File No. S370-40
Order No. GC20-1806-9

Systems

IBM Virtual Machine Facility/370: Operator's Guide

Release 6 PLC 17

This publication is intended for those responsible for the operation and administration of a VM/370 system. It includes descriptions of all those commands that affect the I/O resources and operating characteristics of VM/370, the associated virtual machines, and the real hardware configuration. Also included is information on spooling, resource allocation, system startup and shutdown procedures, and VM/370 service programs.

COREQUISITE PUBLICATIONS

IBM Virtual Machine Facility/370:

System Messages, Order No. GC20-1808

OLTSEP and Error Recording Guide, Order No. GC20-1809

OS/VS Environmental Recording Editing and Printing (EREP) Program, Order No. GC28-0772

Tenth Edition (March 1979)

| This edition (GC20-1806-9) together with Technical Newsletters GN25-0491, dated August 1, 1979 and GN25-0834, dated April 1, 1981 applies to Release 6 PLC17 (Program Level Change) of the IBM Virtual Machine Facility/370 and to all subsequent releases until otherwise indicated in new editions or Technical Newsletters.

Technical changes and additions to text and illustrations are indicated by a vertical bar to the left of the change.

Changes are periodically made to the information contained herein; before using this publication in connection with the operation of IBM systems, consult the IBM System/370 Bibliography, Order No. GC20-0001, for the editions that are applicable and current.

It is possible that this material may contain reference to, or information about, IBM products (machines and programs), programming, or services which are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming, or services in your country.

Publications are not stocked at the address given below; requests for copies of IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for readers' comments is provided at the back of this | publication; if the form has been removed, comments may be addressed to | IBM Programming Publications, Dept. G60, P.O. Box 6, Endicott, New York, | U.S.A. 13760. IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

Preface

This book is for those persons responsible for the operation and administration of a VM/370 system.

This operator's guide describes some of the hardware of the System/370 as well as some user virtual machine requirements. The CMS Batch Facility is the only batch operation discussed in this publication. For general information on how to run batch jobs in VM/370 virtual machines (other than CMS), refer to the IBM Virtual Machine Facility/370 Operating Systems in a Virtual Machine, Order No. GC20-1821.

To manage VM/370 resources effectively, a system operator should be familiar with the operation and hardware requirements of System/370 input/output devices and should know the broad concepts of computer operations. In addition, the system operator should be familiar with the use and capability of the Interactive Problem Control System component of VM/370. Details of this major component are contained in the IBM Virtual Machine Facility/370 Interactive Problem Control System (IPCS) User's Guide, Order No. GC20-1823.

If your installation uses the VM/370 Remote Spooling Communications Subsystem (RSCS), the system operator is generally the operator of the RSCS virtual machine as well as of the real machine at the installation. For details on how to operate the RSCS virtual machine, refer to the IBM Virtual Machine Facility/370 Remote Spooling Communications Subsystem (RSCS) User's Guide, Order No. GC20-1816.

To help the system operator gain insight in virtual machine concepts and capabilities, refer to the IBM Virtual Machine Facility/370 Introduction, Order No. GC20-1800. This book briefly discusses virtual storage, virtual machines feature and facilities of VM/370.

This publication describes VM/370 system startup, logon, VM/370 commands, special service programs, user privilege classes, as well as other data necessary to manage the resources of the system on a daily basis.

There are five sections in this publication. The first section discusses the Operational control of VM/370 under normal and adverse conditions. The next four are the "how-to-do-it" sections. Wherever possible, subject matter adheres to an alphabetic arrangement. The appendix contains a summary of operator and user commands.

The following terms in this publication refer to the indicated support devices:

- "2305" refers to IBM 2305 Fixed Head Storage, Models 1 and 2.
- "270x" refers to IBM 2701, 2702, and 2703 Transmission Control Units or the Integrated Communications Adapter (ICA) on the System/370 Model 135.
- "3330" refers to the IBM 3330 Disk Storage, Models 1, 2, or 11; the IBM 3333 Disk Storage and Control, Models 1 or 11; and the 3350 Direct Access Storage operating in 3330/3333 Model 1 or 3330/3333 Model 11 compatibility mode.
- "3340" refers to the IBM 3340 Disk Storage, Models A2, B1, and B2, and the 3344 Direct Access Storage Model B2.
- "3350" refers to the IBM 3350 Direct Access Storage Models A2 and B2 in native mode.
- "3704", "3705", or "370x" refers to IBM 3704 and 3705 Communications Controllers.
- The term "3705" refers to the 3705 I and the 3705 II unless otherwise noted.
- "2741" refers to the IBM 2741 and the 3767, unless otherwise specified.
- "3800" refers to the IBM 3800 Printing Subsystem.
- "3850" refers to the IBM 3850 Mass Storage System.
- "3270" refers to a series of display devices, namely the IBM 3275, 3276, 3277, 3278 Display Stations. A specific device type is used only when a distinction is required between device types.

Information about display terminal usage also applies to the IBM 3138, 3148, and 3158 Display Consoles when used in display mode, unless otherwise noted.

Any information pertaining to the IBM 3284 or 3286 also pertains to the IBM 3287, 3288 and the 3289 printers, unless otherwise noted.

For the relationship of the VM/370 Operators Guide to other VM/370 prerequisite and corequisite publications, refer to Figure 1, The VM/370 Library.

PREREQUISITE PUBLICATIONS

IBM Virtual Machine Facility/370:

Introduction, Order No. GC20-1800

CMS User's Guide, Order No. GC20-1819

Terminal User's Guide, Order No. GC20-1810

If the 3767 Communucations Terminal is used as the alternate system console, the IBM 3767 Operator's Guide, Order No. GA18-2000, is also a prerequisite.

The VM/370 CMS User's Guide is a prerequisite only if the operator edits files used for system control or accounting purposes.

COREQUISITE PUBLICATIONS

IBM Virtual Machine Facility/370:

CMS Command and Macro Reference, Order No. GC20-1818

Planning and System Generation Guide, Order No. GC20-1801

CMS User's Guide, Order No. GC20-1819

CP Command Reference for General Users, Order No. GC20-1820

If your installation has a 3850 Mass Storage System, the following are corequisite publications:

IBM 3850 Mass Storage System (MSS) Introduction and Pre-installation Planning, Order No. GA32-0038

OS/VS Message Library: Mass Storage System (MSS) Messages, Order No. GC38-1000.

IBM 3850 Mass Storage System (MSS) Principles of Operation: Theory, Order No. GA32-0035.

IBM 3850 Mass Storage System (MSS) Principles of Operation: Reference, Order No. GA32-0036.

Operator's Library: IBM 3850 Mass Storage System (MSS) Under CS/VS, Order No. GC35-0014.

Information on commands that control the virtual machine and its resources applicable to the G privilege class user are contained in the VM/370 CP Command Reference for General Users.

Interactive Problem Control System (IPCS) User's Guide, Order No. GC20-1823

Operating Systems in a Virtual Machine, Order No. GC20-1821

System Programmer's Guide, Order No. GC20-1807

System Messages, Order No. GC20-1808

OLTSEP and Error Recording Guide, Order No. GC20-1809

This publication contains information on the VM/370 error recording process. It also describes briefly the use of CPEREP, the CMS command that uses OS/VS EREP for editing and printing SYS1.LOGREC and VM/370 error recording records. Detailed information on the use of the operands of the CPEREP command is contained in OS/VS Environmental Recording Editing and Printing (EREP) Program, Order No. GC28-0772.

Glossary and Master Index, Order No. GC20-1813

Remote Spooling Communicaticns Subsystem (RSCS) User's Guide, Order No. GC20-1816

The following are corequisite publications if your installation has a 3704 or 3705:

IBM 3704 Control Panel Guide, Order No. GA27-3086

IBM 3705 Control Panel Guide, Order No. GA27-3087

Note: References in the text to prerequisite and corequisite VM/370 publications will be given in abbreviated form.

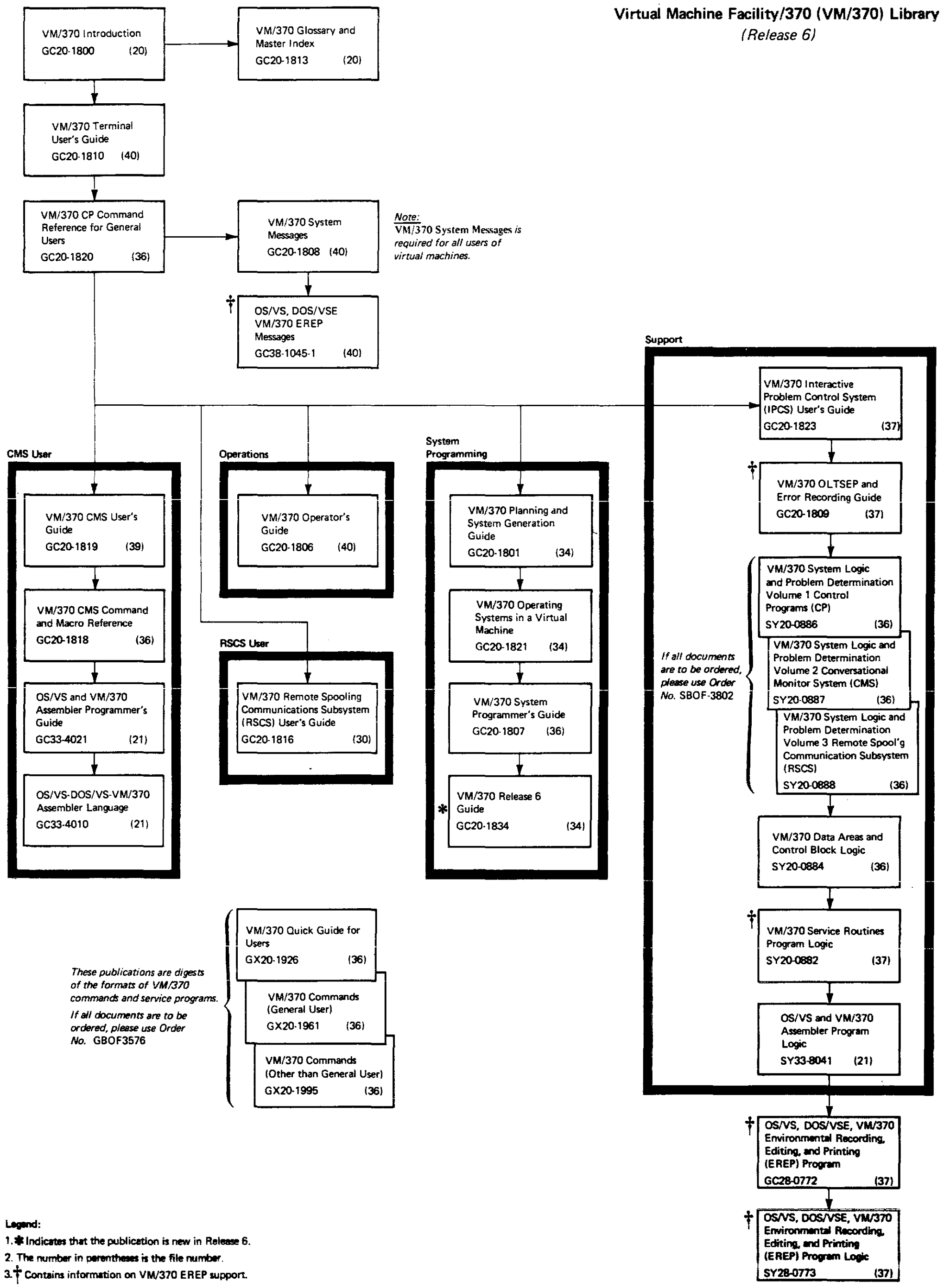

Virtual Machine Facility/370 (VM/370) Library
(Release 6)
VM/370 Introduction GC20-1800 (20)
VM/370 Glossary and Master Index GC20-1813 (20)
VM/370 Terminal User's Guide GC20-1810 (40)
VM/370 CP Command Reference for General Users GC20-1820 (36)
VM/370 System Messages GC20-1808 (40)
Note: VM/370 System Messages is required for all users of virtual machines.
† OS/VS, DOS/VSE VM/370 EREP Messages GC38-1045-1 (40)
CMS User
VM/370 CMS User's Guide GC20-1819 (39)
VM/370 CMS Command and Macro Reference GC20-1818 (36)
OS/VS and VM/370 Assembler Programmer's Guide GC33-4021 (21)
OS/VS-DOS/VS-VM/370 Assembler Language GC33-4010 (21)
Operations
VM/370 Operator's Guide GC20-1806 (40)
RSCS User
VM/370 Remote Spooling Communications Subsystem (RSCS) User's Guide GC20-1816 (30)
System Programming
VM/370 Planning and System Generation Guide GC20-1801 (34)
VM/370 Operating Systems in a Virtual Machine GC20-1821 (34)
VM/370 System Programmer's Guide GC20-1807 (36)
✱ VM/370 Release 6 Guide GC20-1834 (34)
Support
VM/370 Interactive Problem Control System (IPCS) User's Guide GC20-1823 (37)
† VM/370 OLTSEP and Error Recording Guide GC20-1809 (37)
VM/370 System Logic and Problem Determination Volume 1 Control Programs (CP) SY20-0886 (36)
VM/370 System Logic and Problem Determination Volume 2 Conversational Monitor System (CMS) SY20-0887 (36)
VM/370 System Logic and Problem Determination Volume 3 Remote Spool'g Communication Subsystem (RSCS) SY20-0888 (36)
If all documents are to be ordered, please use Order No. SBOF-3802
VM/370 Data Areas and Control Block Logic SY20-0884 (36)
† VM/370 Service Routines Program Logic SY20-0882 (37)
OS/VS and VM/370 Assembler Program Logic SY33-8041 (21)
† OS/VS, DOS/VSE, VM/370 Environmental Recording, Editing, and Printing (EREP) Program GC28-0772 (37)
† OS/VS, DOS/VSE, VM/370 Environmental Recording, Editing, and Printing (EREP) Program Logic SY28-0773 (37)
VM/370 Quick Guide for Users GX20-1926 (36)
VM/370 Commands (General User) GX20-1961 (36)
VM/370 Commands (Other than General User) GX20-1995 (36)
These publications are digests of the formats of VM/370 commands and service programs.
If all documents are to be ordered, please use Order No. GBOF3576
Legend:
1. ✱ Indicates that the publication is new in Release 6.
2. The number in parentheses is the file number.
3. † Contains information on VM/370 EREP support.

Figure 1. The VM/370 Library Publications

Contents

The entries in this Table of Contents are accumulative. They list additions to this publication by the following VM/370 System Control Program Products:

- VM/370 Basic System Extensions, Program Number 5748-XX8
- VM/370 System Extensions, Program Number 5748-XE1

However, the text within the publication is not accumulative; it only relates to the one SCP program product that is installed on your system. Therefore, there may be topics and references in this Table of Contents that are not contained in the body of this publication.

FIGURES

Summary of Amendments
for GC20-1806-9
as updated by GN25-0834
VM/370 Release 6 PLC 17

MISCELLANEOUS

Changed: Documentation

This Technical Newsletter incorporates minor technical and editorial changes.

Summary of Amendments
for GC20-1806-9
as updated by GN25-0491
VM/370 Release 6 PLC 4

CP DUMP SERVICES

New: Program Support

VM/370 provides processing support for the new class G command, VMDUMP. This command allows the user to dump an individual virtual machine's entire storage area. Users can process the VMDUMP command with the VM/IPCS Extension program product or a user written program.

VM/370 EXTENDED CONTROL-PROGRAM SUPPORT

Expanded: Program Support

Functions of the VM/370 Extended-Control Program Support (VM/370: ECPS) have been expanded to support 3031 attached processor systems. Users can activate the (VM/370: ECPS) virtual interval timer and extended virtual machine assist functions on an individual processor basis. The new processor address option is valid when issuing SET CPASSIST CP commands.

VM/370 SUPPORTS THE IBM 3800 PRINTING SUBSYSTEM

New: Hardware Support

VM/370 Release 6 supports the 3800 printing subsystem, a high speed, non-impact, general purpose printer that uses electrophotographic and laser technologies to achieve printing speeds up to 20,040 lines of copy per minute.

VM/370 allows virtual machines, including virtual VM/370, to attach the 3800 printer as a dedicated I/O device and a spooling device. Release 6 also supports the use of the 3800 with a starter system before performing a full system generation. In addition, the BACKSPAC, CHANGE, QUERY, SPOOL, and START CP commands have been modified to support the features of the 3800 printer. (See "Section 2: Initiating VM/370 Operation").

SUPPORT FOR THE IBM 3850 MASS STORAGE SYSTEM

New: Hardware and Programming Support

VM/370 supports user minidisks on Mass Storage System 3330V volumes, where the minidisk appears to the virtual machine to be defined on a permanently mounted IBM 3330-1 drive.

MSS volumes may also be dedicated and attached to a virtual machine as either 3330V or 3330-1 devices.

In addition, the new command DEFINE has been provided to the user, enabling the status of a 3330V volume to be redefined.

LOGON/LINK JOURNALING

New: Program Feature Option

VM/370 optionally supports the journaling of invalid passwords specified in the LOGON and AUTOLOG commands. The functions of the SET and QUERY commands have been expanded to support this option.

LOG SERVICE ROUTINE

New: Function and Documentation

The LOG control record allows the user to specify a fix number to a named module. The fix number is automatically recorded in a log file associated with the named module. (See "Section 4: VM/370 Service Programs".)

PASSWORD-ON-THE-COMMAND-LINE SUPPRESSION

New: Program Feature Option

VM/370 optionally supports passowrd suppression. This feature gives the installation the option to reject LINK and LOGON commands in which the password is entered from a virtual console on the same line as the command. The AUTOLOG command is modified to force the user to enter the password separately. Commands entered from an EXEC or CMS module are not affected.

MONITOR COMMAND

Changed: Function and Documentation

The functions of the MONITOR INTERVAL command have been expanded to give support to the high-frequency sampler.

The new MONITOR SEEKS options enable the user to select a subset of DASD devices for which data collection can be performed.

The MONITOR LIMIT option has been extended to allow specification for spool file closure after a number of samples have been collected.

MULTIPLE ALTERNATE CONSOLE SUPPORT

Changed: Function and Documentation

VM/370 now accepts multiple alternate console definitions at system generation. If the primary console is not operational during VM/370 IPL, the list of alternate consoles is searched in the order specified in the RIOGEN macro statement. The search is continued until an operational device is found. (See "Section 2: Initiating VM/370 Operation".)

MISCELLANEOUS CHANGES

Documentation Only

CMS Batch Facility information has been removed from this publication as this information is contained in the *VM/370 CMS User's Guide*, GC20-1819, a corequisite publication.

Directory Service Program documentation has been removed from this publication. Duplicate information is fully contained in the *VM/370 Planning and System Generation Guide*, GC20-1801, a corequisite publication.

Section 1. Introduction To the Operational Control of the VM/370 System

System operators, in almost all large System/370 installations, serve in an apprentice capacity until they have acquired the skill and knowledge to maintain the installation's operating system, be it DOS, OS, or VM/370.

In an apprentice capacity, an operator acquires a basic proficiency in mounting, making ready, and loading tape, DASD, line control, and other hardware devices. Novice operators also learn the address designations of all hardware devices attached to the system. They become aware of those system resources that can be switched or patched via alternate channel or telecommunication path to expedite system operations. The operator becomes aware of system console terminal operation as well as the function of most of the indicators, switches, buttons, and alarms that are part of the processor control. All of this knowledge is basic and must be acquired before considering the operational control of any system control program (SCP).

The second phase of an operator's training is to be knowledgeable about the SCP that is controlling the data processing operations; he must be aware of its capability and its operating philosophy. The operator must also be aware of the priorities and the demands placed upon the system.

Much of this basic knowledge of VM/370 is described in the _VM/370 Introduction_. In that document there is discussion on virtual machines; how they are built, identified, and used. Each facet of the virtual machine is described: the virtual processor, virtual storage, virtual I/O devices, and the virtual system console. Concurrent virtual machine usage and spooling operations are also described. The _VM/370 Introduction_ describes briefly the command language of CP and the other components of VM/370, namely, the Conversational Monitor System (CMS), the Interactive Problem Control System (IPCS), and the Remote Spooling Communications Subsystem (RSCS).

With an understanding of VM/370 concepts, the power and versatility of VM/370 is soon realized. The system operator is not just controlling one process and its related storage and I/O with control and application programs; the operator is exercising control over the resources of multiple systems (virtual systems). Each virtual system with its own system operation, in turn, is being controlled by a user via a terminal console. When additional facilities and resources or priorities are needed by the virtual system, the virtual system operator must request these services from the VM/370 system operator. The system operator then delegates additional system resources to the virtual machine, if feasible.

Some of the operations that are exclusive functions performed by the VM/370 system operator are as follows:

- Reorder, purge, or copy any closed spool files
- Issue warning and high priority messages
- Attach and dedicate devices to specified virtual machines
- Automatically logon virtual machines
- Force users off the system when warranted

- Change any virtual machine's dispatch priority and operating characteristics

For details on how many of these and other functions are accomplished, see "Section 3. CP Commands" in this publication.

Depending upon the system installation, the system operator may also be the operator of a large OS or DOS batch processing system that is run in the virtual machine environment. Another duty of the system operator is to control the data traffic from remote work stations. Information on VM/370 control of such data transfer is detailed in the _VM/370 Remote Spooling Communications Subsystem_ (_RSCS_) _User's Guide_.

Regardless of the real system resources and the tasks and procedures used, systems are subject to hardware and program malfunction. The Interactive Problem Control System of VM/370 provides a method of recording and maintaining a history of VM/370 and virtual machine program problems. The _VM/370 Interactive Problem Control System_ (_IPCS_) _User's Guide_ provides the details on how to use this system.

Problems that are a result of a hardware fault are usually detected by the hardware itself, and trigger the recording of register contents and sense values (related to the occurrence of the error) in VM/370's error recording area.

Regardless of whether the malfunction origin is hardware or software, the system operator (in most cases) is notified of the seriousness of the situations by means of console indicators/alarms or by console diagnostic messages. These all serve to inform the operator that:

- The system operation can continue as before.
- The system operation can continue with reduced power/resource.
- System restart and recovery is commencing.
- System operation is terminated.

The type of recovery that is attempted is described in "System/370 Recovery Management Support", which follows.

System/370 Recovery Management Support

IBM System/370 attempts correction of most machine errors without program assistance. CP is notified, via an interruption, of both intermittent and permanent machine errors to allow error recording and recovery procedures to start.

The following recovery features are implemented in the IBM System/370 hardware:

- Retry of the failing processor operations

- Validity checking on processor and control storage to correct all single-bit errors

- I/O operation retry facilities including an extended channel status word (ECSW), which provides channel retry data to channel and control unit retry procedures

- Expanded machine check interruption facilities tc improve error recording and recovery procedures

MACHINE CHECK HANDLER (MCH)

You can set the recording mode to record errors corrected by processor retry (logically termed as CPU retry) and Error Correction Code (ECC) with the SET MODE command. In attached processor applications, recording mode can be set for either or both processors. For processor retry, the default setting is record mode.

Note: The SET MODE MAIN command is invalid for 3031, 3032, and 3033 processors.

When processor retry or ECC succeed in correcting errors, and the processor is in record mode, the machine check handler records the error. When processor retry or ECC fail, the machine check handler:

- Attempts to isolate the failure to one page frame and makes that page frame invalid or unavailable for paging.
- Attempts to isolate the failure to one virtual machine and logs off or resets that virtual machine.
- Attempts to isolate the failure to portions of the system and to continue system operation in degraded mode.
- Abnormally terminates the system when recovery is not possible; or, if VM/370 is operating in attached processor mode and the malfunction is isolated to the attached processor and to a particular virtual machine, then, system operation continues in uniprocessor mode.

MCH records an error whenever any of the following conditions occur:

- Processor retry occurred[1].
- ECC corrected data[1].
- Hardware reported a buffer or DLAT (Data Look Aside Table) error.
- Multiple-bit storage failure.
- External damage.
- Storage protection feature damage.
- Timer error.
- System damage.
- Instruction processor damage.

CHANNEL CHECK HANDLER (CCH)

Whenever a channel control check, channel data check, or interface control check occurs, the channel check handler (CCH) constructs an error record and records the results in an IOERBLOK. The error recovery procedures use this IOERBLOK to retry the error. Recovery is not attempted for channel errors associated with virtual machine I/O events.

[1] VM/370 records these errors only under specific conditions. The conditions for recording these errors are detailed in the *VM/370 OLTSEP and Error Recording Guide*.

I/O Error Recording and SVC 76

VM/370 maintains an error recording area that captures I/O, CCH, and MCH error records. Device and control unit detected unit checks during VM/370 spooling, paging, and virtual machine I/O errors generate the I/O records.

VM/370 and the virtual machine's LOGREC data set contain recorded I/O errors; this double recording occurs when the virtual machine's operating system does not invoke SVC 76.

If the virtual machine operating system invokes SVC 76 and passes the correct parameters to VM/370, VM/370 records the error in its own error recording area. VM/370 then passes control back to the virtual machine operating system, thus bypassing virtual machine error recording facilities.

VM/370 Recovery Features

The VM/370 recovery features are described more fully in the _VM/370 OLTSEP and Error Recording Guide_.

RECORDING FACILITIES

The OS/VS Environmental Recording, Editing, and Printing program (EREP) is executed when the CMS CPEREP command is invoked. The output of the CPEREP command consists of printed reports whose content depends upon the specified (or defaulted) CPEREP operands and upon the input system error records. The reports generated by CPEREP have the same format as those generated on an OS/VS system. The input system error records may be from the VM/370 error recording area or from a history tape. The history tape may have been produced earlier by CPEREP from the VM/370 error recording area data or by an OS/VS system from SYS1.LOGREC data. Unlabeled tapes produced on OS/VS systems by OS/VS EREP and on VM/370 systems by CPEREP are compatible and can be transported between systems. Data from both systems can also be accumulated on the same tape. For more details on CPEREP, refer to the following publications: _VM/370 OLTSEP and Error Recording Guide_ and the _OS/VS Environmental Recording Editing and Printing (EREP) Program_.

If the facilities of an IBM 3850 Mass Storage System (MSS) are used with VM/370 virtual machine operations and MSS errors are reflected to VM/370's error recording area, CPEREP must be invoked so that MSS-related errors recorded in the error recording area can be collected on an accumulation (ACC=YES) tape for further processing by the VS System Data Analyzer Program (SDA). Because MSS logged-out data is voluminous and the interrelationships of MSS components are complex, it is imperative that this service program be used to effectively diagnose and isolate mass storage problems.

VM/370 Repair Facilities

The Online Test Standalone Executive Program (OLTSEP) and associated Online Tests (OLT) execute in a virtual machine that can run concurrently with normal system operations. These programs provide online diagnosis of I/O errors for most devices that connect to the System/370.

The service representative (with a CP command privilege class of F) can execute online tests from a terminal as a user of the system; CP console functions, including the ability to display or alter virtual machine storage, are available when these tests are run. Those tests that violate VM/370 restrictions may not run correctly in a virtual machine environment.

VM/370 Restart Facilities

Whenever a system failure causes an abnormal termination of the real machine that does not result in a disabled WAIT state, VM/370 attempts to reload CP. Often, the operator needs to take no action. The system attempts to execute a warm start, thus allowing user's terminals to be reconnected (for logon reinitialization by users) and completed spool files as well as open console spool files to be maintained. In the event of a warm start, device reconfiguration (such as varying a device offline) that was performed by the real computing system operator is remembered by CP for system spooling devices only. Storage reconfiguration data acquired during the process of reccvering from real storage errors is lost. After a VM/370 system failure, each user must re-access VM/370 (LOGON), and each virtual machine must be reloaded (IPLed).

Termination of a virtual machine, whether caused by a real computing system malfunction or a virtual machine program error, normally does not affect the execution of other virtual machines unless the error involves shared segments.

In VM/370 attached processor operations, the priorities and restart activities are similar to a uniprocessor mode of operation. However, if system damage assessment indicates a nonrecoverable operation on the attached processor and the error can be isolated to a virtual machine, VM/370 will be continued in uniprocessor mode on the main processor. All virtual machines with an AFFINITY setting are automatically reset to AFFINITY OFF for subsequent virtual machine processing. (Note, the affected virtual machines are placed in console function mode.) If a virtual machine was running when the malfunction occurred, its operation will be terminated.

Note: See the VARY PROCESSOR command in "Section 3. CP Commands" for instructions on how to vary the attached processcr back online.

If a similar malfunction occurred on the main processor while VM/370 was operating in attached processor mode, VM/370 terminates. VM/370 cannot switch to uniprocessor operation on the attached processor unit because the attached processor unit has no hardware interface for input/output capabilities.

Section 2. Initiating VM/370 Operation

Console Input Procedures

CP and CMS commands are never more than eight characters long and can be truncated to the minimum size indicated in the command format description. The operands, if any, follow the ccmmand on the same input data line. Most CP and CMS commands may not extend beyond one line except on the 3270. Generally, the operands are positional, but some commands have keywords to assist in the translation of the command line. One or more blanks must separate the command frcm any operands (except fcr some EDIT subcommands).

CMS indicates successful command processing with the CMS ready message. Error messages occur for erroneous operands. CP does not recognize CMS commands. However, a CMS user can issue CP commands without leaving the CMS environment. Although nct required, CP commands entered while in the CMS environment should be prefixed with "CP" or "#CP" to decrease VM/370 table and disk search time. CP then handles the command and passes control back to CMS. If the CMS user wants to enter CP mode, he can do so by keying in CP cr #CP with no operands (that is, no CP commands) or with an attention interruption. For details, refer to "Attention Handling" in the VM/370 Terminal User's Guide.

If a user requires communication with CP before responding to specific error messages issued by some virtual machine operating systems, he should enter CP mode by keying in #CP with one cr more CP command lines separated by a logical line end character, or by signalling attention; then perform the necessary console functions. When returning to the virtual machine environment, the virtual machine read will have been canceled by a unit exception, and the virtual operating system normally responds by reissuing the read. The user can then enter the required virtual machine response.

You, as the VM/370 system operator, are like other system users, but with some important differences:

- You are allowed to perform certain console functions that the normal user is not.
- Your terminal normally stays in CP mode and is usually ready to receive messages from the VM/370 system and other users.

Unless you are using a 3270 terminal, before you can type a command, you must press the REQUEST key (or its equivalent) on the designated VM/370 system console. VM/370 responds with the time, and sets up to read data from the console keyboard. You can then enter CP commands into the system.

You can enter all CP commands in lowercase or uppercase, or a mixture of both. All system responses are prefixed with the clock time at which you entered the command from the primary console. Full descriptions of CP commands with a class of A through F are in "Section 3. CP Commands."

Enter commands into the VM/370 system console by pressing the end-of-line function key to terminate the command line. To cancel a miskeyed or inappropriate VM/370 command that you partially or totally entered into the system, enter the designated logical line delete character (normally, the ¢ symbol) or press the CANCEL key (or its equivalent).

Note: The CANCEL key on the 3215 cannot cancel lines of input until you finish defining the VM/370 system during system generation. Also, the PA2 key or the CANCEL key on the 3270 and 3066 cannot cancel input lines.

For details on supported VM/370 typewriter terminal consoles, display terminal consoles and the associated keys that manage the display screen, signal attention, and process line records, consult the VM/370 Terminal User's Guide.

Operating Procedures

CP and CMS, with the exception of the special CP service programs and certain frequently used CP functions, get all of their system functions from disk. Each of the disks may or may not be labeled. If the disks are for CMS file residence, or CP paging and spooling (temporary use), you must label each pack with a six-character label in real cylinder 0 track 0 record 3 by using the CP Format/Allocate program.

The VM/370 system is device-independent; each of the volumes that have CP labels can reside on any available and defined 2314, 2319, 3330, 3333, 3340, 3344, 3350 or 2305 direct access storage device. This definition occurs at system generation time. The CP system residence (SYSRES) volume must be mounted on an available IPL device; then with the load switches dialed to that DASD device pressing the LOAD button will bring up the system. CP-owned volumes (cne each for paging and spooling) must be mounted and ready. All other resident volumes should be mounted and ready when the system is started, although they can be added later (via the CP ATTACH command) to the system, while it is in operation.

Assignment of Special Performance Options to Logged-on Users: (FAVORED, AFFINITY, SASSIST, CPASSIST, RESERVE, PRIORITY, LOCK, and UNLOCK commands): These can be assigned only by an operator with privilege class A.

Handling of Spooled Input and Output: Card decks for users must be fed into the real card reader, printed and punched output properly distributed, and the unit record equipment and spool data files controlled. The control functions for the system's unit record equipment and spool data files can be performed only by a class D operator.

Attaching and Detaching of User and System Volumes: Devices used by virtual machines in dedicated mode must be attached and detached as appropriate. Control over the real System/370 computing system's I/O can be performed by an operator with privilege class B.

Operators with these privilege classes must be logged on to perform these functions. Below is a directory entry for an operator to perform all of these functions:

```
USER OPERATOR OPASS 512 1M ABDG
```

The system and spooling operators do not require virtual devices or options in their VM/370 directory entries unless CMS is used. Multiple virtual machines for operators may be set up, each with all or some of the associated privilege classes. The primary system operator must have class A assigned in the directory to properly initiate VM/370 operation.

SYSTEM DEFINITION

If you intend to define or attach more than 64 virtual devices for a single virtual machine, be aware that any single request for free storage in excess of 512 doublewords (a full page) may cause the VM/370 system to issue an appropriate error message if the extra storage is not available on a contiguous page. Therefore, two contiguous page frames of free storage must be available in order to log on to a virtual machine with more than 64 virtual devices (three contiguous page frames for a virtual machine with more than 128 virtual devices, etc.). Contiguous page frames of free storage are sure to be available only immediately after IPL, before other virtual machines have logged on. Therefore, as a system operator you can inhibit the logon of many virtual machines until the virtual machine or machines with many I/O devices have accomplished logon and are operating.

SYSTEM INITIALIZATION

Successful VM/370 system initialization starts with the initial program loading of the CP system residence (SYSRES) pack. Whether performing a cold, warm, or checkpoint start, the following conditions must be met:

- The processor, processor storage and all necessary devices (and disk volumes) for a minimum VM/370 system, system residence, paging, and spooling requirements must be powered up and ready.

- No unrecoverable catastrophic hardware errors are encountered in the initialization and startup phase of bringing the VM/370 system online.

- There must be a matching compatibility between the generated VM/370 SCP and the System/370 hardware and its attached features. Failure of matching compatibility can result in an error message as well as VM/370 performance degradation (for example, refer to the VM/370 System Messages and the message DMKCPI185E).

System startup involves CP initialization; checkpoint functions; and subsequent warm start, checkpoint, or cold start procedures. When you turn system power on, you may have to run a CS (control storage) patch routine. This is a System/370 operating procedure and this book discusses it no further.

System initialization establishes and maintains the integrity of VM/370 even as it is being started. Checkpoint operations provide continuity for certain CP functions that are carried over from a previous VM/370 session. Checkpoint is invoked at initialization, shutdown, and after a system dump. Warm start procedures allow the operator to bring the VM/370 system up after a previous abnormal termination with certain controls saved from the previous system's session that will be used after the current system initialization is complete. The operator begins cold start if previously captured VM/370 session data or controls were destroyed or are invalid.

To initiate the operation of the VM/370 system, load the CP system residence volume.

System initialization is simplified if:

1. All VM/370 resident volumes (specified in the SYSOWN list for paging and spooling) are mounted and ready at IPL time. If volumes specified in SYSOWN are not mounted when ycu IPL VM/370, they are not available to you. You must mount and attach them later if you need them by using the ATTACH command.

2. Volumes containing user minidisks (such as the CMS system residence volume) are mounted and ready at IPL time. However, volumes not required for startup can be attached to the system after the VM/370 program is up and operational.

Once loaded, CP reads the volume identity labels of all available DASD devices, and calculates the real machine's storage size. If the configuration differs from that specified during system generation (SYSOWN volumes not mounted or storage size nct equal to SYSCOR), a message is printed at the system console and operation continues. If a real machine has more storage than what was specified on SYSCOR, it is not used. For example: if a real machine's storage size is 1 megabyte and SYSCOR is set to 512K, 512K is used. If less storage is actually available in the real machine than the specified SYSCOR value, the lesser figure is used.

If the SYSCOR macro had specified AP=YES and the attached processor is unavailable for startup, the system operates in uniprocessor mode rather than attached processor mode.

During the system startup process, the userid designated in DMKSYS is automatically logged on as the primary system operator; he must have class A command privileges. This process matches the userid supplied by the SYSOPR macro (used in system generation) with the user directory. If there is an incompatibility between userids, then the automatic operator logon process is aborted and system operational control is given to the first user with privilege class A who logs onto the system.

In the case of an automatic logon failure when a privilege class A user explicitly logs onto the system, where the system console is either a 3210 or 3215 (or a 3138, 3148, or 3158 console in printer keyboard mode), the user must protect the security of his password as there are no print-masking or print-inhibiting features on these devices.

After the operator has logged on, VM/370 prompts him for the type of system start desired (whether this is a warm start or not).

At this point, the system is ready for normal use. The operator can now:

- Establish a message of the day to be issued to users when they log on.

- Enable communications lines to permit users to log on.

Time-of-Day Clock

The Time-of-Day (TOD) clock provides an accurate measure of time, independent of system events or activities, and makes accurate measurements available for programming applications. When system power is turned off, the clock value is lost on some System/370 models. Once the time-of-day (TOD) clock has been made operational through the SET CLOCK instructions and the TOD ENABLE SET switch, it is incremented at a constant rate. This timing operation is not affected by:

- Any normal activity or event in the system
- Wait state
- Stopped state
- Instruction-step mode
- Single-cycle mode
- Test mode[1]
- System reset
- Initial program load procedure

If TOD clock hardware errors occur, the System/370 hardware posts a machine check and CP enters a disabled WAIT state.

