execute one operation. (3) See *job step.*

step restart. A restart that begins at the beginning of a job step. The restart may be automatic or deferred, where deferral involves resubmitting the job. Contrast with *checkpoint restart.*

* **storage.** (1) Pertaining to a device into which data can be entered, in which they can be held, and from which they can be retrieved at a later time. (2) Loosely, any device that can store data. See *auxiliary storage, fixed storage, main storage, permanent storage, real storage, virtual storage, volatile storage.*

* **storage allocation.** (1) The assignment of blocks of data to specified blocks of storage. (2) See *dynamic storage allocation.*

* **storage device.** A device into which data can be inserted, in which they can be retained, and from which they can be retrieved.

storage reconfiguration. A function that makes an area of defective storage unavailable and frees any system resources associated with it.

* **store.** (1) To enter data into a storage device. (2) To retain data in a storage device. (3) A storage device.

subsystem. A secondary or subordinate system, usually capable of operating independently of, or asynchronously with a controlling system.

subsystem-id. The three-digit address of a specific MSS device. These devices are part of the MSS subsystem and do not have a unit control block in the host system.

supervisor. The part of a control program that coordinates the use of resources and maintains the flow of CPU operations.

SVC interruption. An interruption caused by the execution of a supervisor call instruction, causing control to be passed to the supervisor.

swap. (1) In systems with time sharing, to write the main storage image of a job to auxiliary storage and read the image of another job into main storage. (2) Under OS/VS2 with TSO, to write the active pages of a job to external page storage and read pages of another job from external page storage into real storage.

swap data set. A data set dedicated to the swapping operation.

swapping. (1) In systems with time sharing, a process that writes a job's main storage image to auxiliary storage, and reads another job's main storage image into main storage. (2) Under OS/VS2 with TSO, a paging technique that writes the active pages of a job to external page storage and reads pages of another job from external page storage into real storage.

symbol. (1) * A representation of something by reason of relationship, association, or convention. (2) In OS/360 and OS/VS, any group of eight or less alphameric and national characters that begins with an alphabetic or national (#,@,$) character.

* **symbolic address.** An address expressed in symbols convenient to the computer programmer.

* **syntax.** (1) The structure of expressions in a language. (2) The rules governing the structure of a language.

SYSGEN. *System generation.*

SYSIN. A system input stream; also, the name used as the data definition name of a data set in the input stream.

SYSLOG. *System log.*

SYSOUT. A system output stream; also, an indicator used in data definition statements to signify that a data set is to be written on a system output unit.

* **system.** (1) An assembly of methods, procedures, or techniques united by regulated interaction to form an organized whole. (2) An organized collection of people, machines, and methods required to accomplish a set of specific functions.

system generation (SYSGEN). The process of using an operating system to assemble and link together all of the parts that constitute another operating system.

system input device. A device specified as the source of an input stream.

system library. A collection of data sets in which the various parts of an operating system are stored.

system log. A data set in which job-related information, operational data, descriptions of unusual occurrences, commands, and messages to or from the operator may be stored. Abbreviated SYSLOG.

system management facilities (SMF). An optional control program feature of OS/360 and OS/VS that provides the means for gathering and recording information that can be used to evaluate system usage.

system output device. A device assigned to record output data for a series of jobs.

system output writer. A job scheduler function that transcribes specified output data sets onto a system output unit, independently of the program that produced the data sets.

system programmer. (1) A programmer who plans, generates, maintains, extends, and controls the use of an operating system with the aim of improving the overall productivity of an installation. (2) A programmer who designs programming systems and other applications.

system queue area (SQA). In OS/VS, an area of virtual storage reserved for system-related control blocks.

system recovery machine checks (SR). Indicates that malfunctions were detected but have been successfully corrected or circumvented without loss of system integrity.

system residence volume. The volume on which the nucleus of the operating system and the highest-level index of the catalog are located.

system resource. Any facility of the computing system that may be allocated to a task.

system restart. (1) A restart that allows reuse of previously-initialized input and output work queues. Synonymous with *warm start*. (2) A restart that allows reuse of a previously-initialized link pack area. Synonymous with quick start.

system task. A control program function that is performed under the control of a task control block.

systems network architecture (SNA). The total description of the logical structure, formats, protocols, and operational sequences for transmitting information units through a communication system.