The current clock value can be obtained by using the System/370 STORE CLOCK instruction, which causes the current clock value to be stored in the storage location specified in the instruction. Thus, virtual machines can use the value of the TOD clock for any purpose that their users desire. However, virtual machine users cannot set or modify the TOD clock. VM/370 uses the TOD clock for establishing certain accounting records for the virtual machines as well as queueing and prioritizing tasks for CP and virtual machine operations.

The privileged System/370 instruction, SET CLCCK, sets the clock to a specific value. If certain operating or program conditions exist, CP executes this instruction when the system is initialized. When this instruction executes, it replaces the current clock value by the value specified in the instruction. The SET CLOCK instruction changes the clock value only when the TOD ENABLE SET switch is enabled.

If the installation is configured with an attached processor, it may be necessary in the initialization procedure to synchronize the TOD clocks. When this occurs, you may be requested to ready the "TOD ENABLE SET" key twice; once to set the clock on the main processor and once to "SET and SYNC" the clock on the attached processor. If the clock value has previously been established for the main processor, then only one prompt message will be issued. Note, that the TOD Enable Set key is or'd to the other processors' TOD Enable SET key. This permits the setting of the clock from either processor. If the low-order 32 bits of the TOD clock are out of synchronization, a TOD Sync check will be received. Again, you will be requested to synchronize the TOD clocks by pressing the TOD Enable Set key.

Details on those segments of VM/370 initialization that are unique only to attached processor applications are detailed at the end of the System Initialization topic.

Setting the Clock (Uniprocessor Application)

If the clock needs to be set, CP prompts you to enter the date by issuing the message:

```
SET DATE MM/DD/YY:
```

You should then enter the date. CP then requests the time:

```
SET TIME HH:MM:SS:
```

You should then enter the time.

[1] The current clock value is lost when the time-of-day microdiagnostic tests are being run, or when system power is turned off.

Notes:

1. Enter the required data immediately following the colon (:). If you are using a 3270 or a VM/370 supported CPU display console, enter the information in the user input area. The information then appears in the output display area on the line following the SET TIME request.

2. You should enter a time value that is some 30 to 60 seconds ahead of the current time, because CP requests you to press the TOD ENABLE SET switch when setting the clock.

For example, if the date is 03/01/74, and the time will soon be 08:28 (EST), the exchange on a printing terminal looks like this:

```
VM/370 VERSION 1 LEVEL 0 mm/dd/yy
SET DATE MM/DD/YY :03/01/74
SET TIME HH:MM:SS :08:28:00
PRESS "TOD ENABLE SET" KEY AT DESIGNATED INSTANT
```

When you press the TOD ENABLE SET key, CP responds with:

```
NOW 08:28:00 EST FRIDAY 03/01/74
CHANGE TOD CLOCK (YES|NO):
```

Note: The previous example executed on a display terminal would show your responses on separate lines.

You now have an opportunity to change the value of the clock. If you enter "no", or press the END function key on the console, clock initialization ends. If you enter "yes", another series of prompting messages permit you to enter a new date and time.

Programming Note: To VM/370 users, the epoch clock value of all zeros indicates the beginning of the day January 1, 1900. To OS and OS/VS users, the epoch clock value of all zeros represents the beginning of the day January 1, 1960. Thus, OS virtual machine users will find that stored clock values differ by 60 years; however, it is possible in OS to use 1900 as the base epoch clock value.

Cold Start

Cold start procedures are usually invoked only on the initial loading operation of a new version of the VM/370 system or if all closed spool files as well as accounting records are lost or rendered unreadable because of a hardware error that prevented valid system recovery and continuation.

Failures in system component power, building power, hardware logic, the CP program, or some (rare) kinds of user programs can be classified as system operating procedure errors; all can require that you start the system with a cold start. Some of these malfunctions may require you to issue the SHUTDOWN command. If the SHUTDOWN command cannot be keyed in, you can press the LOAD button on the System/370 console. Then after the appearance of the shutdown message, you should press the LOAD button a second time to reload the system with VM/370. VM/370 issues a message to you to perform cold start procedures when it recognizes that unrecoverable errors have occurred.

VM/370 cold start procedures can start after the condition that caused the System/370 or the operator to abnormally terminate the VM/370 system has been corrected.

If a cold start is required, start it by responding with "cold" to the following message.

```
    hh:mm:ss  START ((COLD|WARM|CKPT|FORCE)(DRAIN))|(SHUTDOWN):cold
```

A good practice after an abnormal shutdown and cold start procedure is for the system operator to incorporate into the LOGMSG the time of the unplanned shutdown and that a cold start was done. This tells the user that his spooled files were lost, and that reconstruction of some of his other files may be in order.

Note: A warm start is the normal procedure used in recovering accounting data and in order to access previously closed spooled output files. CP informs you when a warm start cannot be done and requests an alternate recovery method.

Warm Start

Warm start implies that the previous VM/370 session ended with an orderly shutdown procedure. A warm start should be attempted after VM/370 has abnormally terminated, and after a successful system dump operation. Begin warm start procedures if you are not alerted by console messages or visual indications that a checkpoint or cold start is required. Warm start restores previous spool controls and files; accounting records remain valid and are to be kept as a base for continuing operations. In many cases, VM/370 recovers from system failures and restarts itself. When this occurs, the following message occurs at the console:

```
    VM/370 SYSTEM RESTART
```

To begin warm start procedures, key in "warm" or simply press the END, ENTER, or RETURN key (depending on the console device) when the following message appears at the console:

```
    hh:mm:ss  START ((COLD|WARM|CKPT|FORCE)(DRAIN))|(SHUTDOWN):
```

(If you want to free up all unit record devices, specify "drain". You can also specify "warm-drain" to warm start the system and drain all unit record devices, or "shutdown" to close down the system.) The warm start then proceeds as follows:

System Initialization Begins: Portions of the VM/370 control program (CP) enter real storage and CP then checks the status of the time of day (TOD) clock. If the clock is not set, CP requests the operator to set it, as previously described. If the TOD clock is error-free, CP examines each DASD to ensure that the volume labels on all DASD packs for VM/370 system use are correct. CP determines if the system residence (SYSRES) volume was defined and formatted during VM/370 system generation. CP also determines if the SYSRES volume resides on a device defined as a component of the real I/O configuration. Label and SYSRES volume residence errors cause error message printing on the system console, and the system startup may abnormally terminate. However, if CP finds no errors, it begins its recovery activity that is an investigation of the integrity of the previously stored spooling and accounting data.

CP examines the queues and chains of user spool files and locates where they reside on the spool DASD volumes. If VM/370 detects a spooling error, a console message alerts the system operator.

CP checks accounting data for irregularities. If any are found, CP displays a message on the system console.

During VM/370 system generation, the installation designates a console or terminal as the primary operator's system console. CP now attempts to automatically log on the primary system operator on that console or terminal. Multiple alternate consoles can be specified in the RIOGEN macro instruction during system generation. If the designated VM/370 system console is turned off or not operational, CP automatically attempts to log on the operator on the first designated alternate console. If operational, this console is used as the VM/370 primary system console. If not operational and multiple alternate consoles were specified during system generation, the next alternate console is accessed. The primary and alternate consoles must be among the following devices:

- A real primary or alternate system console
- A locally attached 3270

In addition, the alternate console can be any VM/370 supported remote terminal on a leased line connected to a 270x transmission control unit. Any terminals on switched lines, and leased line terminals connected to 3704/3705s are not supported as primary or alternate VM/370 consoles. If the alternate devices do not exist, or are not ready, automatic logon ceases and the VM/370 system enters the disabled wait state. If the primary or alternate consoles are operational, but some processing error occurs, manual operator intervention is necessary to bring up the VM/370 system. CP considers the first user to log on VM/370 with privilege class A as the primary system operator. No other users are accepted on the system until the primary system operator has successfully logged on. The system operator's userid, password, and command privilege class or classes are usually stored on the CP system residence volume during system installation.

Checkpoint Start

If the system could not perform a warm start because of I/O errors or invalid data in the warm start area, you can request a checkpoint (CKPT) start. This option attempts to initialize the system using the information that has been dynamically checkpointed during system operation and stored in the checkpoint area. Spool file chains are reconstructed; however, the original order of the files is lost. Accounting and system message data, recovered under a warm start, is lost under a checkpoint start. Also, because each spocl buffer must be read to reconstruct the record allocation blocks, a checkpoint start takes longer than a warm start.

Force Start

If the system cannot perform a checkpoint start because of I/O errors or invalid data in the checkpoint area, you can request a force (FORCE) start. This option operates like the CKPT start except that any spool file that contains unreadable or invalid data is dropped from the system.

When the FORCE option is used, you should not allow any users to log on to the system. The system should be devoted to reclaiming and putting out as many spool files as possible. When this has been accomplished, the system should be re-initialized using the cold (COLD) start option.

Automatic Logon

CP automatically logs on the primary system operator as soon as CP polls the designated system console and finds it ready. Upon successful completion of the IPL and system operator logon, the following message appears:

```
hh:mm:ss LOGMSG- hh:mm:ss EST FRIDAY mm/dd/yy
```

It indicates the date and the time of the last LOGMSG setting. The following message then appears and provides a general map of storage allocations.

```
DMKCPI957I STORsssssk,NUCnnnk,DYNdddddk,TRAtttk,FREEffffk,V=Rvvvvvk
```

Next, the following message is displayed only if queued closed spool files exist.

```
hh:mm:ss FILES: xxx RDR xxx PRT xxx PUN
```

The VM/370 system console is now the only enabled terminal on the system and you can start the system.

Manual Logon

If the automatic logon of the primary system operator fails because the VM/370 directory defines no system operator, the following message occurs:

```
DMKCPI950A AUTO LOGON FAILED; EXPLICIT LOGON REQUIRED
```

You should respond by logging on with your userid:

```
logon userid
```

In the above response, you enter the CP LOGON command, followed by a blank and your identifier name (userid), followed by pressing the end function key. CP checks this entry with the values in the VM/370 directory. If the userid is valid and you have class A command privileges, the following message on the console prompts you for the next response:

```
hh:mm:ss ENTER PASSWORD
```

You then key in your password, which is checked against the password stored in your VM/370 directory entry. The system is not accessible to other users until you log on and enable the necessary teleprocessing lines. Maintaining password security on real printer-keyboard system consoles is your responsibility because password inhibiting or masking is not possible on these devices.

If the password is valid, the VM/370 system can be started.

Unit Record Spool Files

After starting the VM/370 system, the system responds with the status of the system spool files:

```
hh:mm:ss FILES:  031 RDR, 039 PRT, 001 PUN
```

The response indicates that a total of 31 spooled reader files, 39 spooled print files (which also includes closed spooled console files), and 1 spooled punch file were saved at the last system shutdown.

At this point, if 3705 control units are part of the VM/370 system support of remote terminals and the automatic load function for the 3705 was not generated into the VM/370 system, you invoke the NETWORK LOAD command as follows:

```
NETWORK LOAD raddr ncpname
```

where raddr is the base address of the 3705 and ncpname is the desired copy of the 3705 program. For a description of the NETWORK command, see "Operator Commands."

You can then enable any or all of the teleprocessing lines and 3705 resources for VM/370 users by issuing the CP ENABLE command and NETWORK ENABLE command, respectively. For example, pressing the REQUEST function key and entering:

```
hh:mm:ss enable 50
```

enables only line 50, while:

```
hh:mm:ss en all
```

enables all 3270 and 270x lines. CP responds with:

```
hh:mm:ss COMMAND COMPLETE
```

when the requested command has been serviced.

Note: The same message occurs even if no lines connect to the machine. If a patch panel is in use, make sure that all desired lines connect to the system.

The VM/370 system is now operational and waits for users to log on. Similarly, if:

```
network enable all
```

is invoked, the total resources of all VM/370 controlled 3705 control units are enabled if the system is equipped with the devices.

If you performed a system cold start or drained unit record devices, you should start the unit record devices at this time. For example, the CP command

```
start all
```

starts all real unit record equipment. The system then responds with the status of each unit record device.

Mass Storage System Initialization

If an MSS port is attached to a virtual machine running OS/VS with MSS support and the MSS communicator program is running in the virtual machine, the control program can cause automatic 3330V volume mount and demount in response to any of the following:

- A virtual machine logs on with a minidisk defined on a system disk that is not mounted. The control program attempts to mount an MSS volume with the correct volume label.
- A virtual machine logs on having a dedicate directory statement specifying a volid for a volume that is not mounted. The control program again attempts to mount an MSS volume with the specified volume label.
- An operator ATTACH command is issued specifying a device address that is an MSS 3330V address and volid. The control program attempts to mount the volid on the device address as part of the ATTACH processing.
- The operator DEFINE command is used to change the feature (SYSVIRT or VIRTUAL) of a 3330V containing a mounted volume. The control program demounts the volume before completing the DEFINE process.
- A virtual machine issues a LINK command to link to a virtual device that is all or part of a 3330V volume. The control program attempts to mount the volume as part of the LINK process.

The VM/370 control program does not issue orders directly to the mass storage control (MSC). Rather, it passes requests to an OS/VS system with MSS support that is operating in a virtual machine with an MSC port dedicated to it. The OS/VS system then issues orders to the MSC and passes response information back to the control program.

In order for the installation to run with full MSS support, an OS/VS system must be IPLed in a virtual machine. An MSC port must be dedicated to this virtual machine. There are no special requirements for this IPL. The standard installation procedures for IPLing OS/VS in a virtual machine should be followed. After the system is IPLed, the VM/370 communicator program must be started. If the standard installation procedures as documented in the _VM/370 Planning and System Generation Guide_ have been followed, the communicator program is started in the virtual machine through the OS/VS operator command:

```
start dmkmss [.pn]
```

The [.pn] operand is used only in OS/VS1 systems to specify the partition in which the program is to be started.

The absence of error messages received from DMKMSS indicates that MSS support is initialized. The communicator virtual machine may now be disconnected if the installation desires.

There are five error messages that the DMKMSS program can produce. If any of these messages are received, then the VM/370 control program will not be able to communicate with the MSS. The messages are as follows:

DMKMSS ENDING ERROR - MSS NOT INITIALIZED

Explanation: The MSS did not get initialized as it should during the OS/VS IPL process. Either the virtual machine does not have access to the MSC, or there was an MSS error.

Operator response: Correct the problem, use the OS/VS VARY command to initialize the MSS, then reissue the "start dmkmss" command.

DMKMSS ENDING ERROR - DCB FOR COMM. DEVICE NOT OPENED

Explanation: The DMKMSS program was not able to establish communication with the VM/370 control program because the OS/VS DCB control block could not be opened. The probable cause is the definition of the communicator device for the virtual machine.

Operator response: Contact the system programmer. After the problem has been corrected, reissue the "start dmkmss" command.

DMKMSS ERROR READING MSS CONFIGURATION TABLE

Explanation: DMKMSS attempted to read one of the MSS configuration tables (Configuration Index/Manufacturer's ID Conversion table or the Configuration Data Block) and an error was encountered.

Operator response: Contact the MSS system programmer. After the problem has been corrected, reissue the START DMKMSS command.

DMKMSS CPUID NOT FOUND IN MSS CONFIGURATION TABLES

Explanation: DMKMSS has read one of the MSS configuration tables (the Configuration Index/Manufacturer's ID Conversion table) and searched it for a match to this cpuid, but no match was found.

Operator response: Contact the MSS system programmer. The MSS configuration tables may need to be updated. After the problem has been corrected, reissue the START DMKMSS command.

DMKMSS ERROR CREATING MSS CONFIGURATION TABLE

Explanation: DMKMSS has given CP a list of VUAs associated with the central server cpuid. CP has attempted to create its shared and nonshared SDG tables, but has encountered an error.

Operator response: Contact the system programmer. After the problem has been corrected, reissue the START DMKMSS command.

Attached Processor Initialization

In order for the installation to run with an attached processor, it must do the following:

- Specify AP=YES in the SYSCOR macro during VM/370 system generation
- Have the attached processor online during system IPL

If the attached processor was generated as part of the system but was not online, the following message is issued if the multiprocessing feature is installed:

```
DMKCPI959W ATTACHED PROCESSOR NOT ONLINE
```

If the attached processor is not required, processing can continue. However, if the attached processor is required, the operator should ready the unit and reload the system.

Assuming AP=YES and the attached processor was online the following message is issued.

```
DMKCPI958I ATTACHED PROCESSOR NOW OPERATIONAL
```

No operator action is required; the system will now utilize the attached processor.

If the installation is running with an attached processor, the system operator will be issued the following message when the clocks are to be synchronized.

```
CLOCK SYNCHRONIZATION-PRESS 'TOD ENABLE SET'
```

The operator will then press the TOD ENABLE SET key on either processor to synchronize the TOD clocks. Note that once the clocks are synchronized, it will not be necessary to resynchronize the clocks on subsequent IPL operations unless:

- FE diagnostics are run
- Power is turned off on either processor
- The clocks are running but not synchronized
- A clock's error is encountered during initialization

If the clocks should run out of low-order synchronization, after the clocks are synchronized (during or after system initialization), you will receive the following message:

```
DMKCLK970W TOD CLOCK SYNC CHECK RECEIVED
CLOCK SYNCHRONIZATION-PRESS 'TOD ENABLE SET'
```

As before, press the TOD Enable SET Key to synchronize the clocks.

Interval Timer

VM/370 uses the interval timer to support time slicing; therefore, during initialization, the system verifies that the interval timer is running before virtual machine logon operations are performed. The system initialization routine will loop issuing the following message if the interval timer is found not running when tested:

```
TURN ON THE INTERVAL TIMER
TURN ON THE INTERVAL TIMER
  .    .    .        .
  .    .    .        .
  .    .    .        .
```

The messages cease when the operator complies by enabling the interval timer and the system initialization process continues.

In attached processor application the text of the message indicates the processor owning the disabled interval timer. This message reads as follows:

```
TURN ON THE ( MAIN    ) PROCESSOR INTERVAL TIMER
            ( ATTACHED)
```

To continue, the interval timer must be enabled on the requested processor.

CONSOLE DEFINITION

The terms "system console", "primary system console", and "alternate system console" apply to those console devices that exercise primary control of the processor.

The VM/370 primary system console can be a real system console, a real alternate system console, or a local 3270.

The VM/370 alternate system console can be a real system console, a real alternate system console, a local 3270, or any VM/370 supported remote terminal on a leased line connected to a 270x transmission control unit (TCU). The VM/370 alternate system console cannot be a remote terminal connected to a 370x TCU.

Note: If the 370x is loaded with the 270x Emulation Program prior to initializing VM/370, the 370x then appears to VM/370 as a 270x TCU.

The following examples show the various configurations that can control VM/370 operations.

First Configuration:

Real system console = VM/370 system console
Real alternate system console = VM/370 alternate system console

Second Configuration:

Real system console = Batch virtual machine
Real alternate system console = VM/370 system console
Remote terminal = VM/370 alternate system console

Third Configuration:

Real system console = VM/370 system console
Remote terminal = VM/370 alternate system console

Fourth Configuration:

Real system console = Batch virtual machine
Two locally attached 3277 Display Stations = VM/370 system console and alternate system console

Fifth Configuration:

Enabled remote terminal (not 3270) = VM/370 alternate system console (No valid VM/370 primary system console is designated in SYSGEN)

Note: The real system consoles must be turned off to prevent VM/370 from using them.

Sixth Configuration:

System console = VM/370 system console
(No VM/370 alternate console designated in SYSGEN)

The first four configurations allow disabling of the VM/370 primary console and selection of an alternate VM/370 console; or, conversely, reselection of the VM/370 primary console and disabling of the alternate console.

Configurations five and six indicate that VM/370 system generation specified no console substitution. Therefore, if the designated VM/370 system console is inoperative at VM/370 startup, the system enters the wait state and cannot continue. After the console becomes operative again, VM/370 must be reloaded into the real system.

DISABLING THE VM/370 PRIMARY CONSOLE

If the VM/370 primary system console is inoperative, VM/370 automatically selects the first alternate console (specified in the RIOGEN macro instruction) as the console for primary system operation. If the first alternate console specified is not operational, an attempt will be made to start the next alternate console. If an operational console is found, it will be used as the VM/370 system operator's console.

If intermittent errors occur at the VM/370 console, and all alternate consoles are unavailable (or not specified in the RIOGEN macro instruction), issue the DISCONN command and log on again at some other terminal.

If the VM/370 system console becomes inoperative to the point where you cannot log off, press the System/370 external INTERRUPT button to disconnect you from the failing system console (or alternate console if that console is currently controlling the system.) This allows you to log on and reconnect to a terminal with a communications line that has already been enabled. This terminal can be another system console (VM/370 supported device) or a remote terminal attached to a 270x TCU. When the primary console is again operative, control can be regained from the alternate console by pressing the INTERRUPT button and reconnecting using the primary console.

When the device specified as the alternate console is a communication line, VM/370 enables the line. You can then establish a link with the processor through a terminal connected to the line. Once the line is established, VM/370 proceeds with the normal system initiation. The alternate console must not be a telecommunications line on a real IBM 370x Communications Controller. If the alternate console is an IBM 2741 Communication Terminal, it must use the EBCDIC transmission code.

<u>Note</u>: VM/370 370x supports multiple alternate console selection only in 270x emulator mode.

SELECTING A VM/370 ALTERNATE CONSOLE DURING VM/370 INITIALIZATION

If the VM/370 system console is inoperative, a VM/370 alternate system console can start VM/370. To do this, place the primary VM/370 system console in a not-ready status.

Load VM/370 in the normal way (with the System/370 console switches and the LOAD key). When the primary system console (assuming it was generated as the VM/370 system console) indicates "not ready," VM/370 rings the console alarm bell and enables the alternate system console (such as a 3210 Model 2, 1052 Model 7, or local 3270 if one is specified) or one 2701, 2702, or 2703 transmission control unit (TCU) line that has a leased line terminal attached, to allow the system operator to log on. The TCU line address must be defined in the REALIO module during VM/370 system generation. If the real system consoles are dedicated to virtual machine use only, the system can be generated with a fictitious VM/370 primary system console address and the same results occur as the load operation with the not ready console described above.

You can now establish a link with VM/370 by using the designated device. Once CP identifies the terminal in the normal procedure, the operator presses the BREAK, ATTN, RESET, or ENTER key, depending on the terminal type. The VM/370 system then performs automatic or manual logon procedures, as previously discussed.

Controlling the 370x Communications Controller

Because the 3704 and 3705 Communications Controllers are programmable devices, in order for them to be controlled by VM/370, they must be loaded with an appropriate control program before VM/370 uses them for system telecommunications. This loading is normally an automatic function of VM/370 system initialization, but it can be performed manually.

To load a 370x program into the 370x (either manually or automatically), the 370x program must previously have been saved on one of the VM/370 system volumes. Specifications in the VM/370 configuration module, DMKRIO, describe whether the load process is automatic or manual. For more information on saving or loading a 370x control program, see the *VM/370 Planning and System Generation Guide*.

If manual loading is necessary or desirable for a 370x control program, you should use the NETWORK LOAD command as follows:

```
NETWORK LOAD raddr ncpname
```

where raddr specifies the physical device address of the 370x, and ncpname is the name of a 370x control program image.

If the specified 370x Communications Controller does not require loading (that is, it already has an active control program loaded), the following message appears on your console:

```
DMKNLD461R CTLR raddr IPL NOT REQUIRED; ENTER 'YES' TO CONTINUE:
```

You should verify that you specified the correct 370x, and reply "YES" only if the load is to be attempted. (Note that execution of the NETWORK LOAD command causes the 370x to be reset, thus discontinuing any current teleprocessing activity.)

Other messages relating to the operation of the 370x can appear on your console. These messages are described in the *VM/370 System Messages*. For more information about the facilities of the NETWORK command, refer to the "NETWORK Dump Operations", "CP Commands", and "NCPDUMP" sections in this book.

CONSIDERATIONS FOR LOADING ALL 370X CONTROL PROGRAMS

When using the NETWORK LOAD command to load an EP for the 370x, the following events occur:

- Active users are disconnected
- Active I/O operations are reset
- Dedicated devices are detached and released
- Dialed lines are released
- Devices enabled but not dedicated are reset
- Binary synchronous communication are reset

SPECIAL CONSIDERATIONS FOR LOADING THE EP 370X CONTROL PROGRAM

If a 370x Emulation Program (EP) is automatically reloaded after a 370x failure, the system may loop after the restart. The message

```
DMKRNH463I CTLR xxx UNIT CHECK; RESTART IN PROGRESS
```

and two responses

```
CTLR xxx DUMP COMPLETE
CTLR xxx ncpname LOAD COMPLETE
```

indicate that the 370x has been reloaded. If the system loops after the second response, you must reset all emulator lines from the 370x control panel.

If the automatic dump feature is not enabled, one of the messages

```
DMKRNH462I CTLR raddr UNIT CHECK; IPL REQUIRED
DMKRNH464I CTLR 'raddr' CC=3; DEPRESS 370X "LOAD" BUTTON
```

indicates a 370x abnormal termination. You must reload the 370x Emulation Program with the NETWORK LOAD command. If the system loops when an attempt is made to enable the lines, you must reset all emulator lines from the 370x control panel.

The <u>IBM 3704 Control Panel Guide</u> and the <u>IBM 3705 Control Panel Guide</u> describe the procedure for resetting emulator lines from the 370x control panel in its "Generating Channel End/Device End with Emulator Program" section.

A 370x communications controller cannot be used by VM/370, as a 370x, unless it has been loaded by VM/370. Use of the ATTACH and DETACH commands reset the "loaded" status and, therefore, the 370x will need to be reloaded.

Messages on Startup and Initialization

In the course of VM/370 initialization or startup, in addition to the normal expected messages that are a part of loading the system, you may see other messages to which you must respond. These messages relate to functions of initialization, checkpoint, warm start, and automatic checkpoint requirements. These messages and the recommended course of action are described in the VM/370 System Messages.

Console Messages After Startup

System console messages are in three categories:

- System operation status messages
- Virtual machine user messages
- Hardware and program status messages

The operator action required depends on the type and content of the message.

SYSTEM OPERATION STATUS MESSAGES

These messages inform the system operator of the logging on and logging off of users, number of spooled files, output printer or punch accounting data, device attention conditions, and command completion indications. Some of these messages require operator action, while others do not.

MESSAGES FROM VIRTUAL MACHINE USERS

These messages include requests and inquiries from virtual machine users on the system; a sampling of these user messages follows:

```
MSG FROM JONES: CAN YOU GIVE ME A TAPE
MSG FROM SMITH: CAN YOU PUT UP A SCRATCH 181
MSG FROM SMITH: PLEASE TAKE RING OUT OF 181 NOW
MSG FROM BROWN: PLS ATTACH 382
```

HARDWARE AND PROGRAM STATUS MESSAGES

These messages include VM/370 error recording messages, VM/370 system error information, and I/O error messages related to users of virtual machines who have invoked SVC 76. SVC 76 causes VM/370 to send a message to the system operator (for I/O error records only) and to record the error data in VM/370's error recording area. The VM/370 System Messages manual describes these messages.

For more information about error detection and the recording and editing of errors, refer to the VM/370 OLTSEP and Error Recording Guide.

The following summarizes the action that the operator must take in response to an error message:

1. Determine the severity of the error.

2. Take a storage dump if the automatic dump routine failed.

3. Try to preserve spooling and accounting data by issuing the SHUTDOWN command if it has not been done automatically.

4. Try to force error recording if it did not occur.

5. Attempt to analyze the problem by whatever means at his disposal and if necessary, reallocate resources and priorities.

6. Call for technical assistance if necessary.

Whenever there are indications that I/O errors require further analysis or that basic integrity of a device needs to be determined, other service programs may be invoked to help make such an assessment. For example, to test the operational 3344 or 3350 DASD devices, the operator may invoke the OS/VS and DOS/VS Analysis Program-1 (AP-1). This utility program, if resident in the OS/VS or DOS/VS library of a virtual machine, can be used to check the proper operation of the device. The use of this program along with error messages and user action is detailed in the OS/VS and DOS/VS Analysis Program-1 (AP-1) User's Guide, GC26-3855.

In the event that technical assistance is required, it can be obtained from the following sources:

- The customer's own system programmer(s).

- The service representative, who has special test equipment for hardware problem isolation.

- The IBM Program Support Representative, who can also assist with VM/370 system problem diagnosis.

You can assist with problem diagnosis by being alert for the following error recording messages:

```
DMKIOG550E I/O ERROR RECORDING AREA 90 PERCENT FULL; RUN CPEREP

DMKIOG551E I/O ERROR RECORDING AREA FULL; RUN CPEREP
```

You should then dump the error recording data to some other device (usually a tape drive) and clear the error recording area to insure enough space to record additional error records. Information on how to run CPEREP is contained in the VM/370 OLTSEP and Error Recording Guide. Additional information on OS/VS EREP, the program to which CPEREP passes control is detailed in the OS/VS Environmental Recording Editing and Printing (EREP) Program publication.

Other errors may not be severe enough to cause VM/370 system failure, but might cause a specific virtual machine to abnormally terminate. In most cases, the user can IPL the virtual machine again.

Should the VM/370 machine enter an unexpected WAIT state or loop, see the VM/370 System Programmer's Guide referring to the section entitled "How To Use the VM/370 Facilities To Debug."

3800 Printing Subsystem Support

The 3800 printer is supported with all its functions as a dedicated device. It has limited functional support when it is used as a spooling device. Following is a list of features of the 3800 printer:

- FORMS CONTROL permits the user to select the amount of vertical space between printed lines. Lines can be vertically spaced at 6, 8, or 12 lines per inch. Different spacings can also be mixed on the same page.

- FORMS OVERLAY permits photographing of predefined data (forms) on one or more pages. The 3800 printer supports the creation of a form on a page and the immediate printing of data on that form.

- PREDEFINED CHARACTER SETS permit the user to select previously designated character sets that contain up to 64 characters each. Different character sets provide varied character types, sizes, and horizontal spacings between the characters. The 3800 supports the simultaneous use of two character sets and optionally supports the use of two additional character sets. The user can mix characters from the different character sets on a single page. Characters are referenced within the 3800 through the use of translate tables. Up to four translate tables may be loaded at any one time.

- CHARACTER SET MODIFICATION permits user modification and extension of character sets.

- COPY MODIFICATION permits printing of predefined data (for example, printing the same header on each page) or the suppression of printing for selected data.

- MULTIPLE COPIES permits numerous copies of files and datasets to be printed without the use of multiple-ply paper. Numerous copies of a single 3800 buffer can also be printed.

The values that control feature selection and printing on the 3800 are stored in tables and modules. Through use of the CP commands VM/370 can display the symbolic names of the translate tables and modules in addition to the symbolic name of the library where they are stored.

Before printing a spool file, VM/370 initializes the 3800 by loading it with the translate tables that control feature selection and printing. VM/370 provides a capability to purge spool files that cause load errors, which occur when a name is specified that does not match the specified image library. If PURGE is specified, all spool files causing 3800 load errors will be purged. Otherwise, the spool file will be held.

DEDICATED 3800 PRINTER SUPPORT

VM/370 allows virtual machines, including virtual VM/370, to attach the 3800 printer as a dedicated I/O device. When attached, the full capability of the 3800 is available to the virtual machine. All of the functions previously described are available when the 3800 is used as a dedicated device, provided the device is specified in the RDEVICE macro instruction at system generation. See the _VM/370 Planning and System Generation Guide_.

Performance Options

VM/370 offers the following performance options to enhance the operating efficiency of one or more virtual machines. This enhancement is normally at the expense of the performance of other virtual machines. In certain instances, enhancements to a specific virtual machine (or multiple virtual machines) are detrimental to the operating efficiency of other virtual machines. The performance options are:

- Virtual machine assist
- VM/370 Extended Control-Program Support
- Reserved page frames
- Favored execution
- Virtual=real
- Locked pages
- Priority
- Affinity in an attached processor environment

You can dynamically assign or withdraw any of these options except for the virtual=real option (it can only be dynamically withdrawn) from virtual machine users during system operations. You can assign to one virtual machine a "mix" of performance options, or the options may be assigned to separate virtual machines. Efficient use of these options depends upon the characteristics of the program applications; the number, size, and complexity of the virtual machines being run; and the size, I/O configuration, and CPU model of the real computer being used.

In addition to using these performance options, you can also use the SET PRIORITY command to change a virtual machine's execution priority and, thereby, change its performance.

For example, you could assign the following mix of options.

```
  One virtual machine:
            Reserved Pages=10
            Favored Execution=80%
            Virtual Machine Assist

  Another virtual machine:
            Virtual=Real
            Priority=1

  A third virtual machine:
            Locked Pages=2
```

The favored execution (with percentage value specified), VIRT=REAL, and the reserved page option are unique in that each option cannot be applied to multiple virtual machines at the same time. However, no such restrictions apply to the locked pages, virtual machine assist, VM/370 Extended Control-Program Support, priority, or favored execution (without the percentage value) options; they can be applied to multiple virtual machines.

RESERVED PAGE FRAMES OPTION

The CP paging routine uses chained lists of available and pageable pages to control real storage use. CP assigns pages for users from the available list, which it replenishes from the pageable list.

Pages that are temporarily locked in real storage are not available or pageable. Paging proceeds using demand paging to select the best page for swapping. The reserved page option gives a particular virtual machine an essentially private set of pages. The pages are not locked; they can be swapped, but normally only for the virtual machine for which they are reserved. You specify the number of reserved pages for the virtual machine as a maximum. When the page selection routine selects an available page for a reserved user, it marks that page reserved if the maximum specified for the user has not previously been reached. If CP encounters an available (idle) reserved page for the reserved user during page selection, it is used whether or not the maximum has been reached.

Only an operator with privilege class A can specify the number of reserved pages. Issue the CP SET RESERVE command in the following format:

```
SET RESERVE userid xx
```

where:

userid is the name of the VM/370 user to whose virtual machine you are to assign the reserved page option.

xx is the maximum number of pages you assign.

If the CP page selection routine cannot locate an available page for other users because they are all reserved, the routine uses one of the reserved pages as a last resort.

FAVORED EXECUTION OPTION

The favored execution option guarantees that a particular virtual machine gets up to a fixed percentage of processor time. Favored in this sense means that CP provides execution time up to the specified percentage to that virtual machine, _provided_ that virtual machine can use that much processor time. At regular time intervals, the CP dispatcher checks the processor time used by the favored virtual machine. If the guaranteed percentage is exceeded, the machine gets its normal priority for the remainder of the time interval. If the percentage used is less than the guaranteed value, the favored virtual machine has the highest execution priority until it reaches that threshold. You, as the system operator with privilege class A, specify the percentage of processor time assured by the favored option in the CP SET command. The format of the command is:

```
                   ┌   ┐
SET FAVORED userid │xx │
                   │OFF│
                   └   ┘
```

where:

userid is the name of the VM/370 user whose virtual machine is to be assigned the favored option

xx is any percentage value from 1 to 99

OFF operand stops favored operation.

You can assign multiple virtual machines the favored execution option only one virtual machine at a time can be assigned a specified favored percentage value.

Example:

```
set favored operatns 75
set favored payroll
set favored revenue
set favored hotjobs
```

VIRTUAL=REAL OPTION

The virtual=real option permanently relocates the CP nucleus (except for real page 0) to provide space in real lower storage to contain the largest virtual=real machine. For the virtual machine, each page from page 1 to its last or nth page is in its true real storage location; only its page zero is relocated. The virtual machine still runs in relocate mode, but because each virtual page frame address is the same as the real page frame address, the virtual machine requires no CCW translation for the virtual machine. Because no CCW translation is performed, no check is made of the I/O data addresses. The virtual machine must ensure that no I/O data transfer occurs into real page zero or beyond the upper boundary of the virtual machine's storage. Failure to observe these limits can cause damage to the VM/370 system or to other virtual machines.

Initial program loading of a named or shared system is not allowed in the virtual=real machine; the hexadecimal device address must be used.

Several aspects of the operation of the virtual=real option can affect overall system operation:

1. The area of contiguous storage built for the virtual=real machine must be large enough to contain the entire addressing space of the largest virtual=real machine.

2. Only virtual machines with the virtual=real option can use the real storage reserved for virtual=real machines. Only one virtual=real virtual machine can run at a time. The storage is not available for other users, nor for VM/370 use, unless the virtual=real machine is not logged on and you issue an UNLOCK command with the VIRT=REAL option. When you issue this command, all page frames previously reserved for the virtual=real machine are permanently relinquished to the available-page pool. Once these virtual=real page frames are available to the page pool, no virtual=real machine can run until you load VM/370 again (via IPL). For this reason, it is suggested that the virtual=real machine should be a machine with high availability and heavy workload demands. It is not possible to relinquish only some of the virtual=real page frames (if, for example, a smaller virtual=real machine is to be run). All virtual=real page frames are locked until all are released to the pool.

3. The virtual=real machine operates in its allocated storage area with normal CCW translation in effect until the virtual machine operator issues the CP command:

   SET NOTRANS ON

   At this time, all subsequent I/O operations occur from the virtual CCWs in the virtual=real machine without translation. With NOTRANS on, the virtual machine must not perform I/O operations into page zero or beyond its addressable limit. As mentioned previously, violation of this requirement can cause damage to the VM/370 system or to other virtual machines.

4. If the virtual=real machine performs a reset or IPL, the normal CCW translation routine regains control until the virtual machine operator again issues the SET NOTRANS ON command. Only the virtual=real machine operator can use the command. An operator message occurs if the virtual machine enters normal translation mode.

The virtual=real option is used with programs or operating systems that dynamically modify channel programs, or that wish to avoid CP virtual channel program translation and paging degradation. If virtual=real operation is desired, it must be specified at VM/370 system generation time. Virtual=real assigns the same real storage locations to the virtual machine that exist on the real machine (except for page zero, which is relocated).

For example, in Figure 2, if a DOS system needs 160K bytes to run virtual=real, CP puts all but the first 4096 bytes of the program at the real storage address locations. Page 0 of the DOS supervisor is the only part of the DOS virtual machine that can be relocated, and it follows the last real page frame of the DOS machine. Figure 2 also shows that the major portion of the CP nucleus is permanently relocated into upper storage (except for page zero) whether the virtual=real system is operating or not. The real page frame zero remains under control of VM/370. CP dedicates 160K nonpageable bytes to the DOS machine.

However, when the user no longer needs the virtual=real machine performance option and he logs off, the assigned virtual=real storage can be reallocated as available pageable storage for use by other virtual machines (assuming that no other users with the VIRT=REAL directory have need of the area). The reallocation of virtual=real storage as usable and available storage for other virtual machines is controlled by the UNLOCK command and the VIRT=REAL operand. Once the pages are released, no virtual=real machine can be run unless VM/370 itself is reloaded. If more than one virtual machine needs the capabilities of running virtual=real (at different times), VM/370 system generation parameters must specify the largest virtual machine that is to run in the virtual=real environment. A more practical approach may be to generate a different version of VM/370 for each virtual=real machine, if the storage sizes are substantially different.

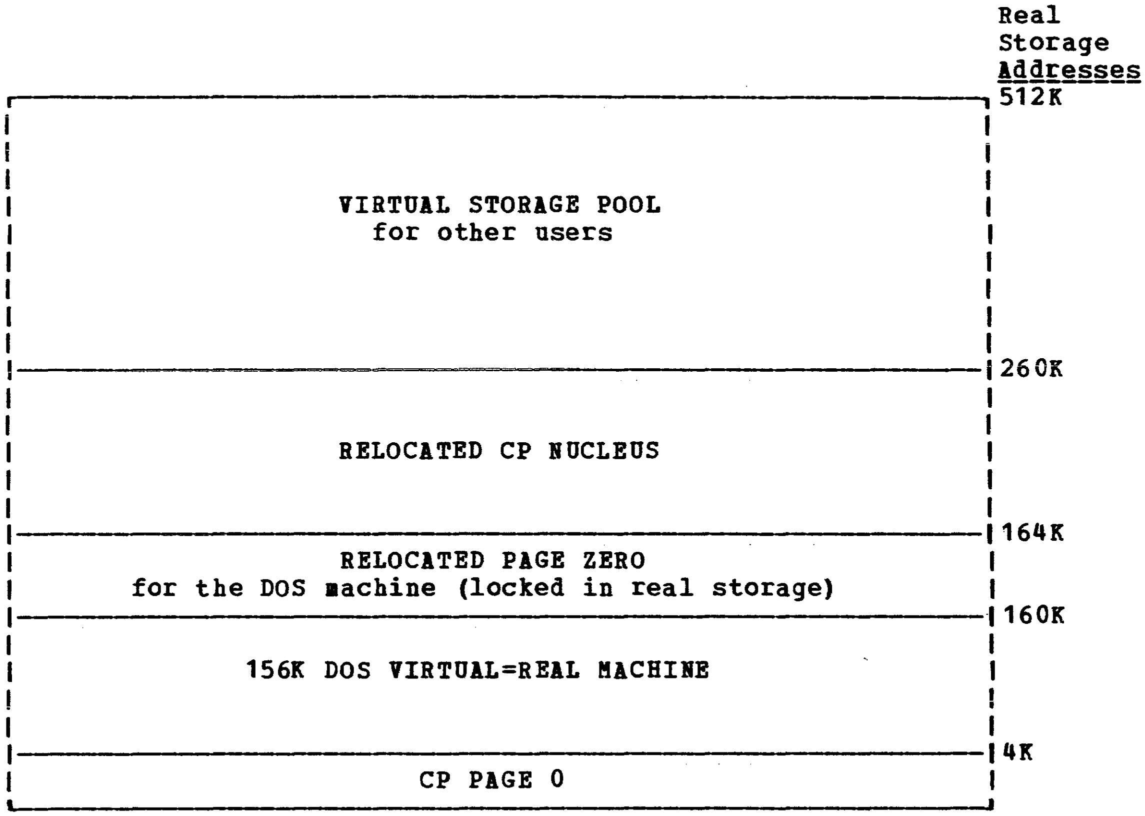

Real Storage Addresses
512K
VIRTUAL STORAGE POOL for other users
260K
RELOCATED CP NUCLEUS
164K
RELOCATED PAGE ZERO for the DOS machine (locked in real storage)
160K
156K DOS VIRTUAL=REAL MACHINE
4K
CP PAGE 0

Figure 2. DOS 160K Virtual=Real Storage Assignments

LOCKED PAGES OPTION

If sufficient page frames are available for other users, you can choose to lock (fix) one or more specific pages of a virtual machine or the pageable CP nucleus in real storage. This performance option differs from the other options in that the LOCK command can apply to several virtual machines. It is also possible to have the reserved page frames option apply to one user's virtual machine, and then have the lock option apply to another user's virtual machine. A locked page is effective from the time you invoke the lock option until the user logs off the system, or until you issue the UNLOCK command. If the page or pages that are locked have a high frequency of reference or change activity, the lock option, like the reserved option, can cause an improvement in the user's virtual machine performance or overall system performance because of the reduction in paging activity.

Notes:

1. If a named system currently has the locked pages option in effect, reloading of the same named system, loading of another named system, or a system reset does not affect the locked pages option of a virtual machine. The loading of named systems over other named systems without using the UNLOCK command beforehand results in pages being made unavailable for use if the first named system has locked pages in effect.

2. In an attached processor system shared pages cannot be locked.

3. You should always prefer the reserved page frames option to the locked pages option, if it is not already in use. The LOCK command locks specific pages into real storage regardless of activity, while the reserved page frames option keeps all the currently most active pages in real storage.

PRIORITY

You can assign specific (dispatch) priority values to different virtual machines to override the assigned or default priority value for each user in the VM/370 directory. In doing so, the virtual machine with a lower nn value is considered for dispatching before a virtual machine with a higher nn value. You set user priorities by using the following class A console function:

```
SET PRIORITY userid nn
```

where:

userid is the user's identification

nn is an integer value from 1 to 99. The higher the nn value, the lower the dispatching priority.

VM/370 ASSISTANCE - HARDWARE ASSIST

The overhead associated with CP's handling of virtual machines and their collective I/O resources is extensive and time consuming. To reduce this overhead, CP is assisted by System/370 processor hardware that duplicates some of the functions provided by CP routines to accelerate CPU processing. This hardware assistance to program function is called | VM/370 hardware assist. Some form of VM/370 hardware assist is | available on all VM/370 supported System/370 systems except the models | 155, 155II, 165, 165-3, and 4331. VM/370 hardware assist on certain processors can consist of virtual machine assist; on other processors VM/370 hardware assistance is composed of virtual machine assist and VM/370 Extended Control Program Support (VM/370 ECPS).

VM/370 hardware assist is activated when VM/370 is loaded or by the use of CP commands, and it is deactivated by CP commands. For the command that activates/deactivates the VM/370 hardware assist process, refer to the SET command described in "Section 3. CP Commands." The status of VM/370 hardware assist may be determined by the use of the QUERY command also described in Section 3.

Virtual Machine Assist

The virtual machine assist facility (VMA) can be ordered via feature number, it is available as an RPQ (Request for Price Quotation), and is standard on some processors. For a list of VM/370 supported processors and how they support VMA and ECPS see *VM/370 Planning and System Generation Guide*. On attached processor systems, virtual machine assist maybe installed on one or both processors. Virtual machine assist relieves CP of some of the processing overhead caused by the simulated execution of certain instructions, privileged operation instructions, and interrupts of virtual machines. On/off system control of virtual machine assist is provided by the A privilege class command:

```
SET SASSIST {ON } [[PROC] nn]
            {OFF}
```

Users of virtual machines, in turn, have discrete control over the functions of virtual machine assist by means of directory entry options or by command. For control of this function on a virtual machine level, refer to the privilege class G command SET ASSIST, described in *VM/370 CP Command Reference for General Users*.

VM/370 Extended Control-Program Support

VM/370 Extended Control-Program Support is an expansion of the function provided by virtual machine assist and consists of three major parts:

- Expanded virtual machine assist
- CP assist
- Virtual interval timer assist

Expanded virtual machine assist is an expansion of function provided by virtual machine assist in that, via hardware, CP is relieved of processing some conditions of CP simulated instruction execution that were not covered by the virtual machine assist process. Expanded virtual machine assist also assists CP in the processing of other instructions and privileged operations not emulated by virtual machine assist.

Expanded virtual machine assist is activated for all virtual machines when VM/370 is loaded. It is also activated when both class A commands are invoked as shown:

```
SET SASSIST ON

    --and--

SET CPASSIST ON
```

Expanded virtual machine assist is deactivated when either SET SASSIST OFF or SET CPASSIST OFF is invoked.

The CP assist portion of the VM/370 Extended Control-Program Support duplicates via hardware the function provided by various high-usage significant overhead portions of CP routines. CP assist is activated upon system initialization or by invoking the privilege class A command:

```
SET CPASSIST ON
```

When the OFF operand of CPASSIST is invoked, CP assist is deactivated (as well as expanded virtual machine assist portion of ECPS, if previously active).

The third function provided by VM/370 ECPS is virtual interval timer assist. This hardware assist provides virtual machines with a more accurate method of updating the virtual machine's interval timer (location X'50'). Overall system control of this function is by the class A commands SET SASSIST ON and SET SASSIST OFF. Control of this facility on the individual virtual machine level is by the class G command SET ASSIST TMR and SET ASSIST NOTMR. For more details on this, refer to the _VM/370 CP Command Reference for General Users_.

To summarize the use of the SET CPASSIST and SET SASSIST in relationship to virtual machine assist and VM/370 Extended Control-Program Support, see Figure 3.

VM/370 Hardware Assist	SET SASSIST ON SET CPASSIST OFF	SET SASSIST ON SET CPASSIST ON	SET SASSIST OFF SET CPASSIST ON	SET SASSIST OFF SET CPASSIST OFF
Virtual Machine Assist	X	X		
Virtual Interval Timer Assist	X	X		
Extended Virtual Machine Assist		X		
CP Assist		X	X	
Legend: X = function activated				

Figure 3. Command Control of VM/370 Hardware Assist

VM/370 Hardware Assist Restrictions

A virtual machine running in certain virtual machine modes or running certain operations cannot use certain facilities provided by VM/370 hardware assist. When these modes or operations are encountered, the action taken is dependent upon the processor facilities of either virtual machine assist or VM/370 Extended Control-Program Support.

For all VM/370 supported processors that have VM/370 hardware assist active when the CP ADSTOP or certain CP TRACE command functions are encountered, the virtual machine assist is deactivated during the interval of their use.

For System/370 Models 135-3, 138, 145-3, and 148, the use of the DOS emulator or PER (Program Event Recording) curtails some of the function provided by VM/370 Extended Control-Program Support. For more details on VM/370 hardware assist, see the *VM/370 System Programmer's Guide*.

AFFINITY

In Attached Processor systems a virtual machine may be designated to execute only on one specified processor. This may be advantageous for performance or functional requirements. Performance gains might be realized with affinity because of the nature of the workload or the availability of performance assists on one processor. Functional requirements could be the existence of required operation codes on one processor. In either case, CP code executed on behalf of the virtual machine may execute on either processor.

System Termination

The VM/370 system may be shut down in one of two ways:

1. By an operator initiated shutdown.
2. By an abnormal termination of operation.

The normal shutdown is initiated by a class A operator entering the command

"shutdown"

from the operator's console. If the SHUTDOWN command is ineffective, press the LOAD button on the system console once to shut down the system (a second pressing of the load button reloads VM/370).

The shutdown operation checkpoints spooling and accounting functions; in addition, user activity is terminated by automatically logging off all virtual machines and disabling communication lines. Open and active files on the spool unit record devices are purged and, if applicable the SUSPEND order is issued to all 3851 device addresses (suspends MCS interface activity to the host virtual machine). Note that the SUSPEND order issued to a Mass Storage Control interface may cause the SHUTDOWN command to run longer than usual.

Note: Before issuing the SHUTDOWN command, a message or warning should be sent to all logged-on users so that they can quickly bring their virtual machines to an orderly halt. Time should also be allotted to the class D (spooling) operator so that he too can terminate in an orderly manner the active files being processed on the unit record devices.

The dump unit for VM/370 system failures is specified in the SYSDUMP macro during VM/370 system generation, but can be changed by a privilege class A or class B system operator.

If the dump unit is set to disk (by default or via the SET DUMP AUTO command) at the time of system failure, the system dumps all or parts of real storage to the specified disk and automatically restarts the VM/370 system.

When automatically restarted, the system preserves all accounting information and spool file data on disk, performs an automatic logon of the primary system operator, restores the system LOGMSG, and continues system operation. It is not necessary to re-enable the lines, since dump and re-IPL automatically re-enable them. The VM/370 online message is sent to the terminal users, indicating that they must log on again.

If, at the time of the system failure, the dump unit is set to a printer or tape, the VM/370 system writes the dump on the specified unit, preserves the spooling and accounting data, and stops. The operator must then re-IPL the VM/370 system as for normal system start up, specifying a WARM START to preserve the accounting and spooling file data. The operator must re-enable the communication lines to permit users to log on again.

Note: An abnormal termination of an attached processor configuration is caused by the same conditions that cause abnormal termination of a uniprocessor configuration (uncorrectable hardware error, program check during the execution of CP, CP issues abnormal termination request because of logical processing errors, or the primary system operator initiates a SYSTEM RESTART).

SYSTEM ABEND DUMPS

Conditions can occur within the CP program that may force an abnormal ending condition (ABEND) and cause the dumping of system registers and storage. The device that receives these records can be a tape, printer, or disk device.

A system restart dump is taken when the system operator presses the SYSTEM RESTART button on the processor console or initiates system restart in some other manner as dictated by the processor device type. In an AP configuration, the attached processor should not be stopped before the system restart is initiated.

Dumping operations are caused by any program interruptions or system restart condition. These interruptions cause routines to gather data from registers and storage and place this data on a previously defined device. The system CP command SET DUMP defines the quantity of data to be dumped.

For example:

```
SET DUMP AUTO
SET DUMP raddr
SET DUMP raddr ALL
SET DUMP AUTO ALL
```

SET DUMP AUTO places the VM/370 system dump on a preselected file device. (The disk dump area is automatically selected at system initialization time if sufficient contiguous space is available.) You can verify the device type and address by entering the QUERY DUMP command.

Use SET DUMP raddr when the dump device is to be a high speed printer or tape device. The real hexadecimal one to three digit device address is substituted for the raddr operand.

The ALL operand used with SET DUMP AUTO or SET DUMP raddr dumps all of storage onto to the DUMP device. If the ALL operand is not specified, the system defaults to dumping only those areas that pertain to CP, and not those areas that pertain to virtual machine operations.

If you dump to a disk file, an additional operation is necessary to transform these records into readable output for programmers or system analysts. You can do this with the CMS VMFDUMP program.

Note that expanded function is available to the VMFDUMP command as generated for the IPCS virtual machine. For more information on this, refer to the _VM/370 Interactive Problem Control System (IPCS) User's Guide_.

If the records are dumped onto a tape drive, other CMS command options must be invoked for printout.

Only abend dumps that are a result of using the SET DUMP AUTO command are spooled as a special virtual card reader file. This card reader file is assigned during system generation to a specific virtual machine user via the SYSOPR macro. The CMS VMFDUMP command formats and prints these CP abend dumps.

The VMFDUMP command creates a CMS file from the CP disk dump data and prints the dump from the CMS file.

The VM/Interactive Problem Control System Extension (VM/IPCS Extension) provides installations with expanded facilities for reporting and diagnosing software failure. The VMDUMP command allows the user to dump a virtual machine's entire storage area. The VM/IPCS Extension program product or a user written program must process the VMDUMP command. If you have installed this program, see the _VM/Interactive Problem Control System Extension User's Guide and Reference_, Order No. SC34-2020.

Section 3. CP Commands

The CP commands represent a set of interactive console functions that the operator uses to control the VM/370 system, and that permit user control of virtual machines. CP indicates the results of command processing with a conversational response, if appropriate (for example, COMMAND COMPLETE, MISSING ARGUMENT, or INVALID OPTION). The commands that a VM/370 user (including the operator) can issue depend upon the user's assigned privilege class(es), as described in the following paragraphs.

PRIVILEGE CLASSES FOR CP COMMANDS

The CP commands are divided into eight groups, each represented by a privilege class that indicates the type of user from which the commands are accepted. Each user is assigned, as part of his entry in the VM/370 directory, one or more privilege classes (Figure 4 shows the function of each class). The exceptions are users with a password of NOLOG. The NOLOG users have no privilege class and can only send messages via the MESSAGE command or receive spooled output as punched cards or printed forms. The NOLOG password allows them to be identified as the recipients of spooled output when a virtual machine user spools output for them.

Class	User and Function
A[1]	Primary System Operator: The class A user controls the VM/370 system. Class A is assigned to the user at the VM/370 system console during IPL. The primary system operator is responsible for the availability of the VM/370 system and its communication lines and resources. In addition, the class A user controls system accounting, broadcast messages, virtual machine performance options and other command operands that affect the overall performance of VM/370. Note: The class A system operator who is automatically logged on during CP initialization is designated as the primary system operator.
B[1]	System Resource Operator: The class B user controls all the real resources of the VM/370 system, except those controlled by the primary system operator and spooling operator.
C[1]	System Programmer: The class C user updates certain functions of the VM/370 system.
D[1]	Spooling Operator: The class D user controls spool data files and specific functions of the system's unit record equipment.
E[1]	System Analyst: The class E user examines and saves certain data in the VM/370 storage area.
F[1]	Service Representative: The class F user obtains, and examines, in detail, certain data about input and output devices connected to the VM/370 system.
G[2]	General User: The class G user controls functions associated with the execution of his virtual machine.
Any[2]	The Any classification is given to certain CP commands that are available to any user. These are primarily for the purpose of gaining and relinquishing access to the VM/370 system.
H	Reserved for IBM use.

[1] Described in this publication.

[2] Described in the *VM/370 CP Command Reference for General Users*.

Figure 4. CP Privilege Class Descriptions

In most cases, a user is assigned to the class G commands. Any attempt by a user to issue a command that is not within his command classes causes an error response from CP. Figure 5 shows the commands that are accepted from each user command class. Some commands fall into more than one class. In those cases, the major operands of those commands are shown in the classes in which they are valid.

Class	Commands	Operands
Any	*	
	#CP	
	CP	
	DIAL	
	DISCONN	
	LOGOFF	
	LOGON	
	MESSAGE	
	SLEEP	
A	ACNT	
	AUTOLOG	
	DISABLE	
	ENABLE	
	FORCE	
	HALT	
	LOCK	
	MESSAGE	ALL
	MONITOR	
	NETWORK	DISABLE
		DISPLAY
		DUMP
		ENABLE
		HALT
		LOAD
		POLLDLAY
		QUERY
		VARY
	QUERY	AFFINITY
		CPASSIST
		JOURNAL
		LOGMSG
		NAMES
		PAGING
		PRIORITY
		PROCESSOR
		SASSIST
		userid
		USERS
	SET	AFFINITY
		CPASSIST
		JOURNAL
		FAVORED
		PRIORITY
		RESERVE
		SASSIST
	SHUTDOWN	
	UNLOCK	
	WARNING	
B	ATTACH	
	ATTACH	CHANNEL
	AUTOLOG	
	DEFINE	SYSVIRT
		VIRTUAL
	DETACH	
	DETACH	CHANNEL
B (cont.)	DISABLE	
	ENABLE	
	MESSAGE	ALL
	MSGNOH	
	NETWORK	DISABLE
		DISPLAY
		DUMP
		ENABLE
		LOAD
		POLLDLAY
		QUERY
		VARY
	QUERY	ALL
		DASD
		DUMP
		GRAF
		LINES
		LOGMSG
		NAMES
		PROCESSOR
		raddr
		STORAGE
		SYSTEM
		TAPES
		TDSK
		UR
		userid
		USERS
	SET	DUMP
		LOGMSG
	VARY	
	WARNING	
C	DCP	
	DMCP	
	LOCATE	
	QUERY	LOGMSG
		NAMES
		PROCESSOR
		userid
		USERS
	STCP	
D	BACKSPAC	
	CHANGE	
	DRAIN	
	FLUSH	
	FREE	
	HOLD	
	LOADBUF	
	ORDER	
	PURGE	
	QUERY	FILES
		HOLD
		LOGMSG
		NAMES
		PRINTER
		PUNCH

Figure 5. Commands Accepted from Each User Class (Part 1 of 2)

Class	Commands	Operands
D (cont.)		READER
		userid
		USERS
	REPEAT	
	SPACE	
	START	
	TRANSFER	
E	DCP	
	DMCP	
	INDICATE	I/O
		LOAD
		PAGING
		USER
		QUEUES
	LOCATE	
	MONITOR	
	QUERY	AFFINITY
		CPASSIST
		JOURNAL
		LOGMSG
		NAMES
		PAGING
		PRIORITY
		PROCESSOR
		SASSIST
		userid
		USERS
	SAVESYS	
F	QUERY	LOGMSG
		NAMES
		userid
		USERS
	SET	RECORD
		MODE
G	ADSTOP	
	ATTN	
	BEGIN	
	CHANGE	
	CLOSE	
	COUPLE	
	DEFINE	
	DETACH	
	DETACH	CHANNEL
	DISPLAY	
	DUMP	
	ECHO	
	EXTERNAL	
	INDICATE	LOAD
		USER
	IPL	
	LINK	
	LOADVFCB	
	NOTREADY	
	ORDER	
	PURGE	
	QUERY	ALL
		CHANNELS
G (cont.)		CONSOLE
		DASD
		FILES
		CPUID
		GRAF
		LINES
		LINKS
		LOGMSG
		NAMES
		PFnn
		PRINTER
		PROCESSOR
		PUNCH
		READER
		SET
		STORAGE
		TAPES
		TERMINAL
		TIME
		UR
		userid
		USERS
		vaddr
		VIRTUAL
	READY	
	REQUEST	
	RESET	
	REWIND	
	SET	AFFINITY
		ACNT
		ASSIST
		AUTOPOLL
		CPUID
		ECMODE
		EMSG
		IMSG
		ISAM
		LINEDIT
		MSG
		NOTRANS
		PAGEX
		PFnn
		RUN
		SMSG
		TIMER
		WNG
	SMSG	
	SPOOL	
	STORE	
	SYSTEM	
	TAG	DEV
		FILE
		QUERY
	TERMINAL	
	TRACE	
	TRANSFER	
	VMDUMP	

Figure 5. Commands Accepted from Each User Class (Part 2 of 2)

OPERATOR COMMANDS

Section 3 describes CP commands available to users with the privilege classes of A, B, C, D, E, and F. These privilege classes are assigned to users who manage the VM/370 system and its spooling and hardware resources. The privilege class G commands control the functions of the general user's virtual machine. Consult the _VM/370 CP Command Reference for General Users_ for information on all G privilege class commands. The class of commands described as "Any", which can be used by any user of the system, are also described in the _VM/370 CP Command Reference for General Users_.

NOTATIONAL CONVENTIONS

The notation used to define the command syntax for VM/370 is:

- Truncations and Abbreviations of Commands

  Where truncation of a command name is permitted, the shortest acceptable version of the command is represented by uppercase letters. (Remember, however, that VM/370 commands can be entered with any combination of uppercase and lowercase letters.) The example below shows the format specification for the INDICATE command.

  ```
  INDicate
  ```

  This representation means that IND, INDI, INDIC, INDICA, INDICAT, and INDICATE are all valid specifications for this command name.

  Operands are specified in the same manner. Where truncation is permitted, the shortest acceptable version of the operand is represented by uppercase letters in the command format box. If no minimum truncation is noted, the entire word (represented by all capital letters) must be entered.

  Abbreviations are shorter forms of command names and operands. Abbreviations for command names are shown below the full name in the format box. Abbreviations for operands are shown in the description of the individual operands that follows the format box. For example, the operand READER has both a minimum truncation and an abbreviation. In the format box, it is shown as:

  ```
  Reader
  ```

  indicating that the minimum truncation is R. In the discussion of the READER operand that follows the format blcck, it is shown as:

  ```
  READER
  RDR
  ```

  indicating that the abbreviation is RDR. Thus, the acceptable specifications for the READER operand are: R, RE, REA, READ, READE, READER, and RDR.

- The following symbols are used to define the command format and should never be typed when the actual command is entered.

underscore	_
braces	{ }
brackets	[ ]
ellipsis	...

- Uppercase letters and words, and the following symbols, should be entered as specified in the format box.

asterisk	*
comma	,
hyphen	-
equal sign	=
parentheses	( )
period	.
colon	:

- Lowercase letters, words, and symbols that appear in the command format box represent variables for which specific information should be substituted. For example, "hexloc" indicates that a hexadecimal storage location address is to be entered.

- Choices are represented in the command format boxes by stacking.

  A
  B
  C

- An underscore indicates an assumed default. If an underscored choice is selected, it need not be specified when the command is entered.

  Example
  The representation

  A
  $\underline{\text{B}}$
  C

  indicates that either A, B, or C may be selected. However, if B is selected, it need not be specified. Or, if none is entered, B is assumed.

- The use of braces denotes choices, one of which must be selected.

  Example
  The representation

  $$\left\{\begin{matrix}\text{A}\\ \text{B}\\ \text{C}\end{matrix}\right\}$$

  indicates that you must specify either A, or B, or C. If a list of choices is enclosed by neither brackets or braces, it is to be treated as if enclosed by braces.

- The use of brackets denotes choices, one of which may be selected.

  Example
  The representation

  ```
  ┌   ┐
  | A |
  | B |
  | C |
  └   ┘
  ```

  indicates that you may enter A, B, or C, or you may omit the field.

- In instances where there are nested braces or brackets on the text lines, the following rule applies; Nested operand selection is dependent upon the selection of the operand of a higher level of nesting.

  Example:

  ```
  [Level 1  [Level 2  [Level 3 ]]]
  [filename [filetype [filemode]]]
  ```

  Where the highest level of nesting is the operand that is enclosed in only one pair of brackets and the lowest level of nesting is the operand that is enclosed by the maximum number of brackets. Thus in the previous example, the user has the option of selecting a file by filename only, or filename filetype only or by filename filetype filemode. The user cannot select filetype alone because filetype is nested within filename; and, as noted, our rule states: the higher level of nesting must be selected in order to select the next level (lower level) operand. The same is true if the user wants to select filemode; filename and filetype must also be selected.

- An ellipsis indicates that the preceding item or group of items may be repeated more than once in succession.

  Example
  The representation

  ```
  {spoolid1 spoolid2...}
  ```

  indicates that more than one spoolid may be coded within the braces.

ACNT

Privilege Class: A

Use the ACNT command to process accounting data, create accounting records for logged-on users, and reset the accumulated accounting data. For each user specified, an accounting record is created giving the time used by his virtual machine since the last ACNT command was issued for him or since he logged on. The record produced is identical to the record given when the user logs off of VM/370. Each user receives a message at his terminal that indicates the accumulated accounting data, and the accumulated accounting data for each user is reset to zeros. A user can suppress the accounting message by entering the CP command SET ACNT OFF. The format of the ACNT command is:

```
┌──────────────────────────────────────────────────────────────────────┐
│  ACNT      │  { userid1 userid2 ... }                                │
│            │  { ALL                 }                                │
└──────────────────────────────────────────────────────────────────────┘
```

where:

userid1 specifies the user(s) for which accounting records are to be created. Multiple users may be specified on the command input line.

ALL specifies that accounting records are to be created for all logged-on users.

Responses

The format of the accounting information message is:

```
CONNECT=hh:mm:ss  VIRTCPU=mmm:ss.hh  TOTCPU=mmm:ss.hh
```

where:

CONNECT hh:mm:ss is the actual clock time spent in the user's current terminal session or the time elapsed since the ACNT command was last issued for the user.

VIRTCPU mmm:ss.hh is the processor time spent in executing instructions within the virtual machine.

TOTCPU mmm:ss.hh is the processor time spent in executing instructions within the virtual machine plus processor time used by the control program in maintaining the virtual machine (paging, I/O, etc.).

ATTACH

Privilege Class: B

Use the ATTACH command to logically connect a real device to a virtual machine for that virtual machine's exclusive use, or to logically connect a DASD device for CP access and control. The operator usually issues the ATTACH command to fulfill a user's request. The format of the ATTACH command is:

```
| ATTach | raddr [To] {userid [As] vaddr [R[/O]] [3330V] [VOLID volid]} |
|        |            {SYSTEM [As] volid                               } |
|        |                                                              |
|        | {raddr...    }                                               |
|        | {raddr-raddr} [TO] userid [R[/O]] [3330V]                     |
```

where:

raddr
raddr...
raddr-raddr
is the real address (cuu) of the device to be attached.

On the command line, one real device address, multiple real device addresses, or a range of real device addresses can be specified.

When the multiple or range form of raddr entry is used, the real addresses are identical to the virtual addresses. When the range form of raddr is used (raddr-raddr), the values Cuu-cuu must be entered with no imbedded blanks. The range supplied must be contiguous addresses in ascending order with the second address in the range being greater than the first.

TO

userid is the user identification of the virtual machine that is to receive the device(s). The user must be logged on to receive the device(s). If the class B user is attaching the device to his own virtual machine, the userid can be specified as an asterisk (*). If the optional keyword "TO" is omitted, the userid cannot be "TO" or "T". If a numeric userid of less than four characters is specified with a range of addresses or multiple device addresses, "TO" must be specified. When specifying a userid that could also be a valid hexadecimal device address (for example, virtual machine id CE), the "TO" operand must be specified to distinguish it as a userid.

[AS] vaddr
is the virtual address (cuu) to be assigned to the device. If the optional keyword "AS" is omitted, vaddr cannot be "A" or "AS". For use of the vaddr operand, see Usage Note 6.

R/O
is optionally used for DASD devices that are attached to a user. This allows the system operator to apply read-only protection to a user's disk. The R/O operand may be truncated to R. The R/O operand is ignored for all devices for which it does not apply whenever more than one address is specified on the command line.

[TO] SYSTEM

specifies, along with volid, that the DASD device is to be made available to CP for accessing users' disks and/or for paging and spooling functions.

[AS] volid

is the volume label of the volume on the raddr being attached. This operand has the following meanings when used with the various forms of the ATTACH command:

- When used to attach an available raddr to a userid and the raddr is a 3330V, the MSS volume with the label volid is mounted on the raddr during the attach.
- When used to attach an raddr to the system and the raddr is not a 3330V, the volume label of the volume on raddr is verified.
- When used to attach an available raddr to the system and the raddr is a 3330V, the MSS volume with the label volid is mounted on the raddr during the attach.
- When used to attach an raddr to a userid and the raddr is not a 3330V, the volid operand is invalid.

3330V

specifies that the vaddr is to be treated as a 3330V. The VM/370 control program assumes the virtual machine contains MSS device support. Consequently, MSS cylinder fault support is not invoked for the vaddr.

VOLID volid

VOLID is a required keyword and 'volid' is the volume label for the 3330V being attached.

Usage Notes:

1. It is possible to attach a disk to "SYSTEM" and attach the same disk to a user without first detaching the disk if the disk was not accessed while it was attached to the system.

2. A 370x cannot be used by VM/370 as a 370x unless it has been loaded by VM/370, DOS/VS or OS/VS with 270x EP program and VM/370 was generated with it specified as a 2701, 2702, or 2703. The ATTACH and DETACH commands reset the "loaded" status.

3. The ATTACH command should not be issued for a 370x in PEP mode unless all emulator lines have been disabled by the DISABLE command.

4. If the device being attached is a 2305 then both the real device address and the virtual device address must be specified as the first exposure on the 2305 (that is, device address 0 or 8). When attaching or detaching a dedicated 2305 to a user, all eight exposures are processed.

5. If the virtual machine to which the device is being attached has the ECMODE option, any address up to X'FFF' is valid as 'vaddr'; otherwise, any address up to X'5FF' is valid.

6. The hexadecimal address of the virtual device (vaddr) that is mapped to the real device must be selected with care so that conflict and contention caused by control unit I/O interface protocol is avoided. Omitting this parameter causes the assigning of the raddr as a default.

The following is an example of ATTACH statements that can cause operational conflict.

```
att 10f to jones as 20c   (10f is a real 3211)
att 20e to jones as 20d   (20e is a real 3277 terminal)
```

In the above example, the virtual addresses of both the 3211 and the graphic device indicate the use of the same channel and control unit. By definition, the devices are virtual and therefore will share one common virtual control unit (VCUBLOK) in CP.

The real 3211 printer operates on a nonshared subchannel, and the real 3270 device is designed for shared subchannel operations. Both of these real devices are mapped to the same VCUBLOK. Thus, the subsequent processing of a channel program can result in a hung or busy condition (caused by a conflict in virtual-to-real I/O control procedures).

Therefore, when defining devices, make sure the devices are defined (and separated) within their own control unit range.

Note: This same type of virtual device addressing conflict can occur with the use of any command or process that alters virtual address(es) in the user's VM/370 directory entry.

If a device containing minidisks is mounted and readied after VM/370 has been loaded onto the system, the system operator must attach the device to the system. Subsequent logon operations by virtual machines can link to this disk if provided by their directory entry. Users who had been previously logged on to VM/370 (who possess a directory entry for the minidisk) must issue, subsequent to the DASD attach operation by the system operator, the LINK command for access to the device.

7. When multiple devices or a range of real devices are specified on the ATTACH command line and the syntax is correct, all real device addresses on the command are processed regardless of whether errors are encountered when attempting to attach these devices. Therefore, if the following command was issued

   ```
   attach 108-10F user1 r
   ```

   and real devices 10A and 10D were not ready and online, the following devices would be successfully attached

   ```
   108,109 10B,10C, 10E,10F
   ```

   Informational messages inform the operator of those devices that were successfully attached. Error messages identify the devices that are not successfully attached.

8. To avoid the repetitious attachment of many devices to a virtual machine, it might be to the user's advantage to have these devices attached to his virtual machine via the DIRECTORY DEDICATE statement. Thus, by fulfilling logon requirements, these devices (if online) are automatically attached to the virtual machine.

In the case of virtual DASD volumes associated with the 3850 Mass Storage System, the ATTACH command allows 3330V devices to be attached to a virtual machine without requiring a volume to be mounted. This is necessary because staging of cylinder 0 on a virtual volume is not accomplished until a MSC mount order is issued associating the virtual unit address with a volume serial. Even if the mount order was issued prior to the ATTACH, the host system may change the data staged on a virtual vclume by issuing another MSC mount order.

9. If R/O is specified with a range of addresses or with multiple addresses, this operand applies only to DASD addresses. The R/O operand is ignored for all other device addresses specified on the command line.

10. Do not attach virtual 3330 volumes (3330V) to the SYSTEM; 3330V cannot be used for minidisks, CP residence, paging, or spooling activity.

Responses

Messages are sent to the user, the operator who issued the command (if different from the user), and the primary system operator (if different from the operator who issued the command), notifying them that the ATTACH was successful.

```
type raddr ATTACHED
raddr... ATTACHED
raddr-raddr ATTACHED
```

Depending upon the form used for the command line, the appropriate message indicates to the user the real device or devices attached to his virtual machine. If multiple addresses or a range of addresses are specified, the device type is not displayed.

```
type raddr ATTACHED TO userid vaddr
type raddr ATTACHED TO SYSTEM volid
raddr... ATTACHED TO userid
raddr-raddr ATTACHED TO userid
```

Depending on the form used by the class B operator for command line entry, the appropriate response indicates the real device(s) that are attached to CP or a specified user's virtual machine (and, if appropriate, the virtual address of the device). If multiple addresses or a range of addresses are specified, the device type is not displayed.

```
type raddr ATTACH TO SYSTEM volid BY userid

    -- or --

type raddr ATTACH TO userid vaddr BY userid
```

is the response to the primary system operator (if he did not issue the command) indicating that the real device has been attached either to the specified user's machine at virtual address vaddr or to the CP system by the specified class B user.

Note: In the above responses, "type" is one of the following:

Type	*Meaning*
DASD	Direct access storage device
TAPE	Magnetic tape
LINE	Communication line
RDR	Card reader
PRT	Line printer
PUN	Card punch
GRAF	Graphics device
CONS	Console
CTCA	Channel-to-channel adapter
CTLR	3704 or 3705 Communications Controller
DEV	Any other device

ATTACH CHANNEL

Privilege Class: B

Use the ATTACH CHANNEL command to dedicate all devices on a particular channel to a specified user. The addresses by which the virtual machine user refers to these devices must be the same as the real addresses (he cannot use minidisks). You do not have to attach the devices individually on the attached channel to the specified virtual machine user; he may use any or all of the devices. However, whether idle or not, you cannot attach the devices on the channel to another user. The virtual machine must have all I/O stopped before executing the command. The format of the ATTACH CHANNEL command is:

```
┌──────────────────────────────────────────────────────────────────────┐
│           │                    ┌      ┐                              │
│  ATTach   │  CHANnel c [To]    │userid│                              │
│           │                    │*     │                              │
│           │                    └      ┘                              │
└──────────────────────────────────────────────────────────────────────┘
```

where:

CHANNEL c

is the real address of the channel to be attached to the user. CHANNEL is a required reserved word.

```
   ┌      ┐
TO │userid│
   │*     │
   └      ┘
```

is the user identification of the virtual machine that is to use the dedicated channel. The user must be logged on in order to get control of the channel. The userid can be specified as an asterisk (*) to indicate that the channel is to be attached to the virtual machine of the class B user issuing the command. If the optional keyword TO is omitted, the userid cannot be TO or T.

Note: A DETACH CHANNEL command followed immediately by an ATTACH CHANNEL command may result in a message stating that the specified channel is in use. DETACH CHANNEL processing causes I/O to be asynchronously scheduled for each dedicated device; DETACH CHANNEL processing does not wait for all I/O to complete prior to returning control to the terminal operator. A subsequent ATTACH CHANNEL command may fail due to this timing situation.

Responses

CP sends messages to the user's terminal on the virtual machine to which the channel is attached, and to the operator who issued the command.

CHANNEL c ATTACHED

is the response to the user who receives dedicated access to the channel.

CHANNEL c ATTACH TO userid

is the response to the operator who issued the command.

CHANNEL c ATTACH TO userid BY operator

is the response to the primary system operator if someone else issued the command.