* **tape drive.** A device that moves tape past a head.

* **tape unit.** A device containing a tape drive, together with reading and writing heads and associated controls.

task. A unit of work for the central processing unit; therefore, the basic multiprogramming unit under the control program.

TCAM. *Telecommunications access method*.

telecommunications. (1) * Pertaining to the transmission of signals over long distances, such as by telegraph, radio, or television. (2) Data transmission between a computing system and remotely located devices via a unit that performs the necessary format conversion and controls the rate of transmission.

telecommunications access method (TCAM). A method used to transfer data between main storage and remote or local terminals. Application programs use either GET and PUT or READ and WRITE macro instructions to request the transfer of data, which is performed by a message control program. The message control program synchronizes the transfer, thus eliminating delays for terminal input/output operations.

teleprocessing. The processing of data that is received from or sent to remote locations by way of telecommunication lines.

temporary data set. A data set that is created and deleted in the same job. Contrast with *nontemporary data set*.

terminal. A device, usually equipped with a keyboard and some kind of display, capable of sending and receiving information over a communication channel.

terminal user. In systems with time sharing, anyone who is eligible to log on.

text. The data portion of a telecommunications message.

tightly-coupled multiprocessing. Two computing systems operating simultaneously under one control program while sharing resources.

time sharing. (1) * Pertaining to the interleaved use of the time of a device. (2) A method of using a computing system that allows a number of users to execute programs concurrently and to interact with the programs during execution.

TOD clock. A binary counter that measures the time elapsed at Greenwich, England, in microseconds, since 0 hours, 0 minutes, 0 seconds, January 1, 1900. See also *Greenwich mean time*.

trace. (1) The record of a series of events. (2) To record a series of events as they occur.

* **tracing routine.** A routine that provides a historical record of specified events in the execution of a program.

train printer. A printer in which the type slugs are assembled in a train that moves along a track. Contrast with *chain printer*.

unallocate. See *deallocate*.

* **unit.** (1) A device having a special function. (2) A basic element. (3) See *central processing unit, control unit*.

unit address. The three-character address of a particular device, specified at the time a system is installed; for example, 191 or 293.

universal character set (UCS). A printer feature that permits the use of a variety of character arrays.

unrecoverable ABEND. An error condition that results in abnormal termination of a program. Contrast with *recoverable ABEND*.

unrecoverable error. An error that results in abnormal termination of a program.

update. To modify a master file with current information according to a specified procedure.

user. Anyone who requires the services of a computing system. See also *terminal user*.

user exit. A point in an IBM-supplied program at which a user exit routine may be given control.

user exit routine. A routine written by a user to take control at a user exit of a program supplied by IBM.

USERID. *User identification*.

user identification (USERID). A one-to-eight-character symbol identifying a system user.

utility program. A problem program designed to perform an everyday task, such as transcribing data from one storage device to another.

* **variable.** A quantity that can assume any of a given set of values.

virtual address. In virtual storage systems, an address that refers to virtual storage and must, therefore, be translated into a real storage address when it is used.

virtual address space. In virtual storage systems, the virtual storage assigned to a job, terminal user, or system task.

virtual equals real (V = R) storage. Same as *nonpageable dynamic area*.

virtual region. In OS/VS2, a subdivision of the dynamic area that is allocated (in segment-size blocks) to a job step or a system task.

virtual storage. Addressable space that appears to the user as real storage, from which instructions and data are mapped into real storage locations. The size of virtual storage is limited by the addressing scheme of the computing system (or virtual machine) and by the amount of auxiliary storage available, rather than by the actual number of real storage locations.

virtual storage region. See *virtual region*.

virtual telecommunications access method (VTAM). A set of IBM programs that control communication between terminals and application programs running under DOS/VS, OS/VS1, and OS/VS2.