AUTOLOG

Privilege Class: A or B

Use the AUTOLOG command to log on any virtual machine that is defined in the VM/370 directory.

The virtual machine logged on operates in disconnect mode. The virtual machine being logged on with the AUTOLOG command must have an automatic IPL defined in its directory and is allowed to issue one read to its virtual console. The same restraints that apply to any disconnected machine also apply to virtual machines logged on with the AUTOLOG command. The format of the AUTOLOG command is:

```
| AUTOLOG |   userid password   [variable data]                     |
```

where:

userid   is the user identification of the virtual machine to be logged on.

password is the password for the virtual machine to be logged on.

variable data
is the information that is passed to the virtual machine when a virtual SIO is issued to the virtual machine's console. Up to 108 bytes of data may be placed in this field. If no variable data is entered, the field is padded with blanks. Note, embedded line-edit characters in this file are treated as data.

Usage Notes:

1. AUTOLOG is automatically invoked after VM/370 initialization if the VM/370 directory contains a userid of AUTOLOG1. Then, with a suitable PROFILE EXEC, other virtual machines can be logged on and brought online (for example, the batch virtual machine).

2. If LOGON journaling is not being used, and an AUTOLOG command that includes a password is issued from a virtual machine, CP determines if the password is correct. If the password is incorrect, a count of such incorrect passwords is maintained by CP. When a total of four incorrect passwords have been entered, an error message is issued and the AUTOLOG command from the virtual machine is disallowed. AUTOLOG can still be issued directly from the terminal (that is, in CP console mode).

3. If LOGON journaling is ON, the count of incorrect passwords is incremented without regard to the origin of the command. When the installation defined limit of invalid AUTOLOG passwords is reached, (as specified in the SYSJRL macro of DMKSYS) AUTOLOG is disallowed regardless of how it is entered.

4. If password suppression has been generated, the password may be included on the command line only if the AUTOLOG command is in an EXEC or MODULE running under CMS. Otherwise, all data following userid is assumed to be variable data and the password is read from the terminal.

Responses

```
hh:mm:ss AUTO LOGON *** userid USERS=nnn BY {SYSTEM }
                                            {userid2}
```

is the response to the system operator upon successful issuance of the AUTOLOG command.

where:

*** indicates no assigned line address for the virtual console.

userid is the identification of the virtual machine initiated by AUTOLOG.

USERS=nnn is the number of logged-on users.

SYSTEM indicates that the AUTOLOG process is part of VM/370 system initialization.

userid2 identifies the virtual machine that invoked the AUTOLOG command.

BACKSPAC

Privilege Class: D

Use the BACKSPAC command to restart or reposition the current output on a real punch or printer. If the BACKSPAC command is issued while an intervention required condition is pending on the printer or punch, the interrupted channel program is terminated and the backspace is performed. The format of the BACKSPAC command is:

Printer BACKSPAC Format

```
┌──────────────────────────────────────────┐
│            │        ┌              ┐ │
│ BAckspac   │ raddr  │File          │ │
│            │        │pages [EOF]   │ │
│            │        │ 1            │ │
│            │        └              ┘ │
└──────────────────────────────────────────┘
```

Punch BACKSPAC Format

```
┌──────────────────────────────────────────┐
│            │                             │
│ BAckspac   │ raddr [File]                │
│            │                             │
│            │                             │
│            │                             │
└──────────────────────────────────────────┘
```

where:

raddr is the address (cuu) of the real unit record device that is to be backspaced. This address must be that of a punch or a printer.

FILE specifies that the output spool printer or punch file is to be restarted from the beginning.

pages specifies the number of pages to be backspaced on the printer. If the field is not specified, the printer is backspaced to the beginning of the page being printed when the command is received.

EOF specifies that printing is to resume from the end of the spool file. Backspacing occurs the requested number of pages from the end of the spool file. Printing is restarted at this location.

Usage Notes:

1. Pages for backspacing are counted starting from the end of the current spooling buffer. Output containing a small number of lines per page may require a count of more than one to backspace one page.

2. Pages are counted in BACKSPAC operations by looking for skip to channel 1 specifications (page ejects). If a file does not contain skip to channel 1 specifications, processing of the file is restarted regardless of the number of pages indicated. Counting begins at the end of the current buffer.

Responses

```
{PRT} raddr BACKSPACE userid FILE = file RECDS = norecs COPY = nn a typ
{PUN}
```

where:

raddr   is the real device address.

userid   is the identification of the user whcse file was active and backspaced.

file   is the unique, system-assigned spoolid number by which VM/370 refers to the file.

norecs   is the total number of logical records in the file.

nn   is the remaining number of copies of the file to be created, where 01 is the last copy.

a   is the spool file class.

typ   is the originating device type, PRT, CCN, or PUN.

CHANGE

Privilege Class: D

Use the CHANGE command to alter one or more of the external attributes of a closed spool file or files. The current attributes of a file may be determined by issuing a QUERY command. In order to change an output file, the file must have been closed but not yet selected for printing or punching. An input (READER) file may be changed at any time before it is opened, that is, before the first read is issued for the file. The format of the CHANGE command is:

```
┌────────────────────────────────────────────────────────────────────────────┐
│        │          /          ┌CLass c1┐                          \          │
│ CHange │         │  Reader   │spoolid │ CLass c2                  │         │
│        │         │           └ALL     ┘                           │         │
│        │         │                                                │         │
│        │         │           ┌CLass c1┐┌CLass c2         ┐1       │         │
│        │         │  PUnch    │spoolid ││COpy nn          │        │         │
│        │         │           └ALL     ┘│┌      ┐         │        │         │
│        │         │                     ││HOld  │         │        │         │
│        │         │                     ││NOHold│         │        │         │
│        │         │                     │└      ┘         │        │         │
│        │         │                     │DIst distcode    │        │         │
│        │         │                     │┌     ┐          │        │         │
│        │         │                     ││SYS  │          │        │         │
│        │         │                     ││NOSYS│          │        │         │
│        │         │                     └└     ┘          ┘        │         │
│        │         │                                                │         │
│        │┌      ┐ │           ┌CLass c1┐┌CLass c2          ┐1      │         │
│        ││userid│ <  Printer  │spoolid ││COpy [*] nn       │       > [NAme[fn[ft]]] │
│        ││SYSTEM│ │           └ALL     ┘│┌      ┐          │       │ [    {dsname}] │
│        │└      ┘ │                     ││HOld  │          │       │         │
│        │         │                     ││NOHold│          │       │         │
│        │         │                     │└      ┘          │       │         │
│        │         │                     │DIst distcode     │       │         │
│        │         │                     │┌     ┐           │       │         │
│        │         │                     ││SYS  │           │       │         │
│        │         │                     ││NOSYS│           │       │         │
│        │         │                     │└     ┘           │       │         │
│        │         │                     │┌                 ┐│      │         │
│        │         │                     ││FLash  name nn   ││      │         │
│        │         │                     ││CHars  name      ││      │         │
│        │         │                     ││MOdify name      ││      │         │
│        │         │                     ││FCB    name      ││      │         │
│        │          \                    └└                 ┘┘     /          │
├────────────────────────────────────────────────────────────────────────────┤
│ 1One of these options must be chosen; however, more than one may be         │
│  specified and they may be combined in any order.                           │
└────────────────────────────────────────────────────────────────────────────┘
```

where:

userid is the user identification for the user whose spool files are to be changed by the spooling operator.

SYSTEM allows the spooling operator to manipulate all spool files in the system regardless of userid.

READER
RDR
: indicates that reader files are to be changed.

PRINTER
PRT
: indicates that printer files are to be changed.

PUNCH
PCH
: indicates that punch files are to be changed.

CLASS c1 designates an existing class. "c1" is a one-character alphameric field from A to Z, 0 to 9 or *.

spoolid is spoolid number of the file that is to be changed. Each file has a unique spoolid.

ALL changes all files belonging to the user.

CLASS c2 designates the new class of the file.

HOLD prevents the printing, punching, or reading of a file by putting it in a user HOLD status until released by the NOHOLD operand.

NOHOLD releases the specified spool file currently held by the user HOLD status.

DIST distcode

changes the distribution code of the file. "distcode" is a one to eight character alphameric identification that appears on the output separator of printer or punch output instead of the identification specified in the VM/370 directory. The distribution code is changed for this file only, and does not affect other files nor change the normal user code.

SYS prevents the printing or punching of the file by putting it in a system hold status until released by using NOSYS.

NOSYS releases the specified spool file currently being held by a system hold status.

Note: A system hold has no effect on the processing of reader files.

NAME $\left\{\begin{matrix}\text{fn [ft]}\\\text{dsname}\end{matrix}\right\}$

changes the name of the file. For CMS files, the specified filename (and, optionally, filetype) become the new file identification. For a non-CMS file, the specified dsname (a 1 to 24-character field) becomes the new file identification.

Note: If the NAME function is selected and only fn is specified, ft is set to blanks. The dsname specification allows a user to specify a file name suitable for OS such as SYS1.SYSLIB.MYMAC.

FLASH name nn

specifies the flashing status of the file. The name identifies a forms overlay sheet contained in the 3800 printer. Name may be varied from one-to-four characters. The number nn, specifies the number of copies to be flashed. This flash count may range from 00 to 99. A flash count of zero is the default, which indicates flashing is not desired.

Note: The FLASH name must match the FLASH name specified by the VM operator in the START command, unless the flash count is zero, in which case the file match is found by class of file only.

CHARS name
: is the name of the character arrangement table used for printing the file. Name may be from one-to-four characters in length. If a character arrangement table is not specified for a file, the default table used for the separator page on the selected printer will be used to print the file.[1]

MODIFY name
: is the name of the copy modification module used for printing the file. Name may be from one-to-four characters in length. If not specified, copy modification will not be used for printing the file.[1]

FCB name
: is the name of the forms control buffer used for printing the file. Name may be from one-to-four characters in length, or the unique numbers 6, 8, or 12. These specifications mean: print the entire spool file at 6, 8, or 12 lines per inch, regardless of the size of the paper that is currently loaded. Use of the unique numbers will never cause an FCB-forms mismatch. If not specified, the value specified for the separator page in the START command is used.[1]

COPY [*] nn
COPY nn
: alters the number of copies for the file. The value of nn (number of copies) must be a number from 1 through 99. For nn less than ten, the leading zero is optional. When the COPY parameter includes an asterisk, the file is transmitted to the 3800 printer only once. The 3800 performs the duplication internally one page at a time. (For example, if COPY*3 is specified, the page numbers are copied as 111 222 333 ...). Otherwise, the file is transmitted once for each copy desired, with VM/370 indicating the copy number to the 3800.

Responses

$\left\{\begin{array}{c}\text{nnnn}\\ \text{NO}\end{array}\right\}$ FILES CHANGED

is the response to the user who issued the CHANGE command.

This response is suppressed if the virtual machine user has issued the SET IMSG OFF command. The system operator monitoring the RSCS virtual machine for his installation should not specify this option because information messages are required for accurate monitoring of RSCS system activity.

Note: Depending upon the options chosen and the combinations used, the actual "changing" of some files may not have occurred because they had the new class prior to issuing the CHANGE command. The response is an indication of the number of referenced files.

[1] A specification of NULL will nullify any previous setting.

DCP

__Privilege Classes__: C or E

Use the DCP command to display the contents of real storage locations at the terminal. In attached processor system configurations, the values of PSA for the main and the attached processor can also be disclosed.

If an invalid operand is entered, the DCP command terminates. However, any previous valid operands are processed before termination occurs. The format of the DCP command is:

```
┌─────────────────────────────────────────────────────────────────────┐
|               | ┌           ┐┌                     ┐               |
|               |/|MLhexloc1||                      |\              |
|               ||NLhexloc1||                      ||              |
|               ||MThexloc1||                      ||              |
|               ||NThexloc1||      ┌          ┐    ||              |
| DCP           || Mhexloc1||{-}| hexloc2  |    ||              |
|               |<| Nhexloc1||{:}|   END    |    |>              |
|               || Lhexloc1||    └          ┘    ||              |
|               || Thexloc1||                      ||              |
|               ||  hexloc1||                      ||              |
|               ||    0     ||     ┌          ┐    ||              |
|               ||└         ┘|{.}|bytecount|    ||              |
|               ||           |    |   END    |    ||              |
|               |\           └    └          ┘   ┘/              |
└─────────────────────────────────────────────────────────────────────┘
```

__where__:

MLhexloc1
NLhexloc1
MThexloc1
NThexloc1
Mhexloc1
Mhexloc1
Lhexloc1
Thexloc1
hexloc1
0

specifies the first storage location to be displayed. If hexloc1 is the only operand, it specifies the only storage location to be displayed. If hexloc1 is not specified, L or T must be specified, and the display begins with storage location 0. If hexloc1 is specified and L or T is not specified, the display is in hexadecimal. T specifies that an EBCDIC translation is to be included with the hexadecimal display. L specifies that the display is to be in hexadecimal only. If hexloc1 is followed by a period and is not on a fullword boundary, it is rounded down to the next lower fullword.

In attached processor systems, M specifies that the addresses are interpreted as though the main processor generated them. Thus, M0 refers to location 0 of the PSA of the main processor. The N prefix specifies that the addresses be interpreted as though the attached processor generated them. The absence of M or N results in the address being interpreted as an absolute address. N is valid only if the attached processor is operational. M is valid if the system has been generated for attached processors. The prefix M or N is not meaningful for uniprocessor systems.

```
      ┌       ┐
 ┌ - ┐ |hexloc2|
 └ : ┘ | END   |
      └       ┘
```

specifies that a range of locations is to be displayed. To display the contents of one or more storage locations by specified storage address location the "-" or ":" must be used. The hexloc2 operand must be 1- to 6-hexadecimal digits; leading zeros need not be specified. In addition, The hexloc2 operand must be equal to hexloc1 and it should not exceed the size of real storage. If END is specified, real storage from hexloc1 through the end of real storage is displayed. If hexloc2 is not specified, END is the default. Note that this occurs only if a dash (-) or a colon (:) follows the first operand.

```
   ┌          ┐
{.}|bytecount|
   |  END     |
   └          ┘
```

is a hexadecimal integer designating the number of bytes of real storage (starting with the byte at hexloc1) to be displayed on the terminal. The sum of hexloc1 and the bytecount must be an address that does not exceed the size of real storage. If this address is not on a fullword boundary, it is rounded up to the next higher fullword. The bytecount operand must be a value of 1 or greater and may not exceed six hexadecimal digits.

<u>Usage</u> <u>Notes</u>:

1. Normally, a user defines or should define the beginning and ending locations of storage in the following manner:

   ```
   dcp Lhexloc1-hexloc2
   dcp Thexloc1-hexloc2
   dcp hexloc1:hexloc2
   dcp hexloc1.bytecount
   dcp hexloc1:hexloc2 hexloc1.bytecount
   ```

   Blanks must not be entered between the limit or range symbols (: or - or .) or any of the operands except for the blank or blanks between the command name and the first operand. A blank is also required between each set of operands when more than one set of operands are entered on one command line.

   However, if a blank immediately follows the designated type character (T or L), DCP displays all of real storage. If the next operand is either a colon (:), a dash (-), or a period (.) followed by a blank character, the system again defaults to a display of all storage locations as this operand assumes a second set of operands.

2. Blanks separate operands or sets of operands if more than one operand is entered on the same command line. Blanks should not occur on the right or left of range or length symbols, unless it is intended to take the default value of the missing operand defined by the blank.

The following are examples of DCP entries that produce full storage displays.

```
dcp 1          dcp 1-         dcp 0-         dcp t:end
dcp t          dcp 1:         dcp 0:         dcp t:end
dcp -          dcp t:         dcp 1-end      dcp 0:end
dcp :          dcp 1.         dcp t-end      dcp 1.end
dcp .          dcp t.         dcp 0-end      dcp 0.end
```

The following displays all of storage three times because of the embedded blanks:

```
dcp 1 . t
```

For default values (that is, the beginning and ending of storage) to be applicable, one operand must be specified.

3. In attached processor applications, the PSA values for the main and the attached processor are not located at absolute page zero but are displacements from it. Therefore, prefixing the hexadecimal location with an M or N will add the hexadecimal value to the displacement of the location of the assigned PSA for the designated processor; for example,

   dcp m40.8 Displays the CSW of the main processor.
   dcp n78-7f Displays the I/O new PSW of the attached processor.

<u>Responses</u>

Requested locations are displayed in the following format:

xxxxxx = word1 word2 word3 word4 [key] *EBCDIC translation*

<u>where</u>:

xxxxxx is the real storage location of word1.

word1 is displayed (word-aligned) for a single hexadecimal specification. Up to four words are displayed on a line. If required, multiple lines are displayed.

EBCDIC translation
is displayed aligned to the next lower 16-byte boundary if Thexloc is specified. Note that nonprintable characters are displayed as a period (.). If the location is at a 2K page boundary, the key for that page is also displayed. The output can be stopped and the command terminated by pressing the ATTN key (or its equivalent).

DEFINE

Privilege Class: B

Use the DEFINE command to redefine the status of a 3330V volume. For example, a 3330V may have been defined as VIRTUAL allowing it to be dedicated to a specific virtual machine. However, if that virtual machine is not logged on, you can redefine the 3330V as SYSVIRT to make it available to the control program for mounting system volumes. The format of the DEFINE command is:

```
┌──────────────────────────────────────────────────────────────┐
│ DEFine    │    ⎧Sysvirt⎫   raddr1  [-raddr2]                  │
│           │    ⎩Virtual⎭                                      │
└──────────────────────────────────────────────────────────────┘
```

where:

SYSVIRT specifies the desired status of the 3330V. This status is reserved for CP. VM/370 system volumes that are MSS 3330V volumes can be mounted on SYSVIRT 3330V devices.

VIRTUAL specifies the desired status of the 3330V. 3330V devices that are dedicated or attached must use the VIRTUAL feature.

raddr1 [-raddr2]
specifies the device address or range of addresses to be defined. If a range is specified, all 3330V devices in the range are set to the desired status.

Usage Notes:

If an MSS volume is mounted and not currently in use on a 3330V that is redefined, the control program cuases the volume to be demounted before the DEFINE command completes. An MSS volume demount should be forced to redefine the status of the 3330V on which the volume is mounted.

Responses

Responses are sent to the operator who issued the DEFINE command. The responses show the device status following execution of the command. Possible responses are:

raddr NOT KNOWN
The raddr is not a valid device address. No action is taken for the raddr.

raddr NOT 3330V
The raddr is not a 3330V. No action is taken for the raddr.

raddr ALREADY SYSVIRT
raddr ALREADY VIRTUAL
The raddr already has the status to which it is to be defined. It is left in its current status.

raddr IS CURRENTLY DEDICATED
The raddr is currently dedicated to a virtual machine. The status is not changed from VIRTUAL to SYSVIRT.

raddr IS OFFLINE. DEFINE WILL BE DONE.
The raddr is currently offline. When it is varied online, it will receive the status specified in the command.

raddr IN USE BY SYSTEM
The raddr is currently attached to the system and there is at least one user minidisk allocated on the volume. The raddr cannot be defined as VIRTUAL.

raddr HAS A MOUNTED VOL. MSS ID NOT AVAILBLE TO DEMOUNT IT
There is a 3330V volume mounted on the raddr. The MSS communicator is not active and the volume cannot be demounted. The status of the raddr is not changed.

raddr - ERROR DEMOUNTING VOLUME
There is a 3330V volume mounted on the raddr. The control program attempted to demount the volume, but MSS encountered an error. The volume is not demounted and the status of the raddr is not changed.

INVALID RANGE
A range of address (raddr1 - raddr2) was specified. Either raddr1 or raddr2 does not exist or is not a 3330V. Status is not changed.

raddr DEFINED AS VIRTUAL
raddr DEFINED AS SYSVIRT
The raddr has been given the specified status. If a volume was previously mounted on the raddr, it is now demounted.

DETACH

Privilege Class: B

Use the DETACH command to remove a real device from the CP system. Volumes in the SYSOWN list, or devices containing minidisks that are in use, cannot be detached. Tape devices are automatically rewound and unloaded when detached. A previously attached device can be detached from a user even if the device is still in use. For devices that were attached by an ATTACH command and then detached by the user, a message is sent to the operator informing him that the device is free. The format of the DETACH command is:

```
| DETach    | {raddr      } [From] {userid}
|           | {raddr...   }        {SYSTEM}
|           | {raddr-raddr}        {  *   }
```

where:

```
{raddr      }
{raddr...   }
{raddr-raddr}
```

specifies the real address (cuu), multiple addresses or range of addresses of the device or devices to be detached.

```
FROM {userid}
     {  *   }
     {SYSTEM}
```

specifies the user identification of the virtual machine to which the real device [raddr] is currently attached. If the optional keyword FROM is omitted, the userid cannot be FROM, FRO, FR or F. If the DETACH command is invoked to detach a device from the class B operator's own virtual machine, an asterisk [*] may be used. SYSTEM indicates that the device is a DASD device in use by CP for allocation and control. When specifying a userid that could also be a valid hexadecimal device address (for example, virtual machine id CE), the FROM operand must be specified to distinguish it as a userid.

Usage Notes:

1. A 370x cannot be used by VM/370 as a 370x unless it has has been loaded by VM/370, DOS/VS, or OS/VS with 270x EP program and VM/370 was generated with it specified as a 2701, 2702, or 2703. The DETACH command resets the "loaded" status.

2. If the device being attached is a 2305, then both the real device address and the virtual device address must be specified as the first exposure on the 2305 (that is, device address 0 or 8). When attaching or detaching a dedicated 2305 to a user, all eight exposures are processed.

3. When multiple addresses or a range of addresses are specified on the DETACH command line, then the affirmative response are issued for devices that were successfully detached. Error messages are issued for devices that cannot be detached.

4. Attempts to detach an active shared device from the system will result in an error message indicating the device is being shared by multiple users. In order to detach the device, the system operator must inform the affected users to detach the device from their virtual systems. The QUERY LINKS commands used by the system operator will inform him of the progress of the virtual machine detachment operation.

5. DOS or OS virtual machine systems should vary the device offline before invoking the DETACH command. Similarly, CMS users should issue the RELEASE command before invoking DETACH.

6. The use of DETACH while a virtual machine is tracing I/O will cause abnormal termination, (DSP004).

Responses

Messages are sent to the user, the operator who issued the command, and the primary system operator (if different from the operator who issued the command), notifying them that the DETACH was successful.

```
(vaddr...     )
<vaddr-vaddr  >
(type vaddr   ) DETACHED BY operator
```

the appropriate response sent to the user if an operator detaches one or more of the user's devices.

```
(raddr...     )
<raddr-raddr  >           r      ┐
(type raddr   ) DETACHED  |userid|
                          |SYSTEM|
                          L      ┘
```

the appropriate response is issued if the user detaches a previously attached device(s) or if the operator detaches the device(s) from a user or the system.

```
(raddr...     )
<raddr-raddr  >           r      ┐
(type raddr   ) DETACHED  |userid| BY operator
                          |SYSTEM|
                          L      ┘
```

the appropriate response is issued to the primary system operator if a class B operator other than the primary system operator issued the DETACH command and the device or devices had been previously attached.

Note: In the above responses (other than multiple real address response or a range of real address responses), the term "type" is one of the following:

Type	Meaning
DASD	Direct access storage device/3330V (MSS 3330 virtual volume)
TAPE	Magnetic tape
LINE	Communication line
RDR	Card reader
PRT	Line printer
PUN	Card punch
GRAF	Graphics device
CONS	Console
CTCA	Channel-to-channel adapter
CTLR	3704 or 3705 communications controller
DEV	Any other device
MSC	3151 port address

CTCA vaddr DROP FROM userid vaddr

is the response if the device detached was a virtual CTCA connected (via the COUPLE command) to another CTCA on the virtual machine specified by the userid. This response is always followed by the response:

```
CTCA vaddr DETACHED
```

DETACH CHANNEL

Privilege Class: B

Use the DETACH CHANNEL command to detach a dedicated channel from a user when the channel path is no longer required for virtual machine operations. If DETACH CHANNEL is issued prior to the normal completion of I/O activities on the dedicated channel path, those activities are terminated. The format of the DETACH CHANNEL command is:

```
┌──────────────────────────────────────────────────────────────────────┐
| DETach     | CHANnel c [From] ⎧userid⎫                               |
|            |                  ⎩  *   ⎭                               |
└──────────────────────────────────────────────────────────────────────┘
```

where:

CHANNEL c

is the real address of the channel that is to be detached.

```
FROM ⎧userid⎫
     ⎩  *   ⎭
```

specifies the user identification of the virtual machine from which the channel is to be detached. If the optional keyword FROM is omitted, the userid cannot be FROM, FRO, FR, or F. If the DETACH CHANNEL command is invoked by a class B user to detach a channel from his own machine, an asterisk (*) may be used.

Note: A DETACH CHANNEL command followed immediately by an ATTACH CHANNEL command may result in a message stating that the specified channel is in use. DETACH CHANNEL processing causes I/O to be asynchronously scheduled for each dedicated device; DETACH CHANNEL processing does not wait for all I/O to complete prior to returning control to the terminal operator. A subsequent ATTACH CHANNEL command may fail due to this timing situation.

Responses

CHANNEL c DETACHED BY operator

is sent to the user from whose virtual machine the channel is being detached.

CHANNEL c DETACHED userid

is sent to the operator issuing the command.

CHANNEL c DETACHED userid BY operator

is sent to the primary system operator if he did not issue the command.

DISABLE

Privilege Classes: A or B

Use the DISABLE command to prevent low-speed communication lines, including 2701, 2702, and 2703 lines (collectively referred to as 270x lines), 370x facilities (in Emulation Program mode only), and locally attached 3270 lines from accessing the VM/370 system.

Note: The remote 3270 configuration must be disabled by using the DISABLE operand with the NETWORK command.

To disable 370x resources in NCP or PEP program mode, use the NETWORK DISABLE command. If the line(s) selected are not active (not dialed into or logged on), the line is disabled immediately. If the selected line is active, it is disabled when the user does a LOGOFF to finish his terminal session or issues a DISCONN command without the HOLD operand specified. An ENABLE command issued to the line before the line is physically disabled reinstates the enabled status, thus negating the DISABLE command. The format of the DISABLE command is:

```
| DISAble  | {raddr...}
|          | {ALL     }
```

where:

raddr... are the address(es) (cuu) of any communication lines the operator wishes to disable.

ALL specifies that all 270x communication lines, 370x facilities in Emulation Program mode only), and locally attached 3270 lines, including the operator's line, are to be disabled.

Note: Issuing a QUERY LINES and NETWORK QUERY command enables you to examine the status of the communication lines.

Line designations may indicate up to 2700 line addresses. These three-digit hexadecimal addresses are checked against the line addresses stored in the system table. If no match is found, an error message is directed to the system console.

Responses

COMMAND COMPLETE

DMCP

Privilege Classes: C or E

Use the DMCP command to print the contents of real storage locations on the user's virtual spooled printer. The output format is eight words per line with EBCDIC translation. Multiple storage locations and ranges may be specified. In a system running with an attached processor, values that are displacements from the assigned locations of the Prefix Storage Area (PSA) of the main or attached processor can also be dumped. To get the output printed on the real printer, the virtual spooled printer must be terminated with a CLOSE command. The format of the DMCP command is:

```
┌──────────────────────────────────────────────────────────────────────┐
|              |/ ┌          ┐┌                  ┐\                     |
|              || |MLhexloc1||                   | |                    |
|              || |NLhexloc1||                   | |                    |
|              || |MThexloc1||     ┌          ┐  | |                    |
| DMCP         | \|NThexloc1||{-}| hexloc2  |  |  |[*dumpid]           |
|              | )| Mhexloc1||{:}| END      |  | |                    |
|              | /| Nhexloc1||    └          ┘  | |                    |
|              |< | Lhexloc1||                   |  >                   |
|              | )| Thexloc1||                   | |                    |
|              | /|  hexloc1||                   | |                    |
|              || |    0    ||                   | |                    |
|              || └          ┘|    ┌          ┐  | |                    |
|              ||             |{.}|bytecount|   | |                    |
|              ||             |   |END      |   | |                    |
|              |\             └   └          ┘  ┘/                     |
└──────────────────────────────────────────────────────────────────────┘
```

where:

MLhexloc1
NLhexloc1
MThexloc1
NThexloc1
 Mhexloc1
 Nhexloc1
 Lhexloc1
 Thexloc1
  hexloc1
   0

specifies the first storage location to be dumped. If hexloc1 is the only operand, it specifies the only storage location to be dumped. If hexloc1 is not specified, L or T must be specified and dumping starts with storage location 0. An EBCDIC translation is included with the dump contents. If hexloc1 is followed by a period and is not on a fullword boundary, it is rounded down to the next lower fullword.

In attached processor systems, M specifies that addresses be interpreted as if the main processor generated them. Thus, M0 refers to location 0 of the PSA of the main processor. The N operand specifies that addresses be interpreted as if the attached processor generated them. The absence of M or N results in the address being interpreted as an absolute address. N is valid only if the attached processor is operational. M is valid if the system has been generated for an attached processor. The prefixes M or N are not meaningful for a uniprocessor generated system.

```
     r        ┐
{ - } |hexloc2|
{ : } |END    |
     L        ┘
```

is a range of real storage locations to be dumped. To dump to the end of real storage, hexloc2 may be specified as END or not specified at all, in which case END is assumed by default.

```
   r          ┐
{.}|bytecount|
   |END      |
   L          ┘
```

is a hexadecimal integer designating the number of bytes of real storage (starting with the byte at hexloc1) to be typed at the printer. The sum of hexloc1 and the bytecount must be an address that does not exceed the size of real storage. If this address is not on a fullword boundary, it is rounded up to the next higher fullword. If the "." is used for a range, hexloc2 is defined as the number of hexadecimal storage locations (in bytes) to be dumped starting at hexloc1. If hexloc2 is specified as a length in this way, it must have a value such that when added to hexloc1 it will not exceed the storage size.

*dumpid is specified for identification purposes. If specified, it becomes the first line printed preceding the dump data. Up to 100 characters with or without blanks may be specified after the asterisk prefix. If dumpid is specified, hexloc2 or bytecount must be specified. The asterisk (*) is required to identify the dumpid.

Usage Notes:

1. Normally, a user defines beginning and ending dump locations in the following manner:

   ```
   dmcp Lhexloc-hexloc

     -- or --

   dmcp hexloc.bytecount
   ```

   There must not be any blanks between length or range symbols (: or - or .) or between any of the operands except for the blank(s) between the command and the first operand. A blank is also required between each set of operands when more than one set of operands are entered. Note that only one period (.), colon (:), dash (-) or no delimiter may be used within each set of operands.

   If, however, a blank immediately follows the designated type character, the default dump starting and ending locations are assumed to be the beginning and/or end of virtual storage. Similarly, if the range or length symbol separates the first character from a blank or END, all of real storage is dumped.

2. Blanks separate operands or sets of operands if more than one operand is entered on the same command line. Blanks should not occur on the right or left of the range or length symbol, unless it is intended to take the default value of the missing operand defined by the blank. Thus, all of the following produce full storage dumps.

```
dmcp 1         dmcp 1-        dmcp t.        dmcp t-end
dmcp t         dmcp t-        dmcp 0-        dmcp 0:end
dmcp -         dmcp 1:        dmcp 0:        dmcp 1.end
dmcp :         dmcp t:        dmcp 0.        dmcp 0.end
dmcp .         dmcp 1.        dmcp 1-end
```

Each of the following produces three full dumps because of the embedded blanks:

```
dmcp 1 . t
dmcp - : .
```

3. In cases where multiple storage ranges or limits are specified on one command line and the line contains errors, command execution successfully processes all correct operands to the encountered error. The encountered error and the remainder of the command line is rejected and an appropriate error message is displayed.

4. For default operand values (the beginning and ending of storage) to be applicable, at least one operand must be specified.

5. In attached processor applications, the PSA values for the main and the attached processor are not located at absolute page zero but are displacements from it. Therefore, prefixing the hexadecimal location with an M or N will add the hexadecimal value to the displacement of the location of the assigned PSA for the designated processor; for example:

```
DMCP m40.8   Dumps the CSW of the main processor
     n78-7f  Dumps the I/O new PSW of the attached processor.
```

Responses

As the dump proceeds, the following message appears at the terminal indicating that the dump is continuing from the next 64K boundary:

```
DUMPING LOC hexloc
```

where "hexloc" is the segment (64K) address for the dump continuation, such as 020000, 030000, 040000.

If the user signals attention on the terminal *while* the above message is displayed, the dump ends.

COMMAND COMPLETE

indicates normal completion of the dump.

DRAIN

Privilege Class: D

Use the DRAIN command to stop spooling operations on the specified real unit record devices after the file currently being processed has completed. This command brings the spooling system or a specified device to a controlled halt, or halts the activities on a device whose spooling status is to be changed. For example, all unit record devices are usually drained before system shutdown, and a printer must be drained before the contents of the UCS printer buffer are changed.

A device is drained immediately if it is not active when the DRAIN command is issued; otherwise, it is drained when processing of the current file has completed. A drained device can be restarted by the START command. If START is issued to a device before draining is complete, the device does not enter the drained status but continues processing. The format of the DRAIN command is:

```
┌──────────────────────────────────────────────────────────────────────┐
│             │    ┌              ┐                                     │
│  DRain      │    │Reader        │                                     │
│             │    │Printer       │                                     │
│             │    │PUnch         │                                     │
│             │    │raddr...      │                                     │
│             │    │ALL           │                                     │
│             │    └              ┘                                     │
└──────────────────────────────────────────────────────────────────────┘
```

where:

READER
RDR — drains the reader.

READER RDR	drains the reader.
PRINTER PRT	drains the printer.
PUNCH PCH	drains the punch.
raddr	drains the real spooled devices at the specified addresses (cuu). Multiple addresses may be specified.
ALL	drains spooled unit record devices in the system.

Responses

The following response occurs for each device that enters the drained status.

```
{PRT} raddr DRAINED SYSTEM   CLASS = a... {  SEP }
{PUN}                                     {NOSEP }

 RDR  raddr DRAINED SYSTEM
```

where:

raddr is the real device address of the device drained.

a... are those output classes that the drained device was servicing. From one to four classes are displayed for printer or punch devices drained.

ENABLE

Privilege Classes: A or B

Use the ENABLE command to enable the following previously disabled or nonenabled devices so users may access the VM/370 system:

- 3270 (locally attached only) and the 3138, 3148, and 3158 consoles.
- 2701, 2702, and 2703 (270x)
- 3704 and 3705 in Emulation Program mode only

Previously enabled lines are not affected by the ENABLE command. This command affects only low speed communication lines with a data rate not greater than 600 bps (bits per second). For 3704/3705 resources running in NCP or PEP mode and 3270 remote devices, use the NETWORK ENABLE command. The format of the ENABLE command is:

```
| ENable   | {raddr...} |
|          | {ALL     } |
```

where:

raddr... are the addresses (cuu) of the lines that are to be enabled. If more than one address is specified, each must be separated from the others by one or more blanks.

ALL enables all previously disabled or non-enabled locally attached 3270 display terminals and display copy printers (3284/3286 etc.), 3704/3705 (Emulation Program mode only), and 270x lines.

Notes: Do not attempt to enable terminal devices that are not supported as VM/370 virtual machine system consoles. Non-supported console devices can only be used as dedicated devices attached to multiple-access system virtual machines.

Response

```
COMMAND COMPLETE
```

Note: No error indication is provided if no lines are available to be enabled. Issue the QUERY LINES command to see how many lines are enabled.

FLUSH

Privilege Class: D

Use the FLUSH command to halt and immediately purge or, optionally, hold the current output on a specified real unit record device. The device resumes activity with the next scheduled spool file. The format of the FLUSH command is:

```
┌──────────────────────────────────────────────────────────────────────┐
| Flush    | raddr [ALL] [HOld]                                        |
└──────────────────────────────────────────────────────────────────────┘
```

where:

raddr — is the address (cuu) of the real unit record output device whose activity is to be terminated (that is, the real printer or punch).

ALL — deletes all copies of the current output file. If more than one copy of the current output file is being printed or punched and this option is specified, all copies are deleted; if ALL is not specified, only the current copy is deleted and the next copy, if any, is printed or punched.

HOLD — places the spool file in system hold status. If this option is specified, the terminated spool file is not purged, but is put in a system hold status and can be reset using the CHANGE command.

Usage Notes:

1. The FLUSH command is usually invoked when it is apparent that a mistake in output queuing has occurred, or if a high priority job must be started before the current job has finished. For example, a file was directed to a printer that is equipped with the wrong printer train to process that file, or a file was misdirected to the real punch rather than a user's virtual punch unit. For example:

   ```
   FLUSH E  HO
   ```

2. Special handling is required if a spool file is started on a real output device that cannot be made ready. When such a condition arises, an error message is directed to the operator, who can recover the file and restart it. To do so, he must enter the following commands:

   ```
   FLUSH raddr [HOLD]
   DRAIN raddr
   ```

   Following the entry of these commands, the "FATAL I/O ERROR" message is issued to the operator and the device is logically offline. No attempt is made by the system to use the device until the operator puts it back online via the VARY command.

If HOLD is specified, the file is placed in system hold status; if HOLD is not specified, the file is requeued and started on another device (if one is available). Both the FLUSH and DRAIN commands must be issued to recover the file.

Responses

```
{PRT} raddr {FLUSHED  } userid FILE = file RECDS = norecs COPY = nn a typ
{PUN}       {FILE HELD}
```

The FILE HELD response is given in addition to the FLUSHED response if the HOLD option is specified.

where:

raddr is the real address of the printer or punch to be affected.

userid is the identification of the user who is to receive the output.

file is the unique spoolid number which identifies the file to the VM/370 system.

norecs is the total number of logical records in the file.

nn is the number of copies remaining for the file, where 01 is the last copy.

a is the spool file class.

typ is the originating device type (PRT, CON, or PUN).

FORCE

Privilege Class: A

Use the FORCE command to force a logoff of any user on the system. The format of the FORCE command is:

```
┌──────────────────────────────────────────────────────────────┐
│ FORCE    │ userid                                            │
└──────────────────────────────────────────────────────────────┘
```

where:

userid is the user identification of the user who is to be logged off.

Responses

The user receives the normal accounting message produced at logoff and a logoff message of the following format:

```
LOGOFF AT hh:mm:ss zone weekday mm/dd/yy BY SYSTEM
```

The primary system operator receives the following message:

```
⎧GRAF raddr⎫
⎨LINE raddr⎬ LOGOFF AS userid USERS = nnn FORCED
⎩DEV  rid  ⎭ DSCONNECT
```

where:

raddr is the real line address.

rid is the real resource identifier.

userid is the identification of the user who was logged off.

nnn is the number of users remaining on the system.

Note: A virtual user may be forced into a disconnected state by the system if an unrecoverable I/O error occurs on the virtual console. The preceeding message will then be issued with DSCONNECT rather than LOGOFF. The user may log on at any time while his disconnected machine is still running. See the _System Logic and Problem Determination Guide_ for further information.

FREE

Privilege Class: D

Use the FREE command to remove a set of spool files belonging to a specified user from a system hold status. A spool file held by a spooling operator can only be freed by a spooling operator; a spool file held by a user can be freed by that user or the spooling operator. A spool file can be in a double hold status and can only be processed after both holds (system and user) are removed.

The FREE command works in conjunction with the HOLD command. It can free only those output files that are in a hold status. The hold status of a file is its pending status; that is, it is neither purged nor called for spooling output. Use HOLD to detain a file or a group of files temporarily to correct or modify a spooling condition. Blocking files from spooled output severely impacts available spooling area.

A particular file may be held from output because of a device restriction, an I/O error on the printer or punch, or a device modification. (For example, a printer may not be available, or it may have an incorrect print train mounted.) When the condition is corrected, the files can then be released for output. The format of the FREE command is:

```
┌───────────────────────────────────────────────────────────────────────┐
│             │              ┌       ┐                                   │
│             │              │Printer│                                   │
│  FRee       │  userid      │PUnch  │                                   │
│             │              │ALL    │                                   │
│             │              └       ┘                                   │
└───────────────────────────────────────────────────────────────────────┘
```

where:

userid is the user whose spool files are to be released from a system hold status. The user need not be logged on, but his files may have previously been held by the HOLD command.

PRINTER
PRT frees the printer files for the specified user.

PUNCH
PCH frees the punch files for the specified user.

ALL frees all printer and punch files for the specified user.

Responses

None.

HALT

Privilege Class: A

Use the HALT command to attempt to terminate any active channel program on a specified real device. VM/370 attempts to reset status by issuing a Reset Immediate command. The format of the HALT command is:

HALT	raddr

where:

raddr is the address (cuu) of the real device that the system operator wishes to stop.

Usage Notes:

1. This command should be used only in extreme cases and after careful consideration. Indiscriminate use of this command may cause unpredictable results.

2. This command does not affect the 3704 or 3705 Communications Controller being run in NCP or PEP mode. See the class A NETWORK command.

3. For device addresses that have multiple I/O paths defined by the VM/370 system generation process, the HALT raddr command causes a HIO to be issued on only that specified path.

Response

DEVICE HALTED

indicates that VM/370 has reset status and halted the device.

HOLD

Privilege Class: D

Use the HOLD command to place user spool files in a system hold status. The spooling operator has the ability to hold the files of any user of the system.

The system hold status for a user is remembered even if the user is not currently logged on. Both the user's files and his hold status can be reset with the FREE command.

Files held by a class D user can only be freed by a class D operator. A spool file can be held by a user or by the system spooling operator. A spool file held by a spooling operator or a user can only be freed by the same spooling operator or user. A spool file can be in a double hold status and can be processed only if both holds (system and user) are removed. The format of the HOLD command is:

```
┌──────────────────────────────────────────────────────────────────────┐
|  HOld      |          ┌        ┐                                     |
|            |          |Printer|                                      |
|            |  userid  |PUnch  |                                      |
|            |          |ALL    |                                      |
|            |          └        ┘                                     |
└──────────────────────────────────────────────────────────────────────┘
```

where:

userid — is the identification of the user whose spool files are to be placed in a system hold status. The user need not be logged on when the ccmmand is issued.

PRINTER
PRT — holds the printer files for the specified user.

PUNCH
PCH — holds the punch files for the specified user.

ALL — holds all the printer and punch files for the specified user.

Responses

None.

INDICATE

Privilege Class: E

Use the INDICATE command to display, at the console, the use of and contention for the major system resources of processor and storage. The users of this command include the general user and the system analyst.

The general user can display the use of and contention for the major system resources of processor and storage. He can also display the number of I/O requests his virtual machine has made and the total amount of resources he has used during his terminal session. If he uses the INDICATE command before and after the execution of a program, the command indicates the execution characteristics of that program in terms of resource usage.

The system analyst can identify the users in queue1 and queue2, the I/O devices they are queued up on, the paging devices that may have been filled, or the execution characteristics of any user and the total amount of resources used by him.

The system analyst can use the data on system resource use and contention to monitor the performance of his system. He can thus be aware of heavy load conditions or low performance situations that may require the use of more sophisticated data collection, reduction, and analysis techniques to resolve these conditions. The format of the Class E INDICATE command is:

```
┌───────────────────────────────────────────────────────────────┐
│              │ ┌                   ┐                           │
│ INDicate     │ │LOAD               │                           │
│              │ │         ┌       ┐ │                           │
│              │ │USER     │*      │ │                           │
│              │ │         │userid │ │                           │
│              │ │         └       ┘ │                           │
│              │ │Queues             │                           │
│              │ │I/O                │                           │
│              │ │         ┌       ┐ │                           │
│              │ │PAGing   │WAIT   │ │                           │
│              │ │         │ALL    │ │                           │
│              │ │         └       ┘ │                           │
│              │ └                   ┘                           │
└───────────────────────────────────────────────────────────────┘
```

where:

LOAD provides an indication on the operating load of VM/370 by displaying values on: the number of users in queue 1 and queue 2, the usage of real storage, and the ratio of active users to users being serviced.

USER [*]
[userid]
allows a system analyst to find out more about the resources used and occupied by his virtual machine, and events that have taken place. USERID allows the system analyst to determine the activity of virtual machines in terms of the resources used and occupied and events that have taken place. Class E users can access data from the VMBLOK of any user currently logged on to the system in their attempts to understand an overload or poor performance situation. The output of this operand is the same

as that of the INDICATE USER * option. For detailed discussion of the class G INDICATE command, see _VM/370 CP Command Reference for General Users_.

QUEUES displays the active users, the queues that they are in, the storage that they are occupying, and the status that they are in. The display indicates those users currently dominating main storage. Users waiting in eligible lists are included in the response because they are contending for main storage and it is only by chance that they were not occupying main storage at the time of the command.

I/O provides information about conditions leading to possible contention within the system. The response gives the userids of all the users in I/O wait state at that instant in time, as well as the address of the real device to which the most recent virtual SIO was mapped. The response, however, does not show the queue of users who issue SIOF to busy devices. Because the response indicates only an instantaneous sample, use the command several times before assuming a condition to be persistent. If it is persistent, use the SEEKS operand of the MONITOR command to conduct a thorough investigation of the suggested condition.

PAGING WAIT
is provided for installations that have 2305s as primary paging devices and other direct access devices as secondary paging devices. A full primary device and subsequent allocation of paging space on the slower device may be responsible for degradation of the system's performance. Use the PAGING WAIT operand when the QUEUES operand shows that a significant proportion of the users in queue 1 and queue 2 are persistently in page wait. The response to the command gives the userids of those users currently in page wait and the number of page frames allocated on drum storage and on disk storage.

PAGING ALL
displays the page residency data of all users of the system (including the system nucleus and pageable routines). The format of the reply message is identical to that of the PAGING WAIT operand.

Usage Notes:

The intent of the INDICATE command is twofold:

1. Provide the operator with a "snapshot" of system activities.

2. Provide a means of determining the execution characteristics of a program with respect to the resources it uses.

In respect to the latter, INDICATE USER should be invoked before and after the execution of a program. If the INDICATE USER command is issued during the execution of the subject program and the program is issuing spooling I/O requests, the results may appear questionable. The reason this occurs is that some fields (specifically, the PCH=nnnnnnn, PTR=nnnnnn and RDR=nnnnnn fields) in the response to the INDICATE USER command are updated once (either at the beginning or at the end of an operation). Other fields in the response are updated more dynamically. Consequently, you should avoid to halting or aborting any spooling operation before its normal termination if the INDICATE USER command is to be issued.

Responses

INDICATE LOAD

```
CPU-nnn%  APU-nnn% Q1-nn Q2-nn  STORAGE-nnn%  RATIO-nn.n
```

CPU-nnn%
: is a smoothed value that indicates the percentage of time that the system is running in the main processor.

APU-nnn%
: is a smoothed value that indicates the percentage of time that the attached processor is running.

Q1-nn Q2-nn
: indicates the contention for CP is represented by smoothed values of the numbers of users in queue 1 and queue 2. The values are maintained by the scheduler.

STORAGE-nnn%
: is a measurement of the use of real storage. It is a smoothed ratio of the sum of the estimated working sets of the users in queue 1 and queue 2 to the number of pageable pages in the system, expressed as a percentage. Because the criterion for allowing a user on the eligible list to enter a queue is that at least 75 percent of his working set size must fit in the available page frames, the value of STORAGE can be more than 100 percent.

RATIO-nn.n
: indicates the scheduler contention ratio; RATIO is a smoothed measure of the contention for real storage, and is defined as RATIO=(E+M)/M. M is the number of users in queue 1 and queue 2 and E is the number of users waiting to be allocated real storage by the scheduler and, therefore, temporarily resident in the scheduler's eligible lists. Thus, RATIO is the ratio of active users to users being serviced, and is 1.0 for optimum response. Optimum response occurs when enough storage is available to accommodate all active users, assuming the CPU can process their commands. If E and M are both 0, the value of RATIO is set to 1.0. Values of RATIO=1.5 and M=10, mean that users are in the eligible lists waiting for the scheduler to allocate real storage space to them and the full discrimination ability of the biased scheduler may be exercised upon them.

```
               ┌      ┐
               |*     |
INDICATE USER  |userid|
               └      ┘
```

```
PAGES: RES-nnnn WS-nnnn READS=nnnnnn WRITES=nnnnnn DISK-nnnn DRUM-nnnn
VTIME=nnn:nn TTIME=nnn:nn SIO=nnnnnn RDR-nnnnnn PRT-nnnnnn PCH-nnnnnn
```

The first line of the response gives all the data from the user's VMBLOCK that is relevant to his virtual machine's paging activity and resource occupancy.

n
: is a decimal number throughout.

RES
: is the current number of the user's virtual storage pages resident in real storage at the time the command is issued.

WS
: is the most recent system estimate of the user's working set size.

READS is the total number of page reads for this user since he logged on or since the last ACNT command was issued for his virtual machine.

WRITES is the total number of page writes for this user since he logged on or since the last ACNT command was issued for his virtual machine.

DISK is the current number, taken at an instant of time, of virtual pages allocated on system disk paging space for this user.

DRUM is the current number, taken at an instant of time, of virtual pages allocated on system drum paging space for this user.

The second line of the response gives user CPU usage and accumulated I/O activity counts since logon or since the last ACNT was issued for his virtual machine.

VTIME is total virtual time for the user.

TTIME is total virtual and simulation time for the user.

SIO is the total number of nonspooled I/O requests issued by the user.

RDR is the total number of virtual cards read. This value is obtained from spool file control information. The total number of cards to be read is incremented once to the RDR=value at the start of each file read operation.

PRT is the total number of virtual cards printed. This value is obtained from spool file control information. The total number of lines to be printed is incremented once to the PRT=value at the end of each file print operation.

PCH is the total number of virtual cards punched. This value is obtained from spool file control information. The total number of cards to be punched is incremented once to the PCH=value at the end of each file punch operation.

INDICATE QUEUES

```
userid aa bb sss/ttt     userid  ........
```

where:

userid is the user identification. (More than one user may be displayed.)

aa is one of the following: E1, E2, Q1, or Q2, indicating the eligible lists or queues that are occupied.

bb is one of the following status indicators:

- RU current RUNUSER in uniprocessor applications. In attached processor applications, RU means current RUNUSER on the main processor.
- RA current RUNUSER on the attached processor. RA does not appear in the response line in uniprocessor applications.
- PG in page wait -- the user is not executing because CP is trying to bring in a page from a paging device.

IO in I/O wait -- the user is in I/O wait because access to the device is not available at the moment.

EX in instruction simulation wait -- the user is waiting for the completion of instruction simulation.

PS in PSW wait -- the user is in an enabled wait state for high-speed I/O devices.

DF in attached processor configurations, the processing of a synchronous (program or SVC) interrupt for this user has been deferred until the system lock is available. DF is not applicable for uniprocessor operations.

Note: In cases where a virtual machine may be in more than one of the above states, only one state is displayed; the state displayed is the first one encountered in the order of priority indicated above.

sss is a hexadecimal number indicating the number of pages resident in real storage.

ttt is a hexadecimal number indicating the number of pages estimated by the dispatcher as the working set of this user.

Note: The lines of the above response are ordered as follows:

1. Q1 and Q2 users in runlist priority order (dispatching priority)
2. Eligible list E1 users in scheduling priority order.
3. Eligible list E2 users in scheduling priority order.

The significance of these values is important in system performance analysis; for details see the VM/370 System Programmers Guide.

NO USERS IN QUEUE

is issued for the QUEUES operand when appropriate.

INDICATE I/O

userid1 xxx userid2 yyy

where:

userid1
userid2 is a user identification name(s).

xxx indicates the real device address.

yyy indicates that two users are waiting for I/O to complete on the device indicated.

Note: In the case where a virtual machine may have issued multiple SIOs, the response indicates the real device address corresponding to the most recent one issued.

NO USERS IN I/O WAIT

is issued for the I/O operand, when appropriate.

```
                  ┌    ┐
                  |ALL |
INDICATE PAGING   |WAIT|
                  └    ┘
```

```
userid1 xxx:yyy   userid2  xxx:yyy
```

where:

userid1
userid2 is user identification name(s).

xxx are the number of pages, in hexadecimal, allocated on drum storage for these users.

yyy are the number of pages, in hexadecimal, allocated on disk storage for these users.

Note: One or more users can be indicated. This sample response indicates two users. If the two users shown in the response were to execute programs of similar characteristics, then userid1 would be expected to experience more page wait than userid2. If users appear to have most of their pages allocated on disk storage, it would be useful to know which users are occupying most of the primary paging device space, and whether or not they are still active. That is, a virtual machine that is executing a large operating system may have been allocated large amounts of primary paging device space at IPL time but then have become inactive. Consequently, the machine is occupying a critical resource but is not using it. If the ALL operand is invoked, then xxx and yyy values are given for all users on the system.

```
NO USERS IN PAGEWAIT
```

is issued for the PAGING WAIT operand, when appropriate.

LOADBUF

Privilege Class: D

Use the LOADBUF command to load the Universal Character Set (UCS) buffer with a specified print chain/train image for the IBM 1403, 3203, or 3211 Printer or the Forms Control Buffer (FCB) with a specified image for the IBM 3203 or 3211 Printer. The device must be drained before the LOADBUF command is issued. This command should be used by the class D operator to load the UCS buffer of a printer under the following conditions:

- Any time the print chain/train is to be changed
- If the buffer has been causing an excessive amount of parity errors
- When the printer or control unit is powered up since the last buffer load
- If the printer was previously attached to a user

The FCB buffer should be loaded:

- Anytime the FCB image is to be changed
- If the printer was previously attached to a user

The LOADBUF command automatically sets the Block Data Check latch for the UCS buffer load. If VER was specified after the command has been executed, the image of the UCS load is printed on the specified printer. The printed output must correspond to the description of the specified buffer load in the publications *IBM 2821 Control Unit*, Order No. GA24-3312, or *IBM 3211 Printer, 3216 Interchangeable Train Cartridge, and 3811 Control Unit Description and Operator's Guide*, Order No. GA24-3543. The format of the LOADBUF command is:

```
┌──────────────────────────────────────────────────────────────┐
│ LOADBUF  │ {raddr  UCS name  [Fold]  [Ver] }                 │
│          │ {raddr  FCB name  [Index  [nn ]]}                 │
└──────────────────────────────────────────────────────────────┘
```

where:

raddr is the address of the printer whose buffer is to be loaded.

UCS
FCB is a required reserved word.

Note: When loading both a print chain image (UCS) and a forms control buffer (FCB), it may be necessary to use two LOADBUF commands for the specified printer.

name is a one- to four-character name of the UCS or FCB image to be loaded. The supplied names are as follows:

- UCS--1403 or 3203

AN	normal AN arrangement
HN	normal HN arrangement
PCAN	preferred character set, AN
PCHN	preferred character set, HN
QN	PL/I - 60 graphics
QNC	PL/I - 60 graphics
RN	FORTRAN, COBOL commercial
YN	high speed alphameric
TN	text printing - 120 graphics
PN	PL/I printing - 60 graphics
SN	text printing - 84 graphics

- UCS--3211

A11	Standard Commercial
H11	Standard Scientific
G11	ASCII
P11	PL/I
T11	Text Printing

- FCB--3211 or 3203

Only one name is provided for an FCB image. For information on how to add more FCB images, see the *VM/370 System Programmer's Guide*.

FCB1 Space 6 lines/inch. Length of page 66 lines

Line Represented	Channel Skip Specification
1	1
3	2
5	3
7	4
7	5
11	6
13	7
15	8
19	10
21	11
23	12
64	9

FOLD loads the UCS buffer with the folding operation code to permit printing of uppercase for lowercase bit configurations. FOLD is optional and valid only for UCS.

Note: The FOLD option causes the control unit to ignore bit positions 0 and 1 of the EBCDIC code when comparing the character codes from the UCS buffer and the print line buffer. Thus, only uppercase characters are printed from either uppercase or lowercase data codes. The FOLD option must be used with discretion since multiple characters may be printed for the same EBCDIC bit configuration depending on the image used.

VER prints the contents of the buffer on the specified printer for verification of the buffer loading function.

Note: If you are using a 3211 with a P11 train, do not use the VER option of the LOADBUF command because a permanent I/O error will result.

INDEX [nn]
starts printing the output in the designated (nn) print position. "nn" must be a number from 1 to 31; a leading zero need not be specified. If INDEX is not specified, no indexing is done. If INDEX is specified without a value, the index value specified in the FCB macro becomes the index value. If INDEX is specified with a value, the specified value overrides the index value in the FCB macro. For a description of the FCB macro and forms control images, see the *VM/370 System Programmer's Guide*. This option is not valid for a 3203 Model 4 or 5 printer.

Responses

The contents of the UCS buffer are printed on the specified printer if VER was specified.

LOCATE

Privilege Classes: C or E

Use the LOCATE command to find the addresses of CP control blocks associated with a particular user, a user's virtual device, or a real system device. The control blocks and their use are described in the VM/370 Data Areas and Control Block Logic. The format of the LOCATE command is:

```
┌──────────────────────────────────────────────────────────────────────┐
| LOCate   |  {userid [vaddr]}                                          |
|          |  {raddr          }                                         |
└──────────────────────────────────────────────────────────────────────┘
```

where:

userid is the user identification of the logged-on user. The address of this user's virtual machine block (VMBLOK) is printed.

vaddr causes the virtual channel block (VCHBLOK), virtual control unit block (VCUBLOK), and virtual device block (VDEVBLOK) addresses associated with this virtual device address to be printed with the VMBLOK address.

raddr causes the real channel block (RCHBLOK), real control unit block (RCUBLOK), and the real device block (RDEVBLOK) addresses associated with this real device address to be printed.

Responses

LOCATE userid

```
VMBLOK = xxxxxx
```

LOCATE userid vaddr

```
VMBLOK    VCHBLOK    VCUBLOK    VDEVBLOK
xxxxxx    xxxxxx     xxxxxx     xxxxxx
```

LOCATE raddr

```
RCHBLOK   RCUBLOK    RDEVBLOK
xxxxxx    xxxxxx     xxxxxx
```

LOCK

Privilege Class: A

Use the LOCK command to permanently lock in real storage selected pages of the pageable CP nucleus or of a user's virtual storage, thus excluding them from future paging activity.

The LOCK command can enhance the efficiency of a particular virtual machine by keeping high activity pages such as virtual page zero in real storage if the reserved page frames option is already in use by some other virtual machine. If the amount of page frames available for paging is limited, the LOCK command should not be issued without the approval of the systems programmer.

Note: If too many pages of real storage are locked, the remaining virtual machines may not have enough available page frames left to operate efficiently. Throughput may then be severely degraded in all virtual machines because of excessive contention for the remaining available page frames.

Once a page is locked, it remains locked until either the user logs off the system or you issue the UNLOCK command for that page. If a user with the "locked pages" option in effect should re-IPL his system or load another system, the locked pages are unlocked and available to the system being loaded. Note that in the attached processor environment, shared pages cannot be locked. Once the virtual=real area is unlocked, a lock cannot be issued to run another V=R machine until VM/370 is reloaded.

In uniprocessor mode you can lock a shared page but, if the shared page becomes nonshared, the system unlocks it. In addition, if a protected shared page is locked and it is changed, the system unlocks it.

The virtual pages locked in processor storage are blocks of 4K (4096) bytes. This block of storage need not represent all of the user's virtual storage. The LOCK command may be issued as many times as required for one virtual machine to lock noncontiguous pages of storage. The remaining virtual machine storage blocks may remain pageable. The format of the LOCK command is:

```
| LOCK     | { userid } firstpage lastpage [MAP]  |
|          | { SYSTEM }                           |
```

where:

userid   is the user identification of a logged-on user.

SYSTEM   locks one or more of the pageable CP pages. The pageable page(s) are locked until released by the UNLOCK command.

firstpage
is the hexadecimal value of the first user page to be brought into storage and locked.

lastpage   is the hexadecimal value of the last user page to be brought into storage and locked. If only one page is to be brought into storage, lastpage must be the same as firstpage.

MAP prints a map of the virtual storage pages locked and the real page that is assigned for each page locked. The response to the MAP operand is:

```
  VIRTPG            REALPG
virt-firstpage   real-firstpage
     .                .
     .                .
     .                .
virt-lastpage    real-lastpage
```

Note: If you omit the MAP operand from a LOCK command and you later want to find the real page, reissue the LOCK command with the MAP operand. No intervening UNLOCK is required.

The SYSTEM and MAP operand are useful when you wish to examine (via DCP) or modify (via STCP) a page that is normally not resident (such as console function routines). The MAP operand locates the real page for display purposes. Page numbers are defined as follows:

Storage Locations (in hexadecimal)	Page Number
000-0FFF	0
1000-1FFF	1
2000-2FFF	2
.	.
.	.
.	.
12000-12FFF	12
.	.
.	.
.	.

Examples

1. For firstpage and lastpage, only the page numbers are specified. For example, to lock USERA's virtual storage locations X'12000' to X'2C000' in real storage, the following command should be given:

   ```
   LOCK USERA 12 2C
   ```

   To lock the first three pages of a USERA's storage, enter:

   ```
   LOCK USERA 0 2
   ```

2. To lock a CP page for display purposes, key in the following lines:

   ```
   LOCK SYSTEM 2A 2A MAP
   VIRTPG REALPG } response to the MAP operand
   02A000 04F000 }
   DCP 4F038-4F048
   etc.
   UNLOCK SYSTEM 2A 2A
   ```

Response

```
COMMAND COMPLETE
```

MESSAGE

Privilege Class: A or B

Use the MESSAGE command to transmit message text to a specified userid or to the primary system operator, or as the primary system operator, to send message text to one or all logged-on users. If the user designated to receive the message is not logged on or has suppressed the receiving of messages, the message is not transmitted and the sender receives a diagnostic message to this effect. If you send a message to ALL you receive a diagnostic message for every user with message typing suppressed. A message that is not received by a user is not saved and must be sent at a later time when the user is receiving messages. The message is displayed at the terminal when the terminal is ready to receive output. If a typewriter terminal, or a display terminal having AUTOREAD set ON, is entering data, the message is held until a carriage return is received. If you want the users to receive messages automatically when they log on, see the class B SET command. The format of the Class A and B MSG command is:

```
┌──────────────────────────────────────────────────────────────────────┐
│  Message  │ ⎧ALL      ⎫  msgtext                                      │
│  MSG      │ ⎨userid   ⎬                                               │
│           │ ⎨*        ⎬                                               │
│           │ ⎩OPerator ⎭                                               │
└──────────────────────────────────────────────────────────────────────┘
```

where:

ALL is for operator use only. It allows him to broadcast a message to all logged-on users.

userid is a single user who is to receive the message.

* the asterisk (*) specifies that the user is sending the message to himself.

OPERATOR specifies that the message is to be sent to the primary system operator regardless of his userid.

msgtext specifies the text of the message that is to be transmitted. As many characters may be entered as will fit on the remainder of the input line.

Responses

```
hh:mm:ss
MSG FROM OPERATOR: msgtext
```

is the format of the message received by a specified users or all users from the system operator.

```
hh:mm:ss
MSG FROM userid: msgtext
```

is the format of the message sent to another user or to the system operator, where userid is the name of the sender.

```
hh:mm:ss
MSG FROM LOGONxxx: msgtext
```

This is the format of a message sent from a user of the VM/370 system to a specified user or to a system operator. This message indicates that the user has not yet achieved logon status. xxx denotes the real line address from which the MESSAGE command was issued.

Note: For all of the above responses:

hh:mm:ss is the time in hours:minutes:seconds when the message was sent to the user.

If the user receiving the message is the primary system operator, the alarm bell at the central computer console rings.

If you are using a 3270, the message puts the screen in a HOLD status and rings the Audible Alarm, if present. Press the cancel key to return to the RUNNING status.

MONITOR

Privilege Class: A or E

EVENT TRACING AND RECORDING

Use the MONITOR command to initiate, override the system generated monitor function, or terminate the recording of events that occur in the real machine. This recording is always active after a VM/370 IPL (manual or automatic). The events that are recorded in the CP internal trace table are:

- External interruptions
- SVC interruptions
- Program interruptions
- Machine check interruptions
- I/O interruptions
- Free storage requests
- Release of free storage
- Entry into scheduler
- Queue drop
- Run user requests
- Start I/O
- Unstack I/O interruptions
- Storing a virtual CSW
- Test I/O
- Halt device
- Unstack IOBLOK or TRQBLOK
- NCP BTU (Network Control Program Basic Transmission Unit)
- Clear Channel operation (CLCH)

Use the trace table to determine the events that preceded a CP system failure. For information on finding and using the internal trace table see, the *VM/370 System Programmer's Guide*

PERFORMANCE MEASUREMENT

Use the MONITOR command to:

- Display the status of the internal trace table, each implemented class of data collection, and the specifications for automatic monitoring using spool files.

- Enable various classes of MONITOR CALL.

- Specify the time intervals for timer-driven data collection.

- Start and stop data collection by MONITOR using tape or spool files.

- Override the specifications for automatic monitoring assigned at system generation time.

- Close the monitor spool files.

- Specify device addresses to be included in or excluded from a selection list for DASD seeks analysis.

For a discussion of the MONITOR command options, see the *VM/370 System Programmer's Guide*. The format of the Class A and E MONITOR command is:

```
 MONitor  |  / Display    ┌     ┐                                          \
          | |             |SPOOL|                                          |
          | |             |TAPE |                                          |
          | |             |ALL  |                                          |
          | |             └     ┘                                          |
          | |  ENable     /PERForm \ 1                                     |
          | |             |RESPonse|                                       |
          | |             |SCHedule|                                       |
          | |             |USER    |                                       |
          | |             |INSTsim |                                       |
          | |             |DAStap  |                                       |
          | |             |SEEKs   |                                       |
          | |             \SYSprof /                                       |
          | |                                                              |
          | |  INTerval   nnnnn  ┌   ┐                                     |
          | |                    |SEC|                                     |
          | |                    |MIN|                                     |
          | |                    └   ┘                                     |
          | |             ┌                                          ┐     |
          | |  STArt      |SPOOL     [to userid] [BUFFS n]           |     |
          | |             |                  ┌              ┐        |     |
          | |             |TAPE    raddr     |MODE ( 800)   |        |     |
          | |             |                  |     {1600}   | [BUFFS n]    |
          | |             |                  |     (6250)   |        |     |
          | <             |                  └              ┘        |     >
          | |             |CPTRACE                                   |     |
          | |             └                                          ┘     |
          | |             ┌       ┐                                        |
          | |  STOP       |SPOOL  |                                        |
          | |             |TAPE   |                                        |
          | |             |CPTRACE|                                        |
          | |             └       ┘                                        |
          | |  CLOSE                                                       |
          | |  AUTOdisk   {ON }                                            |
          | |             {OFF}                                            |
          | |  TIME       (FROM  h1:m1 TO h2:m2)                           |
          | |             {FOR   hh:mm        }                            |
          | |             {ALL                }                            |
          | |             (NONE               )                            |
          | |             ┌      ┐                                         |
          | |  LIMIT  n   |NOSTOP|                                         |
          | |             |STOP  |                                         |
          | |             |SAMPLE|                                         |
          | |             └      ┘                                         |
          | |  SEEKS      (INclude  raddr raddr...)                        |
          | |             {EXclude  raddr raddr...}                        |
          | |             {DELete                 }                        |
          |  \            (DISplay                )                        /
```

[1] Select one or more of the classes, subject to the restrictions listed with the ENABLE operand.

where:

```
DISPLAY   ┌     ┐
          |SPOOL|
          |TAPE |
          |ALL  |
          └     ┘
```

displays the status of the pertinent VM/370 Monitor variables and the status of the internal trace table. SPOOL is the default operand. Regardless of the operand you select, each class of monitor call and its current enabled/disabled state is listed.

If SPOOL is specified, the automatic monitoring specifications are listed, including whether or not automatic monitoring has been requested, its start and stop times, the number of monitor buffers to be used, the userid of the virtual machine to receive the spool file, the spool file record limit and class, and which classes of monitoring are to be enabled.

If automatic monitoring is already in progress, the spool file number is given together with the number of monitor buffer records already written to it.

If the TAPE option is requested, only the status of monitor classes and CPTRACE table is indicated.

If ALL is specified, a combination of SPOOL and TAPE responses is given.

ENABLE enables the specified classes of MONITOR CALL. Each successful completion of this command creates a new value for control register 8. The effect of the MONITOR ENABLE command varies, depending on whether data collection is active or inactive when the command is issued. If data collection is active, the new mask is moved directly into control register 8, replacing the previous mask, and the new mask takes effect immediately. Collection then continues the classes just entered. If data collection is not active, the new mask is saved, and data collection of the specified classes does not begin until a MONITOR START is issued.

PERFORM samples system resource usage data by accessing system counters of interest to system performance analysts.

RESPONSE collects data on terminal I/O. Simplifies analysis of command usage, user response time, and system response time. It can relate user activity to system performance. This class is invalid and no data can be collected for it unless the system programmer changes the LOCAL COPY file and reassembles DMKMCC.

SCHEDULE collects data about scheduler queue manipulation. Monitors the flow of work through the system, and indicates the resource allocation strategies of the scheduler, which is one of the key functions of the system.

USER periodically scans the chain of VMBLOCKs in the system, and extracts user resource utilization and status data.

INSTSIM records every virtual machine privileged instruction handled by the control program (CP). Because simulation of privileged instructions is a major source of overhead, this data may lead to methods of improving performance. If the VMA feature is active, the number of privileged instructions handled by the control program are reduced for those virtual machines that are running with the feature activated.

DASTAP periodically samples device I/O activity counts (SIO), for tape and DASD devices only.

SEEKS collects data for every I/O request to DASD devices. Reveals channel, control unit, or device contention and arm movement interference problems.

Note: No meaningful information is extracted for V=R regions if SET NOTRANS is in effect.

SYSPROF adds to the DASTAP and SCHEDULE classes to support a more detailed "profile" of system performance through closer examination of DASD utilization.

Restrictions

- Every MONITOR ENABLE command yields a new mask. Thus, for example, if PERFORM and USER classes are currently being collected and you enter MONITOR ENABLE INSTSIM, then PERFORM and USER classes are stopped and INSTSIM is started.

- The DASTAP operand in the MONITOR ENABLE command must be specified prior to the MONITOR START TAPE command. DASTAP may be disabled at any time by respecifying the MONITOR ENABLE command with DASTAP absent from the class list.

- The SYSPROF class cannot be activated unless both the DASTAP and SCHEDULE classes are also active.

- If data collection is in progress when you issue a MONITOR ENABLE command and CP detects an error in the command line, no change is made to the monitoring status. Unrecognizable keywords and conflicting operands generate appropriate error messages. If you do not specify operands, an error message results.

```
               ┌     ┐
INTERVAL nnnnn |SEC| mm
               |MIN|
               └     ┘
```

Use this operand if a class that involves collecting data at periodic intervals is enabled or will be enabled, and if you want to override the default intervals of 60 seconds and 2 seconds. The 60 second interval can be considered as the primary sampling interval. It determines the frequency with which the PERFORM, USER and DASTAP classes of monitor data are collected. The two second interval applies to the high-frequency data sampling routine. This routine collects utilization statistics from the I/O subsystem (channels, control units, and devices) when the DASTAP class is enabled. The value specified by nnnnn is the interval between data collections. If you give no interval on the MONITOR INTERVAL command, an error message occurs. Either SEC or MIN may follow nnnnn to specify whether nnnnn represents seconds or minutes. If you give an interval but enter neither SEC nor MIN, the default is SEC. The maximum allowable interval is nine hours (540 minutes or 32,400 seconds). The minimum is five seconds. If no interval is specified in nnnnn, an error message occurs. The mm field is used to change the collection interval for the high-frequency sampler. This operand applies to the DASTAP class of data collection only. The default value for mm is two seconds. If specified, the interval may range from 1 to 99 seconds. It must be less than the specified value of nnnnn.

If the MONITOR INTERVAL command is not issued, the default interval is 60 seconds for the primary sampling interval, and 2 seconds for the high-frequency sampling interval. The MONITOR INTERVAL command can be issued at any time; however, if data collection is already in progress, the new interval does not take effect until the current interval has elapsed. The MONITOR interval is reset to the defaults when:

- You issue MONITOR STOP, or the monitor stops automatically.
- The system stops the monitor because of an unrecoverable I/O error.
- The end-of-tape is reached, or the spool record limit is reached.

```
       ┌                                           ┐
START  |SPOOL[1]  [to userid]  [BUFFS n]           |
       |CPTRACE                                    |
       |          ┌              ┐                 |
       |TAPE   raddr |MODE (  800)| [BUFFS n]      |
       |          |      { 1600 }|                 |
       |          |      ( 6250)|                  |
       └          └              ┘                 ┘
```

If SPOOL is specified, VM/370 Monitor data collection is begun to the spool file or tape, or the CP internal trace table is started. If no optional parameter is provided, SPOOL is the default.

When data collection is stopped and the spool is closed, the spool file is added to the chain of reader files destined for the virtual reader of the virtual machine defined by "to userid." The userid may be specified as an asterisk (*) if the recipient virtual machine is to be the one from which the START command is issued. If the "to userid" option is omitted, the userid specified in the SYSMON macro instruction is used. (See *IBM Virtual Machine Facility/370 Planning and System Generation Guide*.) The "to userid" option overrides the SYSMON specification and stays in effect until the system is re-IPLed or a new command that uses the "to userid" option is entered.

The monitor spool file is closed by a MONITOR STOP or MONITOR CLOSE command, or when the record count limit is reached (as specified in the SYSMON macro), or when a system restart or system shutdown occurs.

The filename and filetype of the monitor spool file are generated internally with the filetype, identifying date and time of starting. The class of spool file is specified in the SYSMON macro and defaults to "M." If no classes of data collection have been specified with an ENABLE command, those specified with the SYSMON macro are used.

The number of monitor buffers used are as specified in the SYSMON macro, or as requested with the BUFFS option of the START command. The BUFFS option overrides the SYSMON

----

[1] The default value is the active trace facility that is SPOOL or TAPE.

specification for the duration of the data collection session. Future monitoring sessions will also return to the SYSMON specification unless again overridden. If the number of buffers specified in the SYSMON macro has been defaulted, then the defaults, as described in the MONITOR START TAPE command, are adopted.

If CPTRACE is specified, the tracing of events that occur on the real machine is begun. The events are recorded on the CP internal trace table in chronological order. When the end of the table is reached, recording continues at the beginning of the table, overlaying data previously recorded.

If TAPE is specified, data collection by MONITOR onto a tape mounted on a 9-track tape drive is begun. Specify the real hexadecimal address of the tape drive that you want to use in raddr. This activates data collection for those classes of MONITOR CALL previously specified in a MONITOR ENABLE command and begins data collection. The mask that was saved by the MONITOR ENABLE command is moved into control register 8. The data is collected into buffer pages in real storage. These pages are separate from the internal trace table pages. The BUFFS n operand, where n is a value from 1 to 10, specifies the number of 4096-byte buffers to be used for monitoring operations. If the operand is omitted, a default value is supplied depending upon the processor model number.

Model	Buffers
135	2
138	2
145	2
148	3
155	3
158	3
165	4
168	4
3031	4
3032	4
3033	4

The operator may specify BUFFS 1 if the command entry is limited to ENABLE PERFORM for data collection. Once monitoring is in progress with one buffer specified, it is not possible to invoke MONITOR with additional ENABLE operands because of the buffer limitation.

Note: Single buffer operation is useful for basic performance analysis in minimum main storage configurations.