* **volatile storage.** A storage device in which stored data are lost when the applied power is removed, for example, an acoustic delay line.

volume. (1) That portion of a single unit of storage which is accessible to a single read/write mechanism, for example, a drum, a disk pack, or part of a disk storage module. (2) A recording medium that is mounted and demounted as a unit, for example, a reei of magnetic tape, a disk pack, a data cell.

volume serial number. A number in a volume label that is assigned when a volume is prepared for use in the system.

volume table of contents (VTOC). A table on a direct access volume, that describes each data set on the volume.

V = R dynamic area. Same as *nonpageable dynamic area*.

V = R storage. Same as *nonpageable dynamic area*.

VTAM. *Virtual Telecommunications Access Method*.

VTOC. *Volume table of contents*.

wait state. (1) The condition of a task that is dependent on one or more events in order to enter thr ready condition. (2) The condition of a central processing unit when all operations are suspended.

warm start. Same as *system restart (1)*.

warning message. An indication that a possible error has been detected. Contrast with *error message*.

work file. (1) In sorting, an intermediate file used for temporary storage of data between phases. (2) See also *work volume*.

work volume. A volume made available to the system to provide storage space for temporary fields or data sets at peak loads.

* **write.** To record data in a storage device or a data medium. The recording need not be permanent, such as the writing on a cathode ray tube display device.

writer. See *output writer*.

writing task. In OS/360, the job management task that controls the transfer of system messages and SYSOUT data sets from the direct access volume on which they were initially written to a specified output device.

Index

Operator's Library: OS/VS2
MVS System Commands
GC38-0229-1

READER'S COMMENT FORM

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. This form may be used to communicate your views about this publication. They will be sent to the author's department for whatever review and action, if any, is deemed appropriate.

IBM shall have the nonexclusive right, in its discretion, to use and distribute all submitted information, in any form, for any and all purposes, without obligation of any kind to the submitter. Your interest is appreciated.

Note: ***Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.***

Possible topics for comments are:

Clarity Accuracy Completeness Organization Coding Retrieval Legibility

If comments apply to a Selectable Unit, please provide the name of the Selectable Unit _____.

If you wish a reply, give your name and mailing address:

_____

_____

_____

Note: Staples can cause problems with automated mail sorting equipment. Please use pressure sensitive or other gummed tape to seal this form.

Cut or Fold Along Line

Please circle the description that most closely describes your occupation.

Customer	(Q) Install Mgr.	(U) System Consult.	(X) System Analyst	(Y) System Prog.	(Z) Applica. Prog.	(F) System Oper.	(I) I/O Oper.	(L) Term. Oper.	(O) Other

IBM	(S) System Eng.	(P) Prog. Sys. Rep.	(A) System Analyst	(B) System Prog.	(C) Applica. Prog.	(D) Dev. Prog.	(R) Comp. Prog.	(G) System Oper.	(J) I/O Oper.	(E) Ed. Dev. Rep.	(N) Cust. Eng.	(T) Tech. Staff Rep.

Number of latest Newsletter associated with this publication: _____

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments.)

GC38-0229-1

Reader's Comment Form

Cut or Fold Along Line

Fold and tape

Please Do Not Staple

Fold and tape

First Class
Permit 40
Armonk
New York

Business Reply Mail
No postage stamp necessary if mailed in the U.S.A.

Postage will be paid by:

International Business Machines Corporation
Department D58, Building 706-2
PO Box 390
Poughkeepsie, New York 12602

Fold and tape

Please Do Not Staple

Fold and tape

Operator's Library: OS/VS2 MVS System Commands (S370-40) Printed in U.S.A. GC38-0229-1

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, N.Y. 10604

IBM World Trade Americas/Far East Corporation
Town of Mount Pleasant, Route 9, North Tarrytown, N.Y., U.S.A. 10591

IBM World Trade Europe/Middle East/Africa Corporation
360 Hamilton Avenue, White Plains, N.Y., U.S.A. 10601