As each data page is filled, it is written on the tape. CP issues a rewind command followed by a SET MODE command for the reset value of tape density when MONITOR is started. You can request a different mode setting by specifying the MODE option in the MONITOR START TAPE command. Mode values of 800, 1600, or 6250 may be specified. If you specify a density mode that the tape drive cannot handle, the control unit does not return an error condition. In this case, the mode setting is ignored and the default control unit setting is used.

```
     ┌        ┐
STOP │SPOOL   │
     │CPTRACE │
     │TAPE    │
     └        ┘
```

If SPOOL is specified, the VM/370 Monitor data collection to a spool file or tape is stopped or the CP internal trace table is stopped. If no option is specified and the VM/370 Monitor is active, then data collection will be terminated whether a spool file or tape is in use. Internal tracing can only be stopped by specific use of the CPTRACE option. If automatic VM/370 Monitor data collection is active when the MONITOR STCP SPOOL command is entered, monitoring ceases and will not start again (even if the current time is within the bounds of the TIME operand of the SYSMON macro) unless the system fails or is shut down and reloaded.

If CPTRACE is specified it terminates the internal trace table event tracing. Event recording ceases but the pages of storage containing the CP internal trace table are nct released. Tracing can be restarted at any time by issuing the MONITOR START CPTRACE command.

If TAPE is specified it stops data collection by MONITOR on the tape. A zero mask is immediately stored in contrcl register 8, thus disabling MONITOR CALL interruptions. The last partially filled page is written out, two tape marks are written, and the tape is rewound and unloaded. The two buffer pages, which were obtained at the time the MCNITOR START TAPE command was processed, are released.

Note: The CPTRACE and TAPE or SPOOL operands of the MONITOR command are completely separate functions. Commands affecting the status of one function have no effect on the other.

CLOSE

closes the current spool file, making it available to the reader of the recipient virtual machine and continues monitoring with a new spool file. This option may be used when the VM/370 Monitor is spooling the collected performance data and you want to reduce the data collected thus far.

```
AUTODISK ⎧ON ⎫
         ⎩OFF⎭
```

overrides the specification in the SYSMON macro for automatic monitoring. This only affects the automatic startup of monitoring. If automatic monitoring is already active, only a MONITOR STOP command will stop it manually.

Note: In general, any attempts to override the definitions of the SYSMON macro instruction with commands are temporary. No Monitor checkpointing is attempted so that an IPL or crash causes full restoration of the initial automatic monitoring definitions.

```
TIME (FROM  h1:m1 TO h2:m2)
     )FOR    hh:mm        (
     )ALL                 (
     (NONE                )
```

temporarily overrides the automatic monitoring start and stop times defined by the SYSMON macro instruction. FROM h1:m1 TO h2:m2 specifies a start and stop time in hours and minutes using a 24-hour clock. (The default start time is 09:00 and end time is 17:00). FOR specifies the length of time data should be collected; it is intended to facilitate performance data collection over short periods of time for testing or benchmarking. Automatic monitoring must be in effect before this parameter is specified (AUTO=YES in the SYSMON macro of DMKSYS or MONITOR AUTO ON command). ALL specifies 24-hour monitoring and NONE specifies that monitoring can only take place under the direction of operator or system analyst.

Note: If automatic monitoring is imminent and the FOR option is specified, the period of monitoring defined by the SYSMCN macro is overridden.

```
        r      ┐
LIMIT n |NOSTOP|
        |STOP  |
        |SAMPLE|
        L      ┘
```

temporarily overrides the LIMIT options of the SYSMON macro. n is a decimal number from 10 to 50000 that represents the maximum buffer count in each spool file. If it is necessary to change the automatic monitoring specification after the limit is reached and the spool file is closed, the STOP or NOSTOP option may be specified. If you wish to change the automatic monitoring specification without changing the limit number, specify an asterisk (*) instead of a decimal number for n.

If SAMPLE is specified, n indicates the number of samples of PERFORM, USER, or DASTAP data (or any combination of the three) that are to be collected before the spool file is closed. After n samples of data have been collected, the monitor spool file is closed and moved to the virtual reader of the data reduction virtual machine. Monitoring continues uninterrupted using a new spool file. Automatic monitoring is specified by using the SYSMON macro instruction in DMKSYS at system generation.

```
SEEKS (INCLUDE  raddr raddr...)
      )EXCLUDE  raddr raddr...(
      )DELETE                 (
      (DISPLAY                )
```

This command option is used to control selective seeks monitoring. This enables the user to obtain seeks trace data on only those non-2305 DASD devices that are specifically requested. The command allows a list of devices to be established for data collection and permits display or deletion of an existing list. This decreases both the overhead involved in collecting the data and the processor time to reduce the data.

The INCLUDE option followed by a list of non 2305 DASD device addresses, specifies the devices for which seeks data is collected. The appropriate MONITOR ENABLE and START commands must be issued to invoke this function.

The EXCLUDE option specifies the devices for which seeks data should not be collected.

The DISPLAY option enables the contents of the list to be reviewed.

The DELETE option cancels the list, thus freeing the occupied real storage.

Note: A device list established with the INCLUDE or EXCLUDE option is retained across separate monitoring sessions. A list is not cleared when a MONITOR STOP command is entered; however, the DISPLAY and DELETE options are provided for list revision when SEEKS analysis is completed.

Though as many device addresses as can fit on a single input line may be entered, for performance reasons the list should be kept as short as possible.

Usage Notes:

The MONITOR command's response to unusual conditions are detailed in the following paragraphs.

During Automatic Monitoring Using Spool Files

SHUTDOWN: Data collection is terminated with a MONITOR STOP SPOCL command. The spool file is closed and moved for data reduction to the reader chain of the designated virtual machine.

SYSTEM FAILURE: The current monitor data collection buffer is abandoned but the spool file containing all previously output buffers is closed and made available to the virtual reader of the designated virtual machine for further data reduction.

INITIAL PROGRAM LOAD: If automatic monitoring has been specified with the SYSMON macro of DMKSYS and the IPL occurs within the time range specified on that same macro, monitor data collection is automatically started. The operator is informed but no action is required. He may, however, use the MONITOR DISPLAY SPOOL command to check the specifications for automatic monitoring.

SPOOL SPACE: If automatic monitor data collection is regularly active and system messages are issued indicating serious loss of spool space, several checks should be made by the operator:

- Ensure that adequate arrangements have been made to read and reduce the monitor spool files on a day-to-day basis so that the occupied spool space can be returned to the system.

- Ensure that the data collection classes do not include trace classes, which produce large volumes of data (e.g. INTSIM or SEEKs). Do this with the MONITOR DISPLAY command remembering that the specifications of the DMKSYS macro SYSMON may be temporarily overridden by the operator or systems analyst using the MONITOR command.

- If spool space is a scarce or critical resource, monitoring can either be done to tape or the LIMIT parameter used. The LIMIT parameter of the SYSMON macro or the LIMIT option of the MONITOR START SPOOL command can be used to restrict the maximum size of the monitor spool file by terminating monitoring when the maximum record count is reached.

SUSPENSION: The system analyst may determine, after studying the data reduction reports, that data collection suspension is occurring because of insufficient buffer space. The operator may be requested to increase the buffer allocations using the BUFFS n option of the MONITOR START command until the SYSMON macro in DMKSYS can be revised.

During Manual Monitoring Using Tape

SUSPENSION: Data reduction may determine that data collection suspension is occurring because of insufficient buffer space. This usually occurs when classes USER and/or DASTAP are enabled, while large numbers of users are logged on, or the system I/O configuration is extensive. Loss of data may be avoided by specifying additional buffers (with the BUFFS option of the MONITOR START TAPE command) until no more suspensions are reported by data reduction.

UNRECOVERABLE TAPE ERROR: Unrecoverable tape errors are errors that the I/O error handling routines cannot recover from by standard recovery procedures. When an unrecoverable error occurs, MONITCR tries to write two tape marks and rewind/unload the tape. The use of the tape is abandoned and data collection stops. The operator is informed or action taken. Whether or not the write-tape-marks and rewind/unload is successful, the tape drive is released.

END-OF-TAPE CONDITION: When end-of-tape occurs, a tape mark is written on the tape and it is rewound and unloaded. MONITOR is stopped and the operator is informed of the action taken.

INITIAL PROGRAM LOAD: MONITOR START CPTRACE is active after real system IPL (manual or automatic). The MONITOR tape data collection is off after IPL.

SHUTDOWN: System shutdown implies a MONITOR STOP TAPE command. The system performs normal command processing for the STOP TAPE function.

SYSTEM FAILURE: If the VM/370 system fails and data collection is active, an attempt is made to write two tape marks and rewind/unload the tape. If the tape drive fails to rewind and unload, be sure to write a tape mark before rewinding and unloading the tape. MONITOR data collection is terminated by the system failure.

I/O DEVICES: A supported tape drive must be dedicated to the system for the duration of the monitoring.

Responses

The following response occurs if you issue the MONITOR DISPLAY command and data collection is on tape. If spool files are used, then the automatic monitoring specifications and status are listed as described in the MONITOR DISPLAY section.

```
CLS   KEYWORD   STATUS
nn    keyword   ENABLED
nn    keyword   DISABLED
 .       .         .
 .       .         .
 .       .         .
--    CPTRACE   ENABLED
```

The following response occurs for MONITOR commands, other than MONITOR DISPLAY, that successfully execute:

```
          COMMAND COMPLETE
```

MSGNOH

Privilege Class: B

Use of the MSGNOH command provides the ability for a system service virtual machine to send messages to specified users without the standard header associated with the MESSAGE command. The format of the class B MSGNOH command is:

```
┌──────────────────────────────────────────────────────┐
│  MSGNOH   │     {userid}    msgtext                   │
└──────────────────────────────────────────────────────┘
```

where:

userid   is the identification of the virtual machine to receive the message.

msgtext   specifies the text of the message that is to be transmitted. Length of the message may be as long as the remainder of the input line.

Responses

msgtext   only the message text is received by the specified virtual machine.

NETWORK

| Privilege Class: A or B

The CP NETWORK command loads, dumps, and controls the operation of a 370x control program in the VM/370 environment. NETWORK is also used to control remote 3270 devices attached to VM/370 via binary synchronous lines and 270X or 370x (operating in 270X emulation mode (EP)) control units.

NETWORK COMMAND USAGE FOR 370X FUNCTIONS

The NETWORK command:

- Causes 370x dump operations
- Initiates 370x load operations
- Enables or disables terminal resources
- Varies resources online or offline
- | Halts a particular resource
- Ceases all 370x operations
- Queries and displays 370x resource status and storage
- Traces line activity to and from a 370x resource

NETWORK COMMAND USAGE FOR REMOTE 3270

Use the NETWORK command to control remote 3270 resources and to vary the polling delay on the binary synchronous lines that are connected to these resources. The NETWORK command may be used whether the remote 3270 resources are connected to a 2701, 2703, or to a 370x in Emulation | Program (EP) mode with lines in EP mode.

The operands that specifically affect remote 3270s are:

- SHUTDOWN (Class A)
- POLLDLAY (Class A,B)
- ENABLE (Class A,B)
- VARY (Class A,B)
- DISABLE (Class A,B)
- QUERY (Class A,B)

RESOURCE IDENTIFICATION FOR REMOTE 3270'S DEVICE AND CONTROL UNITS

Resources are defined as display stations, printers, and control units in the 3270 remote system. Whenever a remote 3270 resource is referred to in the NETWORK command, and is identified in a VM/370 system message, the format is a four-character hexadecimal number, where the three low-order characters are the actual station resource identifier, and the high-order hexadecimal character is a relative line code associated with a particular physical Binary Synchronous Communication (BSC) line that remote 3270 terminals use. When the 3270 resource is referred to in the SET PFnn COPY command, the 3-character hexadecimal number is the remote resource ID. The 3270 remote resource ID is a 12-bit binary value that can have a low value of zero, to a high value determined by the number of CLUSTER and TERMINAL macro instructions defined for a communication line (RDEVICE) during system generation. For example, assume that there are two BSC lines for remote 3270 display systems in a particular configuration. The first line to appear in the DMKRIO module would be assigned a code of 0, the second line would have a code of 1. Any other binary synchronous lines used for other purposes are ignored. Resource ID 12 (decimal) on communication line 1 would be represented as "000C"; resource ID 31 on communication line 2 would be represented as '101F', etc.

HOW TO USE THE NETWORK COMMAND

When using the NETWORK command to control the operation of the 370x Emulation Program or remote 3270s on a 2701, 2703, or 370x in Emulator Program (EP) mode with lines in EP mode, the operator must be aware of the different classes of resources that are defined at generation time for the 370x control program and the 3270 remote support.

When operating with a 270x or an Emulation Program, there is only one address for each logon device, and that is the physical subchannel address for the telecommunications line. For a simple leased line configuration, there is one resource ID for each line, and one resource ID for each terminal (one terminal per line), alternating in numeric value.

The majority of the NETWORK command operations are performed for terminal resources. For example, NETWORK ENABLE, DISABLE, QUERY, HALT, VARY ONLINE, and VARY OFFLINE all operate for terminals. The NETWORK QUERY command line can be used to display the status of a line resource, but only when the "NETWORK QUERY resource" command format is used. The possible states of a line resource are:

- OFFLINE (that is, inactive)
- ACTIVE
- EP-MODE

While the NETWORK VARY ONLINE and VARY OFFLINE command lines may be used for a line resource, they are primarily intended for use with terminal resources, because the state of the line changes automatically if the terminal is enabled or disabled. Also, NETWORK VARY EP is valid only for line resources and, in this case, the terminal resources change state when the line changes state.

The only way to tell which resources are lines and which are terminals is to examine the output from the first stage of the 370x control program generation. The installation system programmer (or whoever performs the 370x control program generation), should prepare a cross-reference list of resource IDs and their characteristics (such as line or terminal, type of line, location, and so on) for the operations personnel. Use the following commands:

- For terminal only

```
NETWORK ENABLE
NETWORK DISABLE
NETWORK QUERY ACTIVE
NETWORK QUERY FREE
NETWORK QUERY OFFLINE
NETWORK QUERY ALL
```

- For lines only

```
NETWORK VARY EP
NETWORK TRACE resource
```

- For either lines or terminals

```
NETWORK QUERY resource
NETWORK HALT
NETWORK VARY ONLINE
NETWORK VARY OFFLINE
```

The format of the class A NETWORK command is:

NETWORK	HALT resource

where:

HALT resource
attempts to terminate any active channel program on the specified resource (line or terminal). The term resource is a 4-digit hexadecimal identity of a 370x resource. The last three digits are the actual NCP resource ID. The first digit is a device sequence number associated with a particular 370x. This device sequence number designates the relative position of the device in the DMKRIO module: the first 370x listed has a device sequence number 0, the second listed has a device sequence number 1, and so on.

Responses

NETWORK HALT

The normal response is:

```
DEVICE HALTED
```

This response indicates that VM/370 has attempted to reset status and halt the device.

This blank page represents page 109 which
was deleted by Release 6 PLC 17

The format of the class A and B NETWORK command is:

```
┌─────────────────────────────────────────────────────────────────────────────────┐
| NETWORK | LOAD raddr ncpname                                                    |
|         |            ┌     ┐                                                    |
|         | DUMP raddr |IMMED|                                                    |
|         |            |OFF  |                                                    |
|         |            |AUTO |                                                    |
|         |            └     ┘                                                    |
|         |         ┌                        ┐                                    |
|         | ENable  |ALL                     |                                    |
|         |         |resource [resource... ]|                                     |
|         |         └                        ┘                                    |
|         |         ┌                        ┐                                    |
|         | DISAble |ALL                     |                                    |
|         |         |resource [resource... ]|                                     |
|         |         └                        ┘                                    |
|         |         ┌                        ┐                                    |
|         | Query   |ACTive                  |                                    |
|         |         |OFFline                 |                                    |
|         |         |FREe                    |                                    |
|         |         |ALL                     |                                    |
|         |         |resource [resource... ]|                                     |
|         |         └                        ┘                                    |
|         |                            ┌    ┌         ┐   ┐                       |
|         | Display raddr hexloc1  |{-}|hexloc2|   |                              |
|         |                        |{:}|END    |   |                              |
|         |                        |    └         ┘   |                           |
|         |                        |    ┌           ┐ |                           |
|         |                        |{.} |bytecount| |                             |
|         |                        |    |END      | |                             |
|         |                        └    └           ┘ ┘                           |
|         | VARY {ONline }  resource [resource... ]                               |
|         |      {OFFline}                                                        |
|         |                                                                       |
|         |                                                                       |
|         |                ┌     ┐                                                |
|         | POLLdlay nnnn  |ALL  |                                                |
|         |                |raddr|                                                |
|         |                └     ┘                                                |
└─────────────────────────────────────────────────────────────────────────────────┘
```

where:

LOAD raddr ncpname
loads an EP 370x control program.

raddr — is the real address of the 370x to be loaded.

ncpname — is the name, previously defined by a NAMENCP macro and saved on a CP volume, of the 370x control program image to be loaded into the 370x specified by raddr.

```
           ┌     ┐
DUMP raddr |IMMED|
           |OFF  |
           |AUTO |
           └     ┘
```

dumps the contents of 370x storage for EP 370x control programs.

raddr — is the real address of the 370x to be dumped.

IMMED specifies that the 370x is to be dumped immediately. See the "NETWORK Dump Operations" section for additional information.

OFF specifies that the 370x is not to be dumped automatically if the 370x control program abnormally terminates.

AUTO specifies the automatic dumping and reloading of the 370x if the 370x control program abnormally terminates.

```
       ┌                      ┐
ENABLE |ALL                   |
       |resource [resource...]|
       └                      ┘
```

activates 370x resources (terminals only) and remote 3270 resources for use by VM/370. The resource operand may be enabled selectively by specifying the four-digit hexadecimal identity of the terminal resource to be enabled. The last three digits are the actual resource identification. The first digit is a device sequence number associated with a particular 370x. This device sequence number designates the relative position of the device in the DMKRIO module: the first 370x listed has a device sequence number 0, the second listed has a device sequence number 1, and so on. The resource specified must be a terminal device. The NETWORK ENABLE command first ensures that the associated line resource is activated, and then enables the terminal device or, in the case of remote 3270s, formats the screen. The response from enabling terminal devices or formatting the screen is the appearance of the "vm/370 online" message on the terminal.

Note: If the device being selectively enabled is a 3275 remote display unit and the unit is equipped with a printer to handle the display's hardcopy output (via PFnn COPY), then the printer must also be enabled.

```
        ┌                      ┐
DISABLE |ALL                   |
        |resource [resource...]|
        └                      ┘
```

disables 370x resources (terminals only) and remote 3270 resources. ALL disables all 370x terminals. The resource operand used to disable selective resources, specify the four-digit hexadecimal identity of the terminal resources to be disabled. The last three digits are the actual resource ID. The first digit is a device sequence number associated with a particular 370x. This device sequence number designates the relative position of the device in the DMKRIO module: the first 370x listed has a device sequence number 0, the second listed has a device sequence number 1, and so on.

If any of the resources specified on the NETWORK DISABLE command are in use at the time the command is issued, they are not immediately disabled. However, as soon as the resource becomes free (usually after a LOGOFF command is issued), the resource is automatically disabled.

```
      ┌                       ┐
QUERY │ACTIVE                 │
      │OFFLINE                │
      │FREE                   │
      │ALL                    │
      │resource [resource...] │
      └                       ┘
```

displays the status of 370x resources (lines or terminals) and remote 3270 resources.

ACTIVE displays only the resources (terminals, remote 3270 display and printer stations) that are active (those being used by VM/370 users).

OFFLINE displays only resources (terminals, remote 3270 display and printer stations) that are not available to VM/370 users.

FREE displays only resources (terminals, remote 3270 display and printer stations) that are not offline and also not currently in use.

ALL displays the active, offline or free status of each resource (terminals or remote 3270 display and printer stations only) attached to all 3704 and 3270 remote control units on the VM/370 system.

With the exception of remote 3270, "resource" displays only the resources (lines or terminals) whose four-digit hexadecimal identity is specified. The last three digits are the actual resource ID. The first digit is a device sequence number associated with a particular 370x. This device sequence number designates the relative position of the device in the DMKRIO module: the first 3704 or 3705 listed has device sequence number 0, the second listed has device sequence number 1, and so on.

```
                       ┌    ┌         ┐  ┐
DISPLAY raddr hexloc1  │{-} │hexloc2  │  │
                       │{:} │END      │  │
                       │    └         ┘  │
                       │    ┌          ┐ │
                       │{.} │bytecount │ │
                       │    │END       │ │
                       └    └          ┘ ┘
```

this operand reserved for 370x operations only, displays the contents of 370x storage. The data is displayed in fullwords. No EBCDIC translation is provided.

The raddr operand is the real address of the 370x whose storage is to be displayed.

The hexloc1 operand specifies the hexadecimal address of the start of the display and must be specified. To display more than one fullword, : or - or . must be specified.

The hexloc2 operand specifies the hexadecimal location of the end of the display.

bytecount
specifies the number of bytes to be displayed. The sum of hexloc1 and bytecount must be an address that does not exceed the 370x storage size. Bytecount must have a value of at least one. A minimum of four bytes is displayed.

END

indicates that the display continues until the end of storage is reached and is the default if hexloc2 or bytecount is not specified.

```
VARY {ONLINE } resource [resource...]
     {OFFLINE}
```

varies the status of specified 370x resources.

ONLINE places a resource (line or terminal) online.

OFFLINE places a resource (line or terminal) offline.

Note: only the ONLINE and OFFLINE operands are valid for remote 3270 resources.

With the exception of remote 3270, resource is a four-digit hexadecimal identity. The last three digits are the actual resource ID. The first digit is a device sequence number associated with a particular 370x. This device sequence number designates the relative position of the device in the DMKRIO module: the first 370x listed has a device sequence number 0, the second listed has a device sequence number 1, and so on.

```
              ┌     ┐
POLLDLAY nnnn │ALL  │
              │raddr│
              └     ┘
```

changes the duration of the polling delay interval for the binary synchronous line to the value of nnnn. The address of the binary synchronous line is raddr and nnnn is the decimal number in tenths of a second (not to exceed 9999) for the polling delay interval. If ALL is specified, the polling delay interval is set for all the 3270 remote lines.

The polling delay interval that is defined at system generation is two seconds.

Note: The polling delay interval is that period of time from the time a binary synchronous line receives a negative response from a general polling sequence until the polling delay interval expires, or a message is sent to the station on the binary synchronous line.

The polling delay interval minimizes unproductive polling and processor meter time. In general, if no data or other communications is being received from the stations on the binary synchronous line, the polling delay interval is started and control is given to the dispatcher.

Responses

NETWORK LOAD

```
CTLR raddr ncpname LOAD COMPLETE
```

The 370x "raddr" was successfully loaded with the control program "ncpname".

NETWORK DUMP

```
CTLR raddr DUMP COMPLETE
```

The 370x "raddr" was successfully dumped.

NETWORK ENABLE, NETWORK DISABLE, NETWORK VARY

The normal response is:

```
COMMAND COMPLETE
```

NETWORK HALT

The normal response is:

```
DEVICE HALTED
```

NETWORK QUERY

```
  DEV rid LOGON AS userid
  DEV rid DISABLE
  DEV rid ENABLED
  DEV rid OFFLINE

  LINE rid ACTIVE
| LINE rid OFFLINE

  DEV rid1 ENABLED, DEV rid2 ENABLED, DEV rid3 ENABLED,...
  DEV rid1 DISABLE, DEV rid2 DISABLE, DEV rid3 DISABLE,...
  DEV rid1 OFFLINE, DEV rid2 OFFLINE, DEV rid3 OFFLINE,...
```

Note: If either of the following responses occurs, a problem exists with the Network Control Program and/or the 370x.

$\left\{\begin{array}{l}\text{ACTIVE}\\\text{FREE}\\\text{OFFLINE}\end{array}\right\}$ rid NOT FOUND

rid NOT FOUND

where:

LOGON indicates that the resource is in use as a virtual machine operator console, by "userid".

DISABLE indicates that the resource is online but is not available for access to VM/370.

ENABLED indicates that the resource is available for user access to VM/370.

ACTIVE indicates that the line resource is online and has been activated. Terminals on the line may or may not be in use.

FREE indicates that the resource is not offline and also not currently in use.

EP-MODE indicates that the line resource is a PEP line currently in emulation mode at real address "raddr".

OFFLINE indicates that the resource is inactive and unavailable for use.

rid is the real resource identifier.

userid is the user identifier.

raddr is the real device address.

This blank page represents page 116 which
was deleted by Release 6 PLC 17

ORDER

Privilege Class: D

Use the ORDER command to place closed spool files (of a specified device type) in a different order. You may determine via the QUERY command the filename, filetype, originating userid, and other attributes of all of your files. The format of the ORDER command is:

```
┌────────────────────────────────────────────────────────────────────────┐
| ORDer      | ┌        ┐ ┌Reader  ┐ ┌CLass c1 CLass c2...┐ ¹            |
|            | |userid  | <Printer > <spoolid1 spoolid2    > ¹            |
|            | |SYSTEM  | └PUnch   ┘ └                    ┘              |
|            | └        ┘                                                 |
|────────────────────────────────────────────────────────────────────────|
|¹Sequencing may be done with the ORDER command using a combination of   |
| "CLASS c" specifications and spoolid specifications. For example       |
|                                                                        |
|        ORDER PRINTER CLASS A 1963 CLASS C                              |
|                                                                        |
| specifies that printer files will be processed in the following        |
| order: all Class A files, the file with the spcolid 1963, all Class    |
| C files.                                                               |
└────────────────────────────────────────────────────────────────────────┘
```

where:

userid — allows the spooling operator to manipulate the spool files for a particular user.

SYSTEM — allows the spooling operator to manipulate all spool files in the system, regardless of userid.

READER
RDR — orders the reader files.

PRINTER
PRT — orders the printer files.

PUNCH
PCH — orders the punch files.

CLASS c — orders the files by their class. "c" specifies class designations for input and output spool files. This specification allows files to be placed in the order of classes specified by CLASS c1, CLASS c2, and so forth. c is a one-character alphameric field with values that extend from A to Z and from 0 to 9.

spoolid — processes the files in the order in which the spoolids are given.

Responses

```
{nnnn} FILES ORDERED
{NO  }
```

is the response to the user who issued the ORDER command.

PURGE

Privilege Class: D

Use the PURGE command to remove closed spool files from the system before they are printed or punched by the spooling devices, or before they are read by a user. The spooling operator can purge any file in the system. Any file can be purged, regardless of its status, as long as it has not been selected for processing. The format of the PURGE command is:

```
┌────────────────────────────────────────────────────────────────────────┐
│ PURge    │ ┌      ┐ ⎧Reader      ┌                             ┐⎫      │
│          │ │userid│ ⎪Printer     │ ALL                         │⎪      │
│          │ │SYSTEM│ ⎨PUnch       │ Class c1 Class c2 ...¹      │⎬      │
│          │ └      ┘ ⎪            │ spoolid1 spoolid2 ...       │⎪      │
│          │          ⎩ALL         └                             ┘⎭      │
├────────────────────────────────────────────────────────────────────────┤
│¹Purging may be done using a combination of CLASS c and spoolid         │
│ specifications. For example:                                           │
│                                                                        │
│    PURGE PRINTER CLASS A 1932 CLASS D 619                              │
│                                                                        │
│ specifies that all class A and class D files and files with spoolid    │
│ 1932 and 619 are to be purged.                                         │
└────────────────────────────────────────────────────────────────────────┘
```

where:

userid is the user identification for the user whose spool files are to be purged.

SYSTEM allows the spooling operator to purge all files on the system, regardless of userid.

READER purges the reader file. Optionally, the reader files may be purged by class or spoolid.

PRINTER purges the printer files. Optionally, the printer files may be purged by class or spoolid.

PUNCH purges the punch files. Optionally, the punch files may be purged by class or spoolid.

ALL purges all the files; all operands that follow it, if specified, are ignored. If ALL follows READER, PRINTER, or PUNCH, only the files for that device type are purged.

CLASS c purges the files of the specified class, for the specified device type.

spoolid purges the files of the specified spool identification, for the specified device type.

Responses

```
{nnnn} FILES PURGED
{NO  }
```

indicates the number of files purged.

Note: This response is suppressed if the virtual machine user has issued the SET IMSG OFF command.

The system operator, monitoring the RSCS virtual machine for his installation, should not specify this option because information messages are required for accurate monitoring of RSCS system activity.

QUERY

Privilege Classes: A, B, C, D, E, or F

Use the QUERY command to request system status and machine configuration information. (For 3704 or 3705 Communication Controllers and remote 3270 resources see the Class A and B NETWORK command.) Not all operands are available in every privilege class.

Operands available to the specified privilege classes are given below. The format of the Class A and E QUERY command is:

```
┌──────────────────────────────────────────────────────────────────┐
│              │ ⎧AFFinity [userid]    ⎫                            │
│              │ ⎪CPAssist¹            ⎪                            │
│  Query       │ ⎨JOurnal²             ⎬                            │
│              │ ⎪PAGing               ⎪                            │
│              │ ⎪PRIORity userid      ⎪                            │
│              │ ⎩SASsist¹             ⎭                            │
├──────────────────────────────────────────────────────────────────┤
│¹The collective use of both QUERY CPASSIST and QUERY SASSIST is used │
│  to determine the current status of the expanded virtual machine    │
│  assist portion  of the VM/370 Extended Control-Program Support     │
│  (VM/370: ECPS).                                                    │
│²The JOURNAL operand is valid only if STQUERY=YES is specified in    │
│  the SYSJRL macro instruction in DMKSYS.                            │
└──────────────────────────────────────────────────────────────────┘
```

where:

AFFINITY [userid]
(attached processor applications only) displays the affinity setting of a specific virtual machine or all virtual machines that are logged onto the system. The affinity setting is accomplished in the VM/370 directory OPTION statement or via the CP SET command. The affinity setting of a virtual machine indicates the selected processor (main or attached) that is used to execute the programs of a virtual machine. If there is no affinity setting for a virtual machine, then the virtual machine is dispatched for execution on the first available processor.

CPASSIST
displays the current status of the CP assist for the VM/370 system.

JOURNAL displays the status of the logon/link journaling facility.

PAGING displays the current system paging activity.

PRIORITY userid
displays the current priority of the specified userid. This is established in the VM/370 directory but can be overridden by the SET PRIORITY nn command.

SASSIST displays the current status of the virtual machine assist for the VM/370 system. QUERY SASSIST also indicates the current status of virtual interval timer assist portion of VM/370: ECPS.

Responses to the Class A and E Query Commands

QUERY CPASSIST

```
CPASSIST {ON }
         {OFF}
```

where:

ON indicates that the CP Assist portion of the VM/370 Extended Control-Program Support is active.

OFF indicates that the CP Assist portion of the VM/370 Extended Control-Program Support is inactive or not available on the System/370 Processor.

Note: To determine the active status of the Expanded Virtual Machine Assist portion of ECPS requires a positive response to QUERY CPASSIST and QUERY SASSIST.

QUERY JOURNAL

```
               {ON }       {ON }
JOURNAL  LOGON {OFF}  LINK {OFF}
```

where:

ON indicates the relevant journaling function is active.

OFF indicates the relevant journaling function is inactive.

QUERY PAGING

```
PAGING nn, SET mm, RATE nnn/SEC INTERVAL=hh:mm:ss

     --or for attached processor application--

PAGING MP-nn/AP-nn, SETmm, RATE nnn/SEC INTERVAL=hh:mm:ss
```

where:

nn specifies the percentage of time the processor was in page wait during this time interval.

mm is the system paging activity index (threshold value). This value affects the paging rate and degree of multiprogramming that VM/370 tries to attain. The value mm is normally 16.

MP-nn specifies the percentage of time the main processor was in page wait during this time interval.

AP-nn specifies the percentage of time the attached processor was in page wait during this time interval.

nnn/SEC is the current CP paging rate in pages per second.

hh:mm:ss is the time interval between the issuance of QUERY PAGING commands.

QUERY PRIORITY userid

```
userid PRIORITY = nn
```

nn is the the assigned priority of the specified user. The lower the value, the higher the priority.

QUERY SASSIST

```
SASSIST {ON }
        {OFF}
```

ON or OFF indicates that the virtual machine assist (and the virtual interval timer assist portion of VM/370 ECPS - if the processor is so equipped) is active or inactive (or not available on the system).

QUERY SASSIST (attached processor applications)

```
SASSIST {ON } PROC nn, {ON } PROC nn
        {OFF}          {OFF}
```

ON or OFF indicates that virtual machine assist (and the virtual interval timer assist portion of VM/370 ECPS - if the main processor or the attached processor is so equipped) is active or inactive for the designated processor, PROC nn.

nn can be any hexadecimal value from 00 to 3F. The response is indicated in hexadecimal values. The values assigned to the main processor and the attached processor are assigned by the installation when the system hardware is installed.

Note: To determine if the expanded virtual machine assist portion of the VM/370 Extended Control-Program Support is active requires a positive response to both QUERY CPASSIST and QUERY SASSIST commands.

QUERY AFFINITY [userid]

```
     USER     PROCESSOR AFFINITY
     userid   affinity
        .        .
        .        .
        .        .
        .        .
```

where:

userid is the user identification of the virtual machine.

affinity is the address of the processor (main or attached) that will execute the virtual machine instructions. Affinity can be any hexadecimal value from 0 to 3F. The processor address is set when the System/370 processors are installed. In attached processor systems, the processor addresses are unique. If NONE is indicated in the response, no affinity is set and the virtual machine is dispatched for execution on the first available processor.

Note: In System/370 Model 168 installations, the role of attached processor and main processor can be reversed by a console switch setting.

The format of the Class B QUERY command is:

```
 ┌────────────────────────────────────────────────────────────────────┐
 │                                                                    │
|│  Query    │  /  /   DAsd     ┌        ┐                \   \       │
|│           │ |  |             |Sysvirt|  ┌        ┐     |   |      │
 │           │ |  |             |Virtual|  |ACTive  |     |   |      │
 │           │ |   \            └        ┘  |OFFline|      )  |      │
 │           │ |    >  TApes                |FREe   |     <   |      │
 │           │ |   /   LINES[1]             |ATTach |     |   |      │
 │           │ |  |    UR                   |ALL    |     |   |      │
 │           │ <  |    GRaf                 └        ┘     |    >     │
 │           │ |   \   ALL                                /    |      │
 │           │ |                                               |      │
 │           │ |   DAsd volid                                  |      │
 │           │ |   TDsk                                        |      │
 │           │ |   STORage                                     |      │
|│           │ |   raddr                                       |      │
 │           │ |   SYStem raddr                                |      │
 │           │  \  DUMP                                       /       │
 │────────────────────────────────────────────────────────────────────│
 │[1]Query  LINES  is not effective  for  3704/3705  resources unless the│
 │ 3704/3705 is operating in  270x Emulation Program (EP) mode.  For  │
 │ 3704/3705 Communications Controllers operating in Network Control  │
 │ Program (NCP) or Partitioned Emulator Program (PEP) mode, use the  │
 │ NETWORK QUERY command.                                             │
 └────────────────────────────────────────────────────────────────────┘
```

where:

DASD displays the real addresses of disk or drum devices.

SYSVIRT displays the addresses and specified status of 3330V devices that have the SYSVIRT feature.

VIRTUAL displays the addresses and specified status of 3330V devices that have the VIRTUAL feature.

TAPES displays the real addresses of magnetic tape units.

LINES displays the real addresses of communication lines.

UR displays the real addresses of unit record devices (card reader, card punches, printers).

GRAF displays the locally attached display devices.

ALL (used as a first operand) displays all devices and the size of real storage.

DASD volid
displays the active or free status of the specified DASD volume.

TDSK displays all the currently allocated temporary disk space (TDSK) from all available system-owned volumes assigned to virtual machine users.

STORAGE displays the size of real storage.

raddr displays forms name and image library name.

SYSTEM raddr
displays the userid, virtual address, and access mode of virtual disks which reside on the specified channel and control unit address raddr belonging to logged-on users.

DUMP displays at the operator's terminal the type of device and device address of the unit designated to receive abnormal termination dumps.

ACTIVE displays the status of only the active devices within the group specified. This is the default. Active devices do not include devices that are "free" or "offline". An active device is one that is in use by a user or the system.

OFFLINE displays only the devices in an "offline" status within the group specified. An offline device is one that is not available for access by any user or the system.

FREE displays all the devices that are not currently in use by the system or a user on the system. Free devices do not include "offline" devices. A free device is one that is not in use by a user or the system.

ATTACH displays all the devices that are dedicated to any user on the system. An attached device is also an active device.

ALL (as the second operand) displays the status of all devices within the group specified. The status is typed in the order of "active", "free", and "offline" and is equivalent to the response from entering

```
QUERY type ACTIVE
QUERY type FREE
QUERY type OFFLINE
```

Responses to the Class B QUERY Command

QUERY DASD

DASD raddr ATTACH TO userid vaddr

is displayed if the real device specified by raddr is attached to a user's (userid) virtual machine at virtual address vaddr.

DASD raddr CP SYSTEM volid nnn

is displayed if the real device designated by raddr is allocated to the system for use as user's minidisks. nnn is the number of active user's minidisks on the physical disk and volid is the volume serial number of the real disk.

DASD raddr CP OWNED volid nnn

is displayed if the real device designated by raddr is used by the system for paging and spooling activity. nnn is the number of active user's minidisks and T-disks (if any) on the physical disk and volid is the volume serial number of the real disk.

QUERY TAPES

```
TAPE raddr CP SYSTEM
```

is displayed if the real tape device designated by raddr is attached to CP for its exclusive use.

```
TAPE raddr ATTACH TO userid vaddr
```

is displayed if the real tape device designated by raddr is attached to a user's (userid) virtual machine at virtual address vaddr.

QUERY LINES

$$\left\{\begin{matrix}\text{LINE}\\\text{CONS}\end{matrix}\right\}\text{ raddr LOGON AS userid}$$

indicates that the user represented by userid is currently logged on at the terminal located at real address raddr.

```
LINE  raddr ATTACH TO userid vaddr
```

indicates that the communication line at raddr is attached to the virtual machine represented by userid at virtual address vaddr.

QUERY UR

$$\left\{\begin{matrix}\text{PRT}\\\text{PUN}\end{matrix}\right\}\text{ raddr }\left\{\begin{matrix}\text{STARTED}\\\text{DRAINED}\end{matrix}\right\}\text{ SYSTEM CLASS = a... }\left\{\begin{matrix}\text{SEP}\\\text{NOSEP}\end{matrix}\right\}$$

$$\text{RDR raddr }\left\{\begin{matrix}\text{STARTED}\\\text{DRAINED}\end{matrix}\right\}\text{ SYSTEM}$$

is displayed for each unit record device assigned to the system for spooling activity.

where:

raddr   is the real device address (cuu).

DRAINED   indicates that the device is not currently available for processing. A START command must be issued to activate the device.

STARTED   indicates that the device is available for spooling activity.

a...   specifies the classes serviced by the output device. Up to four classes may be serviced by an output device. No blanks or commas are allowed between classes.

NOSEP   indicates the device was started with the NOSEP option.

SEP   indicates the device was started without the NOSEP option.

Note: The separator (SEP) option applies to printer output where the edge of the fanfolded continuous forms are heavily printed. This indicates to the spooling operator the beginning and end of adjacent spool files.

```
(PRT )
< PUN > raddr ATTACH TO userid vaddr
( RDR )
```

is displayed if the device is attached to a user's virtual machine at vaddr.

If the unit record device is currently active with a spool file, the following additional response is also given:

```
(PRT )        (PRINTING )
(PUN ) raddr  (PUNCHING ) userid FILE = file RECDS = norecs COPY = nn a typ

 RDR   raddr   READING    userid FILE = file
```

where:

userid   is the name of the spool file owner.

file   is the spool file spoolid number.

norecs   is the total file logical record count.

nn   is the number of copies remaining for output, where 01 indicates the last copy.

a   is the spool file class.

typ   is the originating device type (PRT, PUN, CON).

QUERY GRAF

GRAF raddr LOGON AS userid

indicates that the user represented by userid is currently logged on at the terminal located at real address raddr.

GRAF raddr ATTACH TO userid vaddr

indicates that the display device at real address raddr is attached to the virtual machine represented by userid at virtual address vaddr.

QUERY ALL

Produces the same results as if the following commands were issued:

```
QUERY STORAGE
QUERY UR
QUERY LINES
QUERY DASD
QUERY TAPES
QUERY GRAF
```

QUERY type OFFLINE

This command produces a response for each offline device in the following format:

```
type raddr OFFLINE
```

Multiple responses are displayed in the following format:

```
        type raddr OFFLINE, ...
         .      .      .
         .      .      .
         .      .      .
```

Note: In the above responses the term type refers to one or more of the following device types:

Type	Meaning
DASD	Direct access device
TAPE	Magnetic tape units
LINE	Communication line
RDR	Card reader
PRT	Line printer
PUN	Card punch
GRAF	Graphics device
CONS	Console
CTCA	Channel to channel adapter
CTLR	3704/3705 communications controller
DEV	Any other device

QUERY type FREE

This command produces a response for each device that is not active or offline in the following format:

```
        type raddr FREE
```

For unit record devices the response is:

```
        type raddr DRAINED
```

Note: This response implies that no spool files are queued for this device.

For communication devices the response is:

```
                   (ENABLED )
        type raddr (DISABLED)
```

For DASD devices with mounted volumes the response is:

```
        type raddr (FREE )
                   (volid)
```

Multiple responses are displayed in the following format:

```
        type raddr FREE, ...
         .     .     .
         .     .     .
         .     .     .
```

QUERY DASD volid

The command response is given in either the "active" or "free" format depending upon the device status.

QUERY TDSK

This command displays all the currently allocated user TDSK space from all available system-owned volumes. One entry of the following format is produced for each TDSK:

```
userid vaddr nnn
```

where:

userid is the virtual machine identification.

vaddr is the user's virtual device address.

nnn is the number of cylinders allocated.

Note: If the operator queries any real device or group of devices (such as QUERY DASD) the following message occurs for all devices in a not-ready status and the CPU alarm rings:

```
type raddr INT REQ
```

QUERY STORAGE

STORAGE = xxxxxK

displays the size of real storage (xxxxx) in multiples of 1024 bytes.

QUERY raddr

This command displays the table information that is specified on a real device. The format of output from this option is:

```
        {STARTED}                  {SEP     CHARS  FCB}
PRT rrr {DRAINED}  SYSTEM CLASS=c...{NOSEP             }

                                     {PURGE}
PRT rrr FLASH ffff  IMAGE imagelib {HOLD }
```

where:

rrr indicates the real device address.

STARTED indicates the device is available for spooling activity.

DRAINED indicates the device is not currently available for processing. A START command must be issued to activate the device.

c... indicates the classes serviced by the output device.

NOSEP indicates that a file separator is not desired for the output file.

The following information is displayed for the 3800 printer only:

CHARS indicates the name of the character arrangement table used for the separator page.

FCB indicates the value of the forms control buffer used for the file.

SEP indicates that a file separator is desired for the output file.

ffff indicates the forms overlay sheet contained in the device.

imagelib indicates the named system image library used for tables.

PURGE indicates a delete status for spool files containing load errors.

HOLD indicates a save status for spool files containing load errors.

QUERY SYSTEM raddr

This command requests the number of user minidisks residing on the physical disk located at raddr. The response for each minidisk is given in the following format:

```
        userid vaddr mode, ...
          .      .      .
          .      .      .
          .      .      .
```

where:

userid is the identification of the user who owns the minidisk.

vaddr is the virtual address by which the user refers to the minidisk.

mode is the type of access the user has: either R/O or R/W, or nnn for the number of cylinders of TDSK space allocated.

QUERY DUMP

```
type raddr DUMP UNIT {CP }
                     {ALL}
```

indicates that the device of device type "type" located at raddr is the system dump unit.

The format of the class D QUERY command is:

```
┌──────────────────────────────────────────────────────────────────────────┐
│           │ /                                  \                        │
│  Query    │ │  Files [CLass c] ┌      ┐          │                        │
│           │ │                  │userid│          │                        │
│           │ │                  │  *   │          │                        │
│           │ │                  └      ┘          │                        │
│           │ │           ┌┌        ┐         ┐    │                        │
│           │ │  Reader   ││ALL     │ [userid]│    │                        │
│           │ │  PUnch    ││CLass c │         │    │                        │
│           │ │           │└        ┘         │    │                        │
│           │ <           │  spoolid          │    >                        │
│           │ │           └                   ┘    │                        │
│           │ │           ┌┌        ┐         ┐    │                        │
│           │ │  Printer  ││ALL     │ [userid]│    │                        │
│           │ │           ││CLass c │         │    │                        │
│           │ │           ││TBL     │         │    │                        │
│           │ │           │└        ┘         │    │                        │
│           │ │           │  spoolid          │    │                        │
│           │ │           └                   ┘    │                        │
│           │ │  HOld                              │                        │
└──────────────────────────────────────────────────────────────────────────┘
```

where:

FILES displays the number of spooled input and output files. The class D user receives the total count in the system. Files that are currently being processed are not included in the totals.

CLASS c displays only the spool files of the specified class. If CLASS is omitted, then all spool classes are examined.

userid displays only the spool files owned by the specified userid. If userid is omitted, then spool files owned by all users are examined.

* displays only the spool files of the logon user who issued the QUERY command.

READER
RDR displays basic information concerning reader spool files.

PRINTER
PRT displays basic information concerning printer spool files.

PUNCH
PCH displays basic information concerning punch spool files.

Note: The basic information displayed is:

- Userid of the owner of the spool file. If examining files for a specific user (userid option), the userid indicates the originator of the spool file.
- Spool file spoolid number
- Class and originating device type
- Number of logical records in the file
- Number of copies specified for the file

- File hold status

ALL displays additional information for all spool files examined.

spoolid displays additional information for the specified spool file. The spool identification (spoolid) is a VM/370-generated sequential number assigned to each spool file.

The additional information displayed is:

- Date and time the file was created
- Filename and filetype of the file (if any)
- Distribution code of the file

TBL displays table information for 3800 spool files.

The table information displayed is:

- Flash name
- Name of the character arrangement table
- Forms control buffer (FCB)
- Name of the copy modification module
- Number of copies to be flashed (flash count)

HOLD displays a list of users whose output is being held by the HOLD command.

Responses to the Class D QUERY Command

QUERY FILES [CLass c] [userid]

```
FILES: {NO } RDR, {NO } PRT, {NO } PUN
       {nnn}      {nnn}      {nnn}
```

displays the total number of spool files in the system, of a particular class, or for a particular userid.

```
      {READER }
      {PRINTER}
QUERY {PUNCH  }
```

```
                                         ┌                               ┐
                                         │┌────Additional Information───┐│
                                         │▼                             ▼│
                                         │DATE  TIME     NAME  TYPE  DIST │
┌─────────Basic Information────────┐     │mm/dd hh:mm:ss name  type  code │
▼                                  ▼     │                                │
OWNERID[1] FILE CLASS RECDS  CPY HOLD    │┌────────Table Information─────┐│
userid   file a typ norecs nn  stat      │▼                             ▼│
                                         │FLASH  CHARS  FCB   MDFY  FLSHC │
                                         │ovly   ctab   plpi  cmod  cc    │
                                         └                                ┘
```

Only one file is listed for a QUERY READER, QUERY PRINTER, or QUERY PUNCH command if the spoolid operand is specified.

The DATE, TIME, NAME, TYPE, and DIST information is displayed only when the following commands are issued:

```
QUERY ⎧READER ⎫ ⎧ALL    ⎫
      ⎨PRINTER⎬ ⎩spoolid⎭
      ⎩PUNCH  ⎭
```

FLASH, CHARS, FCB, MDFY, and FLSHC information is displayed for the 3800 printer only. This information is obtained by issuing the following commands:

```
QUERY PRINTER TBL
```

where:

userid   is the identification of the user who owns the file.

file   is a unique, system-assigned number which is used by VM/370 to identify the file.

a   is the spool file class.

typ   is the originating device type (PRT, PUN, CON, or RDR).

norecs   is the number of logical records contained in the file.

nn   is the number of copies specified for the file. (Has no effect for reader files.)

stat   is the file hold status and is either

NONE - no hold
USER - user hold

mm/dd   is the date the file was created in month/day.

hh:mm:ss   is the actual time of the creation of the file in hours:minutes:seconds.

[1] OWNERID heading the title line for the spool file data is altered to ORIGINID when the userid operand is used. In that event, ORIGINID represents the originator of the file.

filename is the filename assigned to the file (if any). If the file has a 24-character data set name (dsname), only 20 characters are displayed. These characters extend from the "name" field through the "type" field.

filetype is the filetype assigned to the file (if any).

distcode is the distribution code of the file.

ovly is the one-to-four character name of the flash overlay sheet loaded into the 3800 printer.

ctab is the one-to-four character name of the character arrangement table to be used for the separator page.

plpi is the print lines/inch value to be used for the separator page.

cmod is the one-to-four character name of the copy modification module to be used for printing the file.

cc is the flash count for the file.

<u>QUERY HOLD</u>

```
HOLD : {NO } RDR, {NO } PRT, {NO } PUN
       {nnn}      {nnn}      {nnn}

         {ALL}
userid - {RDR} , ...
         {PRT}
         {PUN}
```

The first response displays the total number of files within the system which are retained in the system hold status. The second response indicates the type of hold (if any) for any user in the system for which HOLD is in effect. The user who issues QUERY HOLD may receive, depending upon the status of his spooled files, the first response, the second response, or both responses.

The format of the Class A, B, C, and E QUERY command is:

Query	PROCessor

<u>where</u>:

PROCESSOR displays the online processors in the system.

<u>Response to the Class A, B, C, and E Query Command</u>

PROCESSOR nn ONLINE [,PROCESSOR nn ONLINE]

displays the processors that are online. The first processor is the main processor.

The format of the Class A, B, C, D, E, and F QUERY command is:

```
┌──────────────────────────────────────────────────────────────────────┐
|  Query    |  ⎧LOGmsg           ⎫                                     |
|           |  ⎨Names            ⎬                                     |
|           |  ⎨Users   [userid] ⎬                                     |
|           |  ⎩userid           ⎭                                     |
└──────────────────────────────────────────────────────────────────────┘
```

where:

LOGMSG displays the system log messages of the day.

NAMES displays a list of all the users logged on and the real address of the line to which each is connected. If the user is disconnected, DSC is displayed instead of the line address.

USERS displays the number of logged-on users and the number of users dialed to other virtual machines. If userid is specified, the userid and device address of the user's terminal are displayed if he is logged on. If the specified user is not logged on, a message to that effect occurs. Use the USERS operand if the userid is the same as an operand (or its minimum truncation) of the QUERY command.

Note: It is possible for the number of users logged on as indicated by the NAMES operand to differ from the number logged on as indicated by the USERS operand. The number of users in the process of logging on and logging off accounts for this difference.

userid displays the userid and the device address of the user's terminal if he is logged on. If the user is not logged on, a message to this effect occurs.

Responses to the A, B, C, D, E, and F QUERY Command

QUERY LOGMSG

```
*  logmsg text line 1
   .           .
   .           .
   .           .
*  logmsg text line n

   logmsg additional text lines
   .                          .
   .                          .
   .                          .
```

All lines (both those with and without an asterisk) in the log message file are displayed.

QUERY NAMES

```
userid - {DSC  }, ...
.        {raddr}
.
.
userid - {DSC  }, ...
         {raddr}
```

Lists all logged-on users. If the user is currently connected, the real address to which he is connected is displayed (raddr). If he is not connected to the system, DSC is displayed.

QUERY USERS

```
nnn USERS, mmm DIALED
```

where:

nnn is the total number of logged on users.

mmm is the total number of users logically attached via the DIAL command to virtual machines.

Note: The term DIALED means that the line is not available to CP because it is logically attached to a logged-on user and is a part of that user's virtual machine operation.

QUERY userid

```
userid - raddr
```

displays the real address (raddr) to which the specified user is connected.

REPEAT

Privilege Class: D

Use the REPEAT command to:

1. Increase the number of copies of an output file beyond that quantity originally requested by the file's creator.

2. Place the current output file in a HOLD status, with or without increasing the number of copies to be created.

This command applies to files currently being written to a real punch or printer. The format of the REPEAT command is:

```
┌──────────────────────────────────────────────────────────────────────────┐
| REPeat       |             ┌           ┐                                 |
|              |             | ┌  ┐      |                                 |
|              |             | |nn|      |                                 |
|              |             | |1 |      |                                 |
|              |             | └  ┘      |                                 |
|              |   raddr     | ┌  ┐      |                                 |
|              |             | |nn| HOld |                                 |
|              |             | └  ┘      |                                 |
|              |             └           ┘                                 |
└──────────────────────────────────────────────────────────────────────────┘
```

where:

raddr is the address of the output device (printer or punch) whose current active spool file is to be repeated or held.

nn is the number of extra copies of the current file that are to be made. If nn is omitted and the HOLD operand is not specified, the number of extra copies defaults to 1. The maximum value that may be specified for nn is 99.

HOLD holds the file currently being printed or punched. If nn is not specified and HOLD is specified, when the last copy has been completed, the file is placed in system hold status until it is explicitly released.

Note: All copies indicated by a SPOOL command or by a previous REPEAT command are printed before the HOLD operand takes effect.

Responses

```
{PRT}       {REPEATED }
{PUN} raddr {FILE HELD} userid FILE = file RECDS = norecs COPY = nn a typ
```

The FILE HELD response is given in addition to the REPEATED response if the HOLD operand is specified.

where:

raddr is the real address of the device whose current output is being repeated.

userid is the identification of the user who is to receive the output.

file is the unique, system-assigned spoolid number by which VM/370 refers to the file.

norecs is the total number of logical records in the file.

nn is the remaining number of copies of the file to be created, where 01 is the last copy.

a is the spool file class.

typ is the originating device type (PRT, CON, or PUN).

SAVESYS

Privilege Class: E

Use the SAVESYS command to save a virtual machine storage space with registers and PSW as they currently exist. This command is used in the process of creating named systems. The format of the SAVESYS command is:

SAVESYS	systemname

where:

systemname

must be a predefined name representing a definition of installation requirements of the named system. The definition indicates the number of pages to be saved, the DASD volume on which the system is to be saved, and the shared segments (if any). SAVESYS should be invoked immediately after loading the system that is to be saved (while it is in a stopped state). Refer to the discussion of named systems in the *VM/370 System Programmer's Guide* for further information concerning saved systems.

Responses

SYSTEM SAVED

SET

Privilege Classes: A, B, or F

Use the SET command to establish system parameters. The SET command performs various functions to control the CP system and virtual machine options.

The format of the Class A SET command is:

```
┌───────────────────────────────────────────────────────────────────────┐
|              |  /                 ┌   ┐                  \            |
|              | /  AFFinity [userid] |ON |                   \           |
|              ||                     |OFF|                   |           |
|              ||                     |nn |                   |           |
|              ||                     └   ┘                   |           |
|              ||                                             |           |
|              ||   CPAssist         {ON }                    |           |
|              ||                    {OFF}                    |           |
|              ||                                             |           |
|              ||                     ┌   ┐                   |           |
|   SET        ||   FAVORed   userid  |xx |                   |           |
|              ||                     |OFF|                   |           |
|              ||                     └   ┘                   |           |
|              | \                                           /            |
|              | /                    ┌   ┐                   \           |
|              ||   JOurnal¹ {LOgon}  |ON |                   |           |
|              ||            {LInk }  |OFF|                   |           |
|              ||                     └   ┘                   |           |
|              ||                                             |           |
|              ||   REServe   userid {xx }                    |           |
|              ||                    {OFF}                    |           |
|              ||                                             |           |
|              ||   SASsist          {ON } [[PROC[nn]]        |           |
|              ||                    {OFF}                    |           |
|              ||                                             |           |
|              ||   PRIORity  userid   nn                     |           |
|              | \                                           /            |
|───────────────────────────────────────────────────────────────────────|
|¹The JOURNAL operand is valid only if STQUERY=YES is specified in        |
| the SYSJRL macro instruction DMKSYS.                                    |
└───────────────────────────────────────────────────────────────────────┘
```

where:

```
                    ┌   ┐
AFFINITY [userid]   |ON |
                    |OFF|
                    |nn |
                    └   ┘
```

(Attached processor system only) sets affinity for the specified userid. If userid is omitted, affinity is set or reset for the user who issued the command. If ON is specified (specifically or by default), the value in the user's directory entry is used. The value nn must be the hexadecimal (00—3F) processor address of an installed processor of an attached processor system. Affinity restricts execution of the virtual machine to the specified processor, but does not restrict CP code executed on behalf of a virtual machine to any processor.

```
CPASSIST {ON }
         {OFF}
```

sets CP assist to the active or the nonactive state. If CP assist is active, functions normally executed by various sections of CP routines are executed by processor hardware logic. This function made active on VM/370 initialization, a part of VM/370 Extended Control-Program Support (VM/370 ECPS), is available on System/370 models 135-3, 138, 145-3, and 148.

Expanded virtual machine assist, another part of VM/370 ECPS, is also controlled by the SET CPASSIST ON and OFF operands. Expanded virtual machine assist, an extension of function provided by virtual machine assist, is set to the active state upon VM/370 system initialization. Thereafter, to enable this function, both SET CPASSIST ON and SET SASSIST ON commands must be invoked. To set expanded virtual machine assist to the inactive state, use either SET CPASSIST OFF or SET SASSIST OFF.

```
               ┌   ┐
FAVORED userid │xx │
               │OFF│
               └   ┘
```

sets the favored execution performance option for the specified user (userid). The FAVORED function with percentage (xx) specified can be in effect for only one virtual machine at any time. If xx (percentage values from 1 to 99 are valid) is specified, VM/370 attempts to provide up to the specified percentage of processor time to that virtual machine, provided that virtual machine can fully utilize the processor time. At regular time intervals, the VM/370 dispatcher checks the processor time used by the particular virtual machine; if the percentage has been exceeded, the virtual machine is given his normal priority usage for the remainder of the interval. If the percentage is low, the virtual machine is assigned highest priority for the remainder of the interval. If OFF is specified, the specified virtual machine is removed from FAVORED status. If neither xx nor OFF is specified, the virtual machine designated is always dispatchable; that is, any time it is ready to use processor time, it is given preference over other users with the same priority. Multiple users may be classified as favored if they have no percentage assigned; however, only one user may be designated in a single SET command.

```
                  ┌   ┐
JOURNAL {LOGON}   │ON │
        {LINK }   │OFF│
                  └   ┘
```

sets the status of the LOGON and LINK journaling functions.

```
RESERVE userid {xx }
               {OFF}
```

sets the reserved page performance option for the specified user (userid). This performance option can be applied to only one virtual machine at any time. The specified virtual machine uses only the specified number of page frames (xx) for its exclusive use. If the number of page frames does not equal the most active pages, the virtual machine will experience performance degradation. The page frames are not locked and the virtual machine can also use other available page frames; however, no other virtual machine can use a reserved page frame unless all other available page frames have been exhausted. The maximum number of page frames reserved is limited by the size of the virtual machine. The OFF operand returns the reserved pages to the system.

```
SASSIST {ON } [[PROC]nn]
        {OFF}
```

controls whether virtual machine assist, a facility available on most VM/370 supported System/370 processors, is activated or deactivated for the entire system. In VM/370 attached processor applications, virtual machine assist may be set either ON or OFF for the desired processor by selecting the PROC nn operand. If PROC nn is not selected then SET SASSIST ON or OFF determines the setting of virtual machine assist for both the main and the attached processor. The values of nn are processor addresses (hexadecimal values 00 to 3F) established when the System/370 is installed. If nn equates to a value that is not the address of the main processor or attached processor, an error message is issued. Virtual machine assist increases the efficiency of CP processing of certain instructions, privileged instructions, and interrupts reflected to CP.

SET SASSIST in conjunction with CPASSIST also controls whether the VM/370 Extended Control-Program Support provided by System/370 Models 135-3, 138, 145-3, and the 148 is in an active or inactive state. For details on how this is accomplished, refer to the CPASSIST operand previously described. VM/370 Extended Control-Program Support simulates the CP processing of additional privileged instructions and extends the level of handling of other privileged instructions not covered by virtual machine assist. SET SASSIST ON or OFF also controls whether the virtual interval timer assist function of ECPS, a hardware updating facility for the interval timers of virtual machines, (virtual location X'50') is available to users. Use of this timer hardware update facility by the virtual machine is controlled by the class G command SET ASSIST TMR or SET ASSIST NOTMR; for details, see the _VM/370 CP Command Reference for General Users_.

Upon system initialization, the default value SET SASSIST is ON for VM/370 processors except the System/370 Model 155 II and the 165 II.

For details on the extent of virtual machine assist and the VM/370 Extended Control-Program Support, refer to the _VM/370 System Programmer's Guide_.

PRIORITY userid nn

sets the priority of the specified user (userid) to the desired value (nn). This value affects the user's dispatching priority in relation to other users in the system. The priority nn value is one of the factors considered in VM/370's dispatching algorithm. Generally, the lower the value of nn, the more favorable the user's position in relation to other users in VM/370's dispatch queues.

The format of the Class B SET command is:

```
┌──────────────────────────────────────────────────────────────────────────┐
|          |  /  LOGmsg     ┌     ┐            \                           |
|  SET     | |              |nn   |             |                          |
|          | |              |NULL |             |                          |
|          |  \             └     ┘            /                           |
|          |  /                      ┌    ┐    \                           |
|          | |   DUMP     (AUTO )    |CP  |     |                          |
|          | |            (raddr)    |ALL |     |                          |
|          |  \                      └    ┘    /                           |
└──────────────────────────────────────────────────────────────────────────┘
                          580        ALL
```

where:

```
       ┌     ┐
LOGMSG |nn   |
       |NULL |
       └     ┘
```

LOGMSG or LOGMSG nn causes CP to respond with the message LOGMSG: followed by a console read to allow the operator to enter the line. The maximum log message length per line is 76 characters, including spaces. Use LOGMSG with no operand to start a new log message or add to a log message. LOGMSG nn specifies that a change is to be made to existing line number nn in the log message. A null line entered terminates the LOGMSG entry (for adding or initializing) or deletes line nn if that option is used. LOGMSG NULL deletes the entire existing LOGMSG allowing a new LOGMSG to be created. Preceding the message text with an asterisk automatically displays the message on the console as soon as a user logs on to VM/370.

```
                ┌    ┐
DUMP  (AUTO )   |CP  |
      (raddr)   |ALL |
                └    ┘
```

designates the unit to receive the VM/370 system ABEND dump, where raddr is a real printer or a 9-track tape address. If the dump unit is a tape drive, the dump data must fit on one reel because VM/370 does not support multiple tape volumes. If AUTO is specified, the system dump unit is disk. The CP option dumps only storage occupied by the control program, whereas the ALL option dumps all storage.

The format of the Class F SET command is:

```
┌──────────────────────────────────────────────────────────────────────────┐
| SET | RECord ⎧OFF                                   ┌                 ┐ ⎫|
|     |        ⎨ON raddr LIMIT nn BYTE nn BIT n |⎧AND⎫ BYTE nn BIT n| ⎪|
|     |        ⎪                                  |⎩OR ⎭                | ⎬|
|     |        ⎩                                  └                 ┘ ⎭|
|     |                                                                    |
|     | MODE    ⎧RETRY⎫ ⎧Quiet ⎫ [cpuid]                                  |
|     |         ⎩MAIN ⎭ ⎩Record⎭                                          |
└──────────────────────────────────────────────────────────────────────────┘
```

where:

RECORD sets the recording mode for a device. ON allows the class F user to specify intensive recording mode for the device raddr. LIMIT identifies the value at which a recording will be made, and the BYTE and BIT operands define the test condition in the sense bytes. Two bits may be tested by an AND or OR relationship. The OFF function turns off intensive recording mode. Only one device at a time may have intensive recording specified. If SET RECORD OFF is not entered, intensive recording is automatically terminated after the accumulation of ten error records for that device.

MODE sets the error recording mode for soft errors. Soft errors are recoverable processor storage or instruction errors corrected by hardware or software logic.

MODE RETRY QUIET turns off error recording for soft machine check errors successfully corrected by instruction retry.

MODE RETRY RECORD resets the error threshold count to zero and records soft machine check errors (up to the threshold of 12), successfully corrected by instruction retry, on the error recording cylinder.

MODE MAIN QUIET turns off error recording for single bit main storage errors, successfully corrected by Error Correction Code logic.

MODE MAIN RECORD resets the error threshold count to zero and records up to 12 successfully corrected main storage errors on the error recording cylinder before entering MAIN QUIET mode.

Note: SET MODE MAIN is invalid for 3031, 3032, and 3033 processors.

cpuid in attached processor mode, indicates the address of the processor for which the command is executed. If no address is specified, the command is processed for both processors. If a value is specified, it must be in agreement with the processor designated identity specified at the time of the installation of the processor complex. Valid hexadecimal values are 00 to 3F.

Responses

There are no responses to the SET command. However, when the command is invoked, information messages related to the use of specific SET operands are displayed. These messages are described in the *VM/370 System Messages*.

SHUTDOWN

Privilege Class: A

Use the SHUTDOWN command to systematically end all VM/370 functions and to checkpoint the system for an eventual warm start. This is the normal method to shut the system down. All VM/370 functions are suspended and communications lines are disabled. Use the NETWORK SHUTDOWN command to stop communications on 370x or remote 3270 binary synchronous lines. The format of the SHUTDOWN command is:

```
 SHUTDOWN |
```

This command has no operands.

Usage Notes:

1. Because the SHUTDOWN command stops all virtual machine operation, any knowledge of an impending shutdown situation on the part of the system operator should be conveyed to the virtual machine users as soon as possible via appropriate LOGMSG information, messages and warnings so they may quiesce their virtual systems in an orderly manner.

2. In VM/370 installations that use the 3850 Mass Storage System, the VM/370 shutdown process issues a SUSPEND order to all 3851 devices to suspend the MSC interface. Note, the SUSPEND order may cause the shutdown process to take longer than usual as it causes the destaging of data on 3330V volumes.

Responses

```
SYSTEM SHUTDOWN COMPLETE
```

SPACE

Privilege Class: D

Use the SPACE command to force the output on the specified printer to be single spaced for the current active spool file, regardless of the carriage control commands contained in the actual file. The format of the SPACE command is:

```
┌─────────────────────────────────────────────────────────────────────┐
│ SPAce     │     raddr                                                │
└─────────────────────────────────────────────────────────────────────┘
```

where:

raddr is the real address (cuu) of the printer whose output is to be single spaced.

Responses

None.

START

Privilege Class: D

Use the START command to restart a spooling device after it has been drained, or to change the output class that it may service. If the device is busy when the command is given, the changed classes take effect after the current file is complete. The format of the START command is:

```
  ┌──────────────────────────────────────────────────────────────────────┐
  |          | ┌                                                ┐        |
  | STArt    | |  Printer                                       |        |
  |          | |  PUnch                                         |        |
  |          | |  Reader                  ┌               ┐     |        |
| |          | |  raddr [CLass c...] [NOsep] |FLash name     |  |        |
| |          | |                          |CHars name     |     |        |
| |          | |                          |FCB name       |     |        |
| |          | |                          |IMage namedsys |     |        |
| |          | |                          |PUrge          |     |        |
| |          | |                          └               ┘     |        |
  |          | |  ALL                                           |        |
  |          | └                                                ┘        |
  └──────────────────────────────────────────────────────────────────────┘
```

where:

PRINTER
PRT starts or changes the status of the printer.

PUNCH
PCH starts or changes the status of the punch.

READER
RDR starts or changes the status of the reader.

raddr is the address (cuu) of a particular spooling device which is to be started. More than one device may be specified.

CLASS c...
specifies the output classes the specified device is allowed to service. From one-to-four classes (single characters with no interrvening blanks) may be specified. If multiple classes are specified for a device, output files receive service by the device in the order that the classes were specified. For example, if the option:

    CLASS AD1

is specified for a printer, all class A files are spooled before class D files, and all class D before class 1.

Note: a class of * can be specified for printers or punches so that any spool file class can be serviced by the device for output.

NOSEP specifies that no file separator is desired for the output files. If not specified, a separator is produced. If NOSEP is specified for the punch, separator cards are not punched; instead, the operator receives the message:

    PUN raddr START FOR OUTPUT

The punch can then be started for the next file by issuing the CP START command or physically activating the START/STOP key on the device.

FLASH name
specifies the flash overlay sheet loaded into the device. Name may be from one-to-four characters in length.[1]

CHARS name
specifies the character arrangement table used for the separator page. Name may be from one-to-four characters in length.[1]

FCB name specifies the forms control buffer used for the separator page and also for any spool file for which an FCB has not been specified (through the use of the SPOOL or CHANGE command). name may be number 6, 8, or 12. These specifications mean: print the entire spool file at 6, 8, or 12 lines per inch, regardless of the size of the paper that is currently loaded. Use of the unique numbers will never cause an FCB-forms mismatch.[1]

IMAGE namedsys
specifies the named system image library used for tables. Namedsys may be from one-to-eight characters in length and must be a valid named system.[1]

PURGE specifies that all files with load errors be purged. If not specified, all files will be held.

<u>Note</u>: The FLASH, CHARS, FCB, IMAGE, and PURGE options are valid only for the 3800 printer.

ALL starts or changes the status of all readers, printers, and punches.

<u>Note</u>: Printers and punches that are in a ready status with files available for processing immediately start to produce output when the START command is issued. Readers, however, do not start until a card deck is supplied and the device is made ready.

<u>Responses</u>

```
{PRT} raddr STARTED SYSTEM   CLASS = a... {SEP  }
{PUN}                                     {NOSEP}

 RDR  raddr STARTED SYSTEM
```

<u>where</u>:

raddr is the real device address of the device started.

a... indicates the classes to be serviced by the device.

[1] If not specified, the last value specified by the installation is used.

STCP

Privilege Class: C

Use the STCP command to alter the contents of real storage. The real PSW or real registers cannot be altered with this command. In a system running in attached processor mode, data in shared pages cannot be modified by the use of this command. The format of the STCP command is:

```
┌──────────────────────────────────────────────────────────────────────┐
│ STCP      │ / (   hexloc \                                   \       │
│           │ | |  Mhexloc |                                    |      │
│           │ | |  Nhexloc < hexword1 [hexword2... ]            |      │
│           │ | |  Lhexloc |                                    |      │
│           │ | | MLhexloc |                                    |      │
│           │ < | NLhexloc /                                    >      │
│           │ |                                                 |      │
│           │ |                                                 |      │
│           │ | (  Shexloc \                                    |      │
│           │ | < MShexloc > hexdata                            |      │
│           │ \ ( NShexloc /                                   /       │
└──────────────────────────────────────────────────────────────────────┘
```

where:

hexloc
Mhexloc
Nhexloc
Lhexloc
MLhexloc
NLhexloc

stores the data given in hexword1 [hexword2...] in successive fullword locations starting at the address specified by hexloc. The smallest group of hexadecimal values that can be stored using this specification is one fullword. Data is aligned to the nearest fullword boundary. If the data being stored is less than a fullword (eight hexadecimal digits), it is right-adjusted in the word and the high order bytes of the word are filled with zeros.

In attached processor systems, hexloc and Lhexloc values are treated as absolute addresses. The letter M prefixing the hexadecimal address causes that address to be interpreted through the prefix register of the main processor. The letter N prefixing an address functions the same way through the prefix register for the attached processor. N is valid while the attached processor is in operation. M is valid if the system was generated for the attached processor mode of operation.

Shexloc
MShexloc
NShexloc

stores the data given in hexdata in the address specified by hexloc without word alignment. The shortest string that can be stored is one byte (two hexadecimal digits). If the string contains an odd number of characters, the last character is not stored. An error message occurs and the function ends.

MShexloc and NShexloc operands are used in attached processor VM/370 applications. By prefixing Shexloc with M or N, the specified hexadecimal addresses are interpreted through the associated prefix registers of the main or the attached processor, respectively.

hexword specifies up to eight hexadecimal digits. If less than eight digits are specified, the string is right-justified in a fullword and left-filled with zeros. If two or more hexwords are specified, they must be separated by at least one blank.

hexdata specifies a string of two or more hexadecimal digits with no embedded blanks.

Usage Notes:

```
STCP  N64 20000
```

stores X'00020000' into the second word of the SVC new PSW in the PSA (prefix storage area) of the attached processor.

Responses

```
STORE COMPLETE
```

TRANSFER

__Privilege Class__: D

Use the TRANSFER command to direct one or more reader spool files to a specified user's virtual reader or to reclaim reader spool files that were previously sent to one or more users. The file is deleted from the sender's virtual card reader and is available in the recipient's spooled reader. The format of the TRANSFER command is:

```
┌──────────────────────────────────────────────────────────────────┐
│            │  ┌       ┐   ┌         ┐   ┌      ┐   ┌        ┐    │
│  TRANsfer  │  │userid │  ⎨ spoolid  ⎬   │To    │  ⎨ userid  ⎬    │
│            │  │SYSTEM │  ⎨ CLass c  ⎬   │FROM  │  ⎨ ALL     ⎬    │
│            │  └       ┘  ⎨ ALL      ⎬   └      ┘                 │
└──────────────────────────────────────────────────────────────────┘
```

__where__:

userid   is the user identification of the user whose reader spool files are to be transferred or reclaimed.

SYSTEM   allows the spooling operator to manipulate all reader spool files in the system regardless of the userid.

spoolid   is the identification of the file to be directed to or from the named userid.

CLass c   transfers all files of the given class.

ALL   transfers all reader spool files.

```
To {userid}
   {ALL   }
```

is the user to whom the files are to be directed. If the optional keyword TO is omitted, userid cannot be T or TO. ALL cannot be specified with TO.

```
FROM {userid}
     {ALL   }
```

is the user from whom the spool reader files are to be reclaimed. ALL may be specified to reclaim files from all users that were originated by the userid specified first.

__Responses__

```
RDR FILE spoolid TRANSFERRED {TO  } userid
                             {FROM}
```

__where__:

spoolid   is the reader spool file number of the file that is transferred. The number does not change.

TO   is the response to the user that currently owns the spool file and userid is the recipient.

FROM   is the response to the user that receives the transferred spool file and the userid of the sender.

$\left\{\begin{matrix}\texttt{nnnn}\\ \texttt{NO}\end{matrix}\right\}$ FILES TRANSFERRED

This is the response to the user who issued the TRANSFER command.

Note: This response is suppressed if the virtual machine user has issued the SET IMSG OFF command.

The system operator, monitoring the RSCS virtual machine for his installation, should not specify this opticn because information messages are required for accurate monitoring of RSCS system activity.

UNLOCK

Privilege Class: A

Use the UNLOCK command to unlock page frames previously locked by a LOCK command. The format of the UNLOCK command is:

```
┌──────────────────────────────────────────────────────────────────────┐
| UNLOCK    |  ( ( userid )                       )                     |
|           |  | {        } firstpage lastpage    |                     |
|           |  ) ( SYSTEM )                       (                     |
|           |  |                                  |                     |
|           |  ( { VIRT=REAL }                    )                     |
|           |  ( { V=R       }                    )                     |
└──────────────────────────────────────────────────────────────────────┘
```

where:

userid   is the user identification of the virtual user previously specified in a LOCK command.

SYSTEM   specifies to unlock a previously locked pageable CP nucleus page.

firstpage
is the hexadecimal value of the first page to be unlocked.

lastpage
is the hexadecimal value of the last page to be unlocked.

Note: For firstpage and lastpage, only the page numbers are specified. For example, to unlock USERA's virtual storage locations X'12000' to X'2CFFF', the following command should be issued:

```
UNLOCK USERA 12 2C
```

VIRT=REAL
V=R
releases for normal paging use all the page frames normally for the virtual=real machine if the area is not occupied by a virtual=real virtual machine.

Note: Once the virtual=real area is released, it cannot be reactivated to run another virtual=real machine until VM/370 is loaded again.

Response

COMMAND COMPLETE

occurs even if some (or all) of the pages unlocked were not previously locked.

VARY

Privilege Class: B

Use the VARY command to mark a device available or unavailable for use by a user or the control program. (For 370x resources running in NCP or PEP mode, or remote 3270 resources, use the NETWORK VARY command.) Spooled unit record devices marked offline are remembered by the VM/370 system through any system failures. A shutdown and subsequent cold start assumes that all devices are available. The format of the VARY command is:

```
┌──────────────────────────────────────────────────────────────────────────────┐
│ VARY      │ ┌ONLine ┐ ┌raddr-raddr  ┐                                        │
│           │ └OFfline┘ ┤raddr...     ├                                        │
│           │           └PROCessor nn ┘                                        │
└──────────────────────────────────────────────────────────────────────────────┘
```

where:

ONLINE makes a device(s) previously offline available.

OFFLINE makes the device(s) unavailable.

raddr is the real address of the device whose status is to be altered. Multiple devices as well as a range of device addresses may be specified.

PROCESSOR nn
is the processor to be made available or unavailable. The value of nn is the processor address (hexadecimal value from 00 through 3F) established when the prccessor is installed.

Usage Notes:

1. If the device specified is a 2305, the address must be for the first of 8 "exposures," that is, address xx0 or xx8.

2. Execution of the VARY device address (raddr) command is accomplished in two phases. In phase one, the command line is checked to see if it is syntactically correct. If it is not, the command line is rejected. If the command line is correct, then the VARY ONLINE/OFFLINE process begins. When processing multiple devices or a range of devices, VARY command processing continues regardless of whether or not an error is encountered when attempting to vary any one of the devices online/offline. An error message will be issued for every device that encounters an error situation. An informational message will be issued for those devices in the range or those multiple addresses on the command line that could be varied online/offline.

3. Execution of the VARY PROCESSOR nn command is accomplished as follows. Command processing checks the command authority and syntax and that the specified processor number is valid. If the command is to vary the processor online, a check will be made to see if it is already online. If it is not, and if the processor is available and operational, command processing will establish prefixing, initialize the control registers and prefix areas, and set the clock comparator and CPU timer; the system can then function in attached processor mode.

If the command is to vary the processor offline, a check will be made to see that the specified processor is not the main processor and that it is online and operational. It will then be taken offline, processing will be quiesced on the specified processor, and the control registers and prefix register will be cleared. Those users having affinity to the specified processor will be notified that the affinity is set off; these users will then be put in console function mode.

4. It is recommended that if system performance monitoring is taking place, it should be stopped with a MON STOP command and restarted after the configuration change is complete.

5. If the system mode switch is set to UP mode, make sure you stop the online processor, set the switch to MP, and then vary the offline processor online.

<u>Responses</u>

```
{raddr...    } VARIED {ONLINE }
{raddr-raddr}         {OFFLINE}
```

One or more of the previous message responses will be sent to the system operator depending upon errors encountered in the command execution phase; see Usage Note 2.

RADDR MISSING OR INVALID

is issued if the PROCESSOR operand was entered without a processor number, or if the specified number is invalid.

PROCESSOR nn DOES NOT EXIST

is issued if the processor specified by nn in the VARY PROCESSOR nn command is not defined to the system.

VARY PROCESSOR COMMAND FAILED

if the VARY OFFLINE PROCESSOR nn command was entered, this response indicates the specified processor is in uniprocessor mode or not online, or it is the main processor.

if a VARY ONLINE PROCESSOR nn command was entered, this response indicates that the system is not generated for AP mode, the MP function is not installed, or the system is already in AP mode.

PROCESSOR nn ONLINE

indicates that the specified processor is now online and operational.

PROCESSOR nn OFFLINE

indicates that the specified processor is now offline and not operational.

WARNING

Privilege Classes: A or B

Use the WARNING command to transmit high-priority messages to a specified user or to all users. The WARNING does not wait for input operations at the terminal to complete, but breaks in on the line currently being displayed and sends the message immediately (unless the SET WNG OFF command was previously issued for the virtual machine). The format of the WARNING command is:

```
┌──────────────────────────────────────────────────────────────────────┐
| Warning   |  ( userid   )   msgtext                                  |
| WNG       | <  OPerator  >                                           |
|           |  ( ALL      )                                            |
└──────────────────────────────────────────────────────────────────────┘
```

where:

userid — is the identification of the user who is to receive the message.

OPERATOR — sends the message to the primary system operator regardless of his userid.

ALL — allows the system operator to broadcast the message to all users receiving warning messages.

msgtext — is the text of the message to be sent.

Responses

```
hh:mm:ss
WNG FROM userid: msgtext
```

is the response to the users who are to receive the warning message. The hh:mm:ss represents the time in hours, minutes, and seconds when the message was sent to the user.

Note: WARNING commands ring the alarm, if the ALARM feature is installed, and put the screen in a HOLD status (for 3270). When this occurs, press the CANCEL key to put the screen in a RUNNING status.

Section 4. VM/370 Service Programs

Controlling the 3704/3705 Communications Control Program

Operational control of the 3704/3705 Communications Controller (hereinafter referred to as 3705) resides with the system operator via the NETWORK command. In addition, the behavior of the 3705 device is, for a large part, dependent upon the communications control program with which it was loaded. The communication control programs are created, assembled, or modified by a set of CMS commands specifically designed for this purpose. These commands are:

```
ASM3705
GEN3705
LKED
SAVENCP
ZAP
```

The cited commands, described in greater detail in the _VM/370 System Programmer's Guide_, are the system programmer's responsibility. The 3705 NETWORK and NCPDUMP commands, however, are the responsibility of the VM/370 system operator and are described elsewhere in this publication.

The 3704/3705 Dynamic Trace Facility is supported for virtual machine operation. However, if the 3704/3705 in emulator mode is not dedicated to the user invoking the trace, the user is required to have in his directory a privilege class other than (or in addition to) G.

Using CMS To Alter VM/370 Files

The VM/370 system operator is usually assigned a virtual machine with a large DASD storage allotment. This DASD storage can be assigned to other users as a temporary work area or for short-term DASD storage. In addition, the operator's virtual machine can record system statistical data, or create and maintain directory files and other system files.

For whatever reason the operator uses the system, it is probable that some files will need to be modified at some time. One of the easiest ways to modify such system files is to use the CMS Editor (EDIT) facilities.

The CMS EDIT facilities provide ways to create and modify all types of CMS files. For a complete description of the CMS EDIT facilities and commands, see the _VM/370 CMS User's Guide_ and the _VM/370 CMS Command and Macro Reference_.

OTHER CMS COMMANDS

There may be occasions, caused by user demands, when other facilities of CMS may be invoked. For example, to create punched output of a file for use by another system, or to create a history file. The commands likely to be needed by the system operator are the following:

CMSBATCH
COMPARE
COPYFILE
ERASE
FORMAT
LISTDS
LISTFILE
MOVEFILE
PRINT
PUNCH
QUERY
READCARD
SET
STATE
TAPE
TAPPDS
TYPE
UPDATE

A complete description of these and other commands and their usage is contained in the _VM/370 CMS User's Guide_ and the _VM/370 CMS Command and Macro Reference_.

DASD Dump Restore (DDR) Service Program and How To Use It

Use the DASD Dump Restore (DDR) program to dump, restore, copy, or print VM/370 user minidisks. The DDR program may run as a standalone program, or under CMS via the DDR command.

The DDR program has five functions:

1. Dumps part or all of the data from a DASD device to tape.
2. Transfers data from tapes created by the DDR dump function to a direct access device. The direct access device must be of the same type as that which originally contained the data.
3. Copies data from one device to another of the same type. Data may be reordered, by cylinder, when copied from disk to disk. In order to copy one tape to another, the original tape must have been created by the DDR DUMP function.
4. Prints selected parts of DASD and tape records in hexadecimal and EBCDIC on the virtual printer.
5. Displays selected parts of DASD and tape records in hexadecimal and EBCDIC on the terminal.

To generate the VM/370 starter system from the distribution tape, the standalone RESTORE function must be used.

INVOKING DDR UNDER CMS

The format of the DDR command is:

```
┌──────────────────────────────────────────────────────────────┐
│            │       ┌  ┐                                       │
│   DDR      │ [fn  ft │fm│ ]                                   │
│            │       │* │                                       │
│            │       └  ┘                                       │
└──────────────────────────────────────────────────────────────┘
```

where:

```
      ┌  ┐
fn ft │fm│  is the identification of the file containing the control
      │* │  statements for the DDR program. If no file
      └  ┘  identification is provided, the DDR program attempts to
            obtain control statements from the console. The filemode
            defaults to an asterisk (*) if a value is not provided.
```

Note: If you use the CMS DDR command, CMS ignores the SYSPRINT control statement and directs the output to the CMS printer 00E.

INVOKING DDR AS A STANDALONE PROGRAM

To use DDR as a standalone program, the operator should IPL it from a real or virtual IPL device as he would any other standalone program. Then indicate where the DDR program is to obtain its control statements by responding to prompting messages at the console.

Notes:

1. Be aware that DDR when run as a standalone program has only the most elementary error recovery support. However, when DDR is invoked in CMS, in a virtual machine environment, the I/O operation is performed by CP which has better error recovery facilities.

2. When running standalone, DDR will search for a console at address 009 or 01F. If these consoles are not operational, the program will enter a wait state, waiting for an interrupt to identify the console. If any nonconsole—type device is physically connected to address 009 or 01F, it must be made nonoperational or the results will be unpredictable.

3. The standalone DDR utility will not support cylinder faults for MSS virtual volumes when performing the DUMP or COPY functions.

DDR CONTROL STATEMENTS

DDR control statements describe the intended processing and the needed I/O devices. I/O definition statements must be specified first.

All control statements may be entered from either the console or the card reader. Only columns 1 to 71 are inspected by the program. All data after the last operand in a statement is ignored. An output tape must have the DASD cylinder header records in ascending sequences; therefore, the extents must be entered in sequence by cylinder. Only one type of function — dump, restore, or copy — may be performed in one execution, but up to 20 statements describing cylinder extents may be entered. The function statements are delimited by (preceded by) an INPUT and OUTPUT statement. If additional functions are to be performed, the sequence of control cards must be repeated. If you do not use INPUT or OUTPUT control statements to separate the functions you specify when the input is read from a card reader or CMS file, an error message (DMKDDR702E) is displayed. However, the remainder of the input stream will be checked for proper syntax, but no further DDR operations will be performed. Only those I/O devices defined by the INPUT statement and the OUTPUT statement must be redefined in subsequent steps. The SYSPRINT I/O definition remains the same.

To return to CMS, enter a null line (carriage return) in response to the prompting message (ENTER:). To return directly to CP, key in #CP.

The PRINT and TYPE statements work differently from other DDR control statements in that they operate on only one data extent at a time and it is not necessary to respecify the INPUT statement ahead of each PRINT or TYPE statement. If the input is from a tape created by the dump function, the tape must be positioned at the header record for each step. The PRINT and TYPE statements have an implied output of either the console (TYPE) or system printer (PRINT), so no OUTPUT statement is required.

I/O DEFINITION STATEMENTS

The I/O definition statements describe the tape, DASD, and printer devices used while executing the DASD Dump Restore program.

INPUT/OUTPUT Control Statement

An INPUT or OUTPUT statement describes each tape and DASD unit used. The format of the INPUT/OUTPUT statement is:

```
┌────────────┬──────────────────────────────────────────────────────┐
│            │              ┌       ┐                                │
│  INput     │  cuu  type   │volser│  [(options...)]                │
│  OUTput    │              │altape│                                │
│            │              └       ┘                                │
│            │       Options:                                        │
│            │       ┌          ┐ ┌             ┐ ┌        ┐         │
│            │       │SKip  nn  │ │MOde   6250  │ │REWind│         │
│            │       │SKip  0   │ │MOde   1600  │ │UNload│         │
│            │       └          ┘ │MOde    800  │ │LEave │         │
│            │                    └             ┘ └        ┘         │
└────────────┴──────────────────────────────────────────────────────┘
```

where:

INPUT indicates that the device described is an input device.

OUTPUT indicates that the device described is an output device.

Note: If the output device is a DASD device and DDR is running under CMS, the device is released using the CMS RELEASE command function and DDR processing continues.

cuu is the unit address of the device.

type is the device type (2314, 2319, 3330, 3330-11, 3340-35, 3340-70, 3350, 2305-1, 2305-2, 2400, 2420, or 3420) (no 7-track support for any tape devices). Specify a 3410 as a 3420. Specify a 3340-70F as a 3340-70, and a 3333 as a 3330. Specify a 3350 that is in 3330-1 or 3330-11 compatibility mode as a 3330 or 3330-11. Specify a 3344 as a 3340-70, and specify 3350 for a 3350 operating in native mode (as opposed to compatibility mode).

Note: The DASD Dump Restore (DDR) program, executing in a virtual machine, uses I/O DIAGNOSE 20 to perform I/O operations on tape and DASD devices. DDR under CMS requires that the device type entered agree with the device type of the real device as recognized by VM/370. If there is a conflict with device types, the following message is issued:

```
DMKDDR708E INVALID OPTION
```

However, if DDR executes standalone in a virtual machine, DDR uses DIAGNOSE 20 to perform the I/O operation if the device types agree. If the device types do not agree, error message DMKDDR708E is issued.

volser is the volume serial number of a DASD device. If the keyword "SCRATCH" is specified instead of the volume serial number, no label verification is performed.

altape is the address of an alternate tape drive.

Note: If multiple reels of tape are required and "altape" is not specified, DDR types the following at the end of the reel:

```
END OF VOLUME CYL xxx HD xxx, MOUNT NEXT TAPE
```

After the new tape is mounted, DDR continues automatically.

Options:

SKIP nn
0
forward spaces nn files on the tape. nn is any number up to 255. The SKIP option is reset to zero after the tape has been positioned.

```
MODE ┌    ┐
     |6250|
     |1600|
     | 800|
     └    ┘
```

causes all output tapes that are opened for the first time and at the load point to be written or read in the specified density. All subsequent tapes mounted are also set to the specified density. If no mode option is specified, then no mode set is performed and the density setting remains as it previously was.

REWIND rewinds the tape at the end of a function.

UNLOAD rewinds and unloads the tape at the end of a function.

LEAVE leaves the tape positioned at the end of the file at the end of a function.

Notes:

1. When the wrong input tape is mounted, the message DMKDDR709E is displayed and the tape will rewind and unload regardless of options REWIND, UNLOAD, or LEAVE being specified.

2. If DDR is executed from CMS, failure to attach the tape drive or the disk device (or both) to your virtual machine prior to invoking the input/output statement causes the following response to be displayed:

```
INVALID INPUT OR OUTPUT DEFINITION
```

SYSPRINT Control Statement

Use the SYSPRINT control statement (in the standalone DDR virtual machine only) to describe the printer that is to print data extents specified by the PRINT statement. It also can print a map of the cylinder extents from the DUMP, RESTORE, or COPY statement. If the SYSPRINT statement is not provided, the printer assignment defaults to 00E. CMS ignores the SYSPRINT statement when you invoke DDR as a command under CMS, and CMS always directs the output to 00E. The format of the SYSPRINT control statement is:

SYsprint	cuu

where:

cuu specifies the unit address of the device.

Function Statements

The function statements tell the DDR program what action to perform. The function commands also describe the extents to be dumped, copied, or restored. The format of the DUMP/COPY/RESTORE control statement is:

```
┌──────────────────────────────────────────────────────────────────────────┐
|           | ┌                                            ┐               |
|   DUmp    | |cyl1 [To]  [cyl2 [Reorder] [To] [cyl3]]     |               |
|   COpy    | |CPvol                                       |               |
|   REstore | |ALL                                         |               |
|           | |NUcleus                                     |               |
|           | └                                            ┘               |
└──────────────────────────────────────────────────────────────────────────┘
```

where:

DUMP requests the program to move data from a direct access volume onto a magnetic tape or tapes. The data is moved cylinder by cylinder. Any number of cylinders may be moved. The format of the resulting tape is:

Record 1: a volume header record, consisting of data describing the volumes.

Record 2: a track header record, consisting of a list of count fields to restore the track, and the number of data records written on tape. After the last count field the record contains key and data records to fill the 4K buffer.

Record 3: track data records, consisting of key and data records packed into 4K blocks, with the last record truncated.

Record 4: either the end-of-volume (EOV) or end-of-job (EOJ) trailer label. The end-of-volume label contains the same information as the next volume header record, except that the ID field contains EOV. The end-of-job trailer label contains the same information as record 1 except that the cylinder number field contains the disk address of the last record on tape and the ID field contains EOJ.

COPY requests the program to copy data from one device to another device of the same or equivalent type. Data may be recorded on a cylinder basis from input device to output device. A tape-to-tape copy can be accomplished only with data dumped by this program.

RESTORE requests the program to return data that has been dumped by this program. Data can be restored only to a DASD volume of the same or equivalent device type from which it was dumped. It is possible to dump from a real disk and restore to a minidisk as long as the device types are the same.

cyl1 [TO] [cyl2 [REORDER] [TO] [cyl3]]
Only those cylinders specified are moved, starting with the first track of the first cylinder (cyl1), and ending with the last track of the second cylinder (cyl2). The REORDER operand causes the output to be reordered, that is, moved to different cylinders, starting at the specified cylinder (cyl3) or at the starting cylinder (cyl1) if cyl3 is not specified. The REORDER operand must not be specified unless specified limits are defined for the operation; the starting and, if required, ending cylinders (cyl1 and cyl2) must be specified. Note that if the input device cylinder extents exceed the number of

cylinders specified on the output device, an error message results.

Caution: The REORDER operand is intended to be used to move minidisks to new locations; it is not intended for re-locating non-minidisk cylinders. To understand the difference, consider a 10 cylinder minidisk. Its cylinders are numbered 0-9 and the count fields of its records refer to cylinders 0-9. Furthermore, if the minidisk contains location-dependent data, then references to cylinders 0-9 will be hidden within that data. When REORDER is used to move the minidisk to a new real location, the minidisk cylinders are nevertheless still regarded as being cylinders 0-9 and there is no need to change the cylinder numbers in the count field of the records. On the other hand, when moving non-minidisk cylinders, one would generally want the count fields of the moved records to reflect the new cylinder addresses; but REORDER keeps the original cylinder numbers in the count fields.

CPVOL specifies that cylinder 0 and all active directory and permanent disk space are to be copied, dumped, or restored. This indicates that both source and target disk must be in CP format, that is, the CP Format/Allocate program must have formatted them.

ALL specifies that the operation is to be performed on all cylinders, except alternate track cylinders on some devices. (See "Restrictions" below.)

Note: The occurrence of message DMKDDR705E (issued upon completion of the copy restore or dump operation) indicates that an attempt was made to copy restore or dump the contents of cylinders beyond the extents of the designated minidisk.

NUCLEUS specifies that record 2 on cylinder 0, track 0 and the nucleus cylinders are dumped, copied, or restored.

Restrictions:

- Each track processed by this utility must have a valid home address on it containing the real cylinder and track location. Even when restoring and copying data to a track it must have a pre-existing home address on it.

- Each track on an input DASD device must have a valid record zero on it, with no more than eight bytes in the key and data fields of the record. Each track on an output DASD device must also have a valid record zero on it unless that device is 2314, 2319, 2305-1, or 2305-2.

- Flagged tracks are treated just as any other track for all 2314, 2319, and 2305 devices. That is, no attempt is made to substitute the alternate track data when a defective primary track is read. In addition, tracks are not inspected to determine whether they were previously flagged when written. Therefore, volumes containing flagged tracks should be restored to the same cylinders of the volume from which they were dumped. The message DMKDDR715E occurs each time a defective track is dumped, copied or restored, and the operation continues. When ALL is specified for these device types, both the primary cylinders and the high-order cylinders normally reserved for alternate tracks are dumped, copied, or restored.

- Flagged tracks on 3330, 3340, and 3350 devices are handled so that data is transferred to or from the assigned alternate track in place of the defective track. (For 3330 and 3350 this is accomplished automatically by the hardware of the control unit while for the 3340 and 3344 it is accomplished through software.) The tape created by dumping one of these types of DASD devices will appear as if it had been dumped from a defect-free device and the tape can be restored to any device of the same type, even though that device might not have the same tracks flagged defective as the original device had. (The COPY function works this way also.) If a track is flagged as defective, but has no alternate assigned, a warning message is issued and the only data transferred is the home address record and record zero. When ALL is specified for these device types, only the primary cylinders are processed; the cylinders reserved for alternate tracks are not processed except that an assigned alternate track is processed whenever the corresponding defective track is processed. However, by specifying the cylinder range explicitly (cyl1 to cyl2 format), all cylinders, including cylinders in the alternate track area, can be dumped or copied from. But these same cylinders cannot be restored to or copied to explicitly. It is intended that explicitly dumped cylinders in the alternate track area will be restored to another area via the REORDER operand. The only reason the explicit dumping and copying of cylinders from the alternate track area is allowed at all is to facilitate conversion of 3340 and 3344 disks that were written using early releases of VM/370. Prior to Release 5 PLC 6, it was sometimes the practice to treat the alternate track area as if it was just another primary cylinder and to allow minidisks to be located there. But this is no longer permitted.

__Example__:

```
     INPUT 191 3330 SYSRES
     OUTPUT 180 2400 181 (MODE 800
     SYSPRINT 00F
     DUMP CPVOL
     INPUT 130 3330 MINI01
     DUMP 1 TO 50 REORDER 51
     60 70 101
```

This example sets the density to 800 bpi, then dumps all pertinent data from the volume labeled SYSRES onto the tape that is mounted on unit 180. If the program runs out of space on the first tape, it continues dumping onto the alternate device (181). A map of the dumped cylinders is printed on unit 00F while the program is dumping. When the first function is complete, the volume labeled MINI01 is dumped onto a new tape. Its cylinder header records are labeled 51 to 100. A map of the dumped cylinders is printed on unit 00F. Next, cylinders 60 to 70 are dumped and labeled 101 to 111. This extent is added to the cylinder map on unit 00F. When the DDR processing is complete, the tapes are unloaded and the program stops.

If cylinder extents are being defined from the console, the user need only enter DUMP, COPY or RESTORE on the command line. The following is displayed:

```
ENTER CYLINDER EXTENTS
ENTER:
```

For any extent after the first extent, the message

```
ENTER NEXT EXTENT OR NULL LINE
ENTER:
```

is displayed.

The user may then enter additional extents to be dumped, restored, or copied. A null line causes the job step to start.

<u>Notes</u>:

1. When a cylinder map is printed on the virtual printer (00F as in the previous example) a heading precedes the map information. Module DMKDDR controls the disk, time and zone printed in the heading. Your installation must apply a local modification to DMKDDR to ensure that local time, rather than GMT (Greenwich Meridian Time), is printed in the heading.

2. Attempts to restore cylinders beyond the capacity that had been recorded on the tape produces a successful EOJ, but the printout only indicates the last cylinder found on the tape.

<u>PRINT/TYPE Function Statement</u>

Use the PRINT and TYPE function statement to print or type (display) a hexadecimal and EBCDIC translation of each record specified. The first of a group of PRINT or TYPE statements must be preceded by an INPUT statement defining either a direct access device or a tape. The output is directed to the system console for the TYPE function, or to the SYSPRINT device for the PRINT function. (This does not cause redefinition of the output unit definition.) PRINT and TYPE may be used to display the contents of any track including those in the alternate track cylinders. For 3330, 3340, and 3350 devices, the following is displayed when alternate tracks are involved:

- When displaying a defective track that has a properly assigned alternate, the home address record displayed is taken from the defective track while record zero and all other records are taken from the alternate. The "defective" flag, visable in the displayed home address, is the only hint that this is not a normal track.

- When displaying a flagged defective track which does not have a proper alternate, only the home address record and record zero are displayed, and they are both taken from the defective track.

- When displaying an alternate track explicitly, all data displayed is from that track.

The format of the PRINT/TYPE control statement is:

```
┌──────────────────────────────────────────────────────────────────────────┐
│  PRint    │ cyl1 [hh1 [rr1]] [To cyl2 [hh2 [rr2 ]]] [ (options...[) ]] │
│  TYpe     │                                                              │
│           │          options:                                            │
│           │          [Hex] [Graphic]  [Count]                            │
└──────────────────────────────────────────────────────────────────────────┘
```

where:

cyl1 is the starting cylinder.

hh1 is the starting track. If present, it must follow the cyl1 operand. The default is track zero.

rr1 is the starting record. If present, it must follow the hh1 operand. The default is home address and record zero.

TO cyl2 is the ending cylinder. If more than one cylinder is to be printed or typed, "TO cyl2" must be specified.

hh2 is the ending track. If present, it must follow the cyl2 operand. The default is the last track on the ending cylinder.

rr2 is the record ID of the last record to print. The default is the last record on the ending track.

Options:

HEX prints or displays a hexadecimal representation of each record specified.

GRAPHIC prints or displays an EBCDIC translation of each record specified.

COUNT prints or displays only the count field for each record specified.

Usage

If the TYPE statement follows the occurrence of error message DMKDDR705E and specifies the same cylinder, track, and record extents indicated in the error message, the contents of the printed record must be interpreted in the context of the I/O error information given in the initial message.

Examples

PRINT 0 TO 3

Prints all of the records from cylinders 0, 1, 2, and 3.

PRINT 0 1 3

Prints only one record, from cylinder 0, track 1, record 3.

PRINT 1 10 3 TO 1 15 4

Prints all records starting with cylinder 1, track 10, record 3, and ending with cylinder 1, track 15, record 4.

The example in Figure 6 shows the information displayed at the console (TYPE function) or system printer (PRINT function) by the DDR program. The listing is annotated to describe some of the data fields.

Responses

ENTER CYLINDER EXTENTS
ENTER:

This message is received only if you are entering input from your terminal.

END OF VOLUME CYL xxx HD xx, MOUNT NEXT TAPE

DDR continues processing, after the mounting of the next tape reel.

RESTORING volser

where:

volser is the volume serial number of the disk dumped. The RESTORE operation has begun.

COPYING volser

where:

volser is the volume serial number described by the input unit. The COPY operation has begun.

DUMPING volser

where:

volser is the volume serial number described by the input unit. The dumping operation has begun.

PRINTING volser

where:

volser is the volume serial number described by the input unit. The PRINT operation has begun.

END OF DUMP

The DUMP operation has ended.

END OF RESTORE

The RESTORE operation has ended.

END OF COPY

The COPY operation has ended.

END OF PRINT

The PRINT operation has ended.

END OF JOB

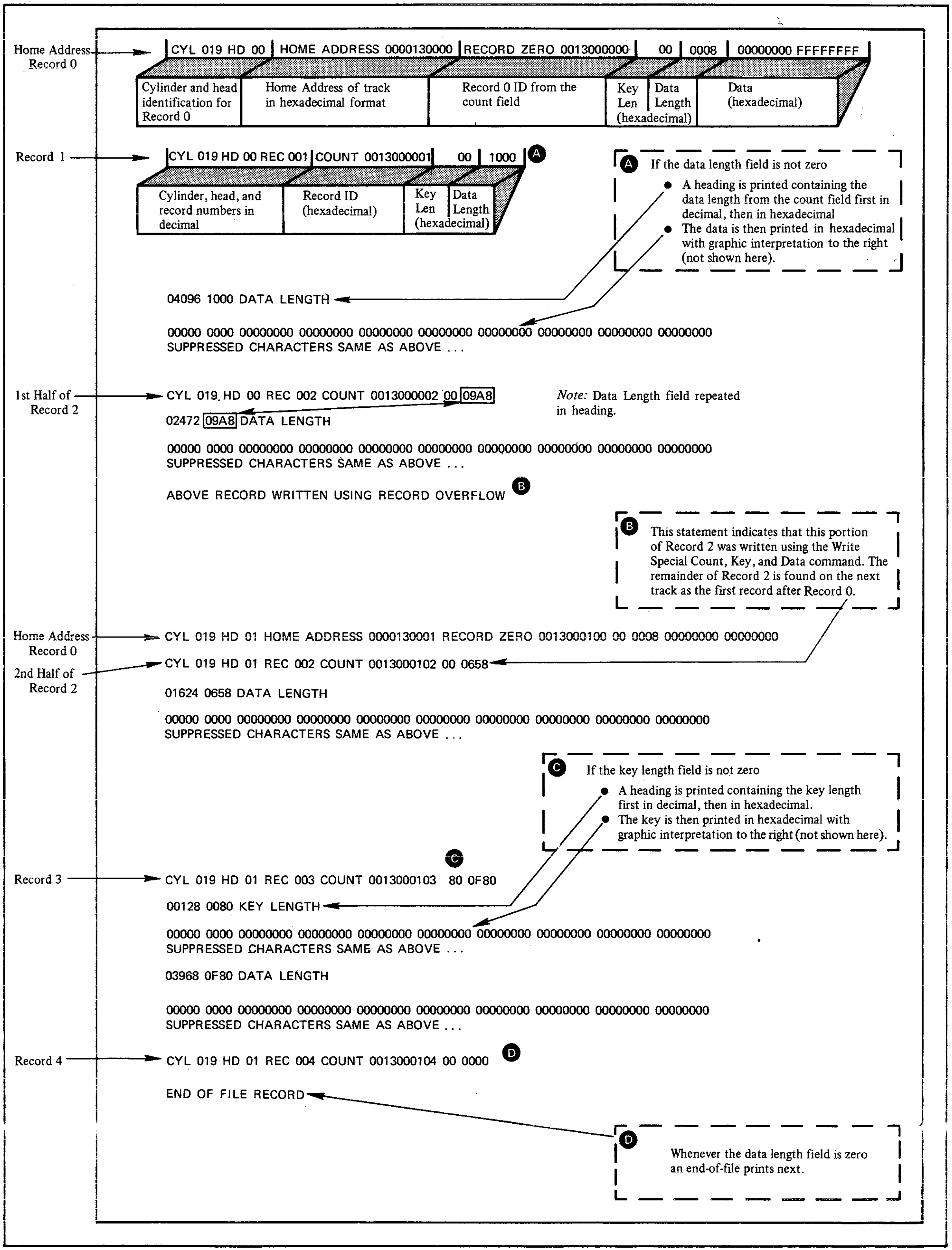
All specified operations have completed.

ENTER:

Prompts input from the terminal. A null line (Press the Enter key or equivalent) causes control to return to CMS, if the virtual machine is in the CMS environment.

In addition to the above responses, other informational messages that call for a response by the operator to continue, terminate, or reinitiate the current operation are described in *VM/370 System Messages*.

For return codes associated with DDR messages, see the *VM/370 CMS Command and Macro Reference*.



```
CYL 019 HD 00 HOME ADDRESS 0000130000 RECORD ZERO 0013000000  00  0008  00000000 FFFFFFFF

CYL 019 HD 00 REC 001 COUNT 0013000001  00  1000

04096 1000 DATA LENGTH

00000 0000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
SUPPRESSED CHARACTERS SAME AS ABOVE ...

CYL 019 HD 00 REC 002 COUNT 0013000002 00 09A8
02472 09A8 DATA LENGTH

00000 0000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
SUPPRESSED CHARACTERS SAME AS ABOVE ...

ABOVE RECORD WRITTEN USING RECORD OVERFLOW

CYL 019 HD 01 HOME ADDRESS 0000130001 RECORD ZERO 0013000100 00 0008 00000000 00000000
CYL 019 HD 01 REC 002 COUNT 0013000102 00 0658

01624 0658 DATA LENGTH

00000 0000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
SUPPRESSED CHARACTERS SAME AS ABOVE ...

CYL 019 HD 01 REC 003 COUNT 0013000103  80 0F80

00128 0080 KEY LENGTH

00000 0000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
SUPPRESSED CHARACTERS SAME AS ABOVE ...

03968 0F80 DATA LENGTH

00000 0000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
SUPPRESSED CHARACTERS SAME AS ABOVE ...

CYL 019 HD 01 REC 004 COUNT 0013000104 00 0000

END OF FILE RECORD
```

A If the data length field is not zero

- A heading is printed containing the data length from the count field first in decimal, then in hexadecimal
- The data is then printed in hexadecimal with graphic interpretation to the right (not shown here).

Note: Data Length field repeated in heading.

B This statement indicates that this portion of Record 2 was written using the Write Special Count, Key, and Data command. The remainder of Record 2 is found on the next track as the first record after Record 0.

C If the key length field is not zero

- A heading is printed containing the key length first in decimal, then in hexadecimal.
- The key is then printed in hexadecimal with graphic interpretation to the right (not shown here).

D Whenever the data length field is zero an end-of-file prints next.

Figure 6. Annotated Sample of Output from the TYPE and PRINT Functions of the DDR Program

Genimage and Imagelib Service Routines

VM/370 uses the OS utility program IEBIMAGE to enable the installaticn to change the character arrangement tables, forms control buffer, copy modification and graphic tables that are available to the users. The CMS system communicates with this utility through the use of two CMS commands:

GENIMAGE presents the input control file to the IEBIMAGE utility program. GENIMAGE creates the text files on a CMS disk. Text files are the images used by the 3800 printer. The text files represent new images or replacements to existing images. The format of the GENIMAGE control statement is:

```
┌──────────────────────────────────────────────────────────────────────────────┐
│            │ ┌     ┐  ┌    ┐  ┌  ┐  ┌         ┐  ┌        ┐  ┌   ┐            │
│  GENIMAGE  │ │fn   │  │ft  │  │fm│  │sfn      │  │sft     │  │sfm│            │
│            │ │SYSIN│  │FILE│  │* │  │SYSPRINT │  │LISTING │  │A1 │            │
│            │ └     ┘  └    ┘  └  ┘  └         ┘  └        ┘  └   ┘            │
└──────────────────────────────────────────────────────────────────────────────┘
```

__where__:

fn specifies the filename of the input control file. The name may be up to eight alphameric characters in length. If not specified, the default filename is SYSIN.

ft specifies the filetype of the control file. Filetype may be up to eight alphameric characters in length. If not specified, the default filetype is FILE.

fm specifies the filemode of the control file. Filemode may be up to two alphameric characters in length. If not specified, the default filemode is *.

sfn specifies the filename into which the messages listing is placed. If not specified, the default file name is SYSPRINT.

sft specifies the filetype for the messages listing. If not specified, the default file type is LISTING.

sfm specifies the filemode for the messages listing. If not specified, the default file mode is A1.

IMAGELIB reads the control file containing the list of text images. The files are then loaded into the specified named system. The format of the IMAGELIB control statement is:

```
┌──────────────────────────────────────────────────────┐
│               │                                      │
│   IMAGELIB    │      namedsys                        │
│               │                                      │
└──────────────────────────────────────────────────────┘
```

__where__:

namedsys specifies the named system that is being created or replaced.

Responses:

The programs generate messages that indicate successful completion or errors. Completion messages and codes are returned to the user in the file indicated by the sfn and sft operands of the GENIMAGE control | statement.

Note:

Further discussion of the IEBIMAGE utility program is contained in the OS/VS IBM 3800 Printing Subsystem Programmer's Guide.

Formatting Volumes--General Information

Disk Initialization

Disk initialization and alternate track assignment should be performed by the IBCDASDI utility program on all direct access volumes used in a VM/370 installation, but with the following exceptions:

- On 2314 and 2319 disks only if there are minidisks, then the individual minidisks should be initialized, rather than initializing the real disk.
- The 2305 drum should not have alternate tracks assigned. Instead, defective tracks should be rewired to alternate tracks by the CE. The rewiring makes the defective track transparent to all programming. (CP itself has no recovery procedures for "software assigned" alternate tracks on the 2305; however, an operating system running in one of its virtual machines might make use of them.)

Disk Formatting

All direct access volumes used by the VM/370 system (for paging, spooling, system residence, directory, or temporary disk allocation) must be properly labeled, formatted, and allocated. The CP Format/Allocate service program (DMKFMT module) prepares disks for use by CP. A CMS Format program is also available and must be used to format CMS and RSCS disks.

All direct access volumes (including both real disks and minidisks) to be used by VSAM under CMS, OS, or DOS, must be formatted by the IBCDASDI utility.

If certain information in the OS Format 4 label on track 0 cylinder 0 is destroyed, no additional alternate tracks can be assigned by IBCDASDI until the volume is reformatted by IBCDASDI. When either IBCDASDI or the CP Format/Allocate program (DMKFMT) is used to format a volume, this information in the OS Format 4 label is preserved. But, when the CMS Format Program formats the volume (or when it formats a minidisk whose origin is cylinder 0 of the volume) the OS Format 4 label is destroyed and no further alternate tracks can be assigned until IBCDASDI is used to reformat the volume.

Format/Allocate Services Program (DMKFMT)

The Format/Allocate service program formats, allocates, and labels direct access volumes for paging, spooling and CP file residence. This service program is executed as part of CP system generation procedures and may also be executed as a stand-alone program to:

- Format direct access volumes for CP use
- Allocate specific disk areas to particular functions or to CP use
- Write six-character volume serial number labels

Note: The Format/Allocate program should be used with care since it destroys existing data (if any). Also, user minidisks and temporary minidisks must not begin on real cylinder zero of CP-owned volumes, because information critical to CP is stored in that cylinder.

An object deck version of the CP Format/Allocate service program is a standalone program and can be loaded from a virtual or real card reader into a virtual or a real machine. (If run in a virtual machine, the virtual machine must have write access to the volume being formatted.) The program accepts control statements from the operator's system console (commands) or from the IPL device (card reader).

Note: Messages DMKFMT736E and DMKFMT735E may be issued if an available path to the device cannot be found after an appropriate number of retries. High activity may cause this situation.

Cylinders used by CP for paging, spooling, and so on, must be preformatted with fixed length unblocked records of 4096 bytes.

Device capacity when formatted for CP use is:

2314/2319	32 records/cylinder	(8 records/5 tracks)
3330	57 records/cylinder	(3 records/track)
2305	24 records/cylinder	(3 records/track)
3340	24 records/cylinder	(2 records/track)
3350	120 records/cylinder	(4 records/track)

The format operation writes 4096-byte blocks on all cylinders being formatted. The service program does write-checking to verify that parts of the track are not defective. A count is maintained of pages with read check errors detected during the format operation. At the completion of the format operation, the count of the pages with read check errors is printed.

Format for All Cylinders Except Cylinder Zero

For example, the 3330 track format for all formatted cylinders except cylinder 0 is shown in Figure 7.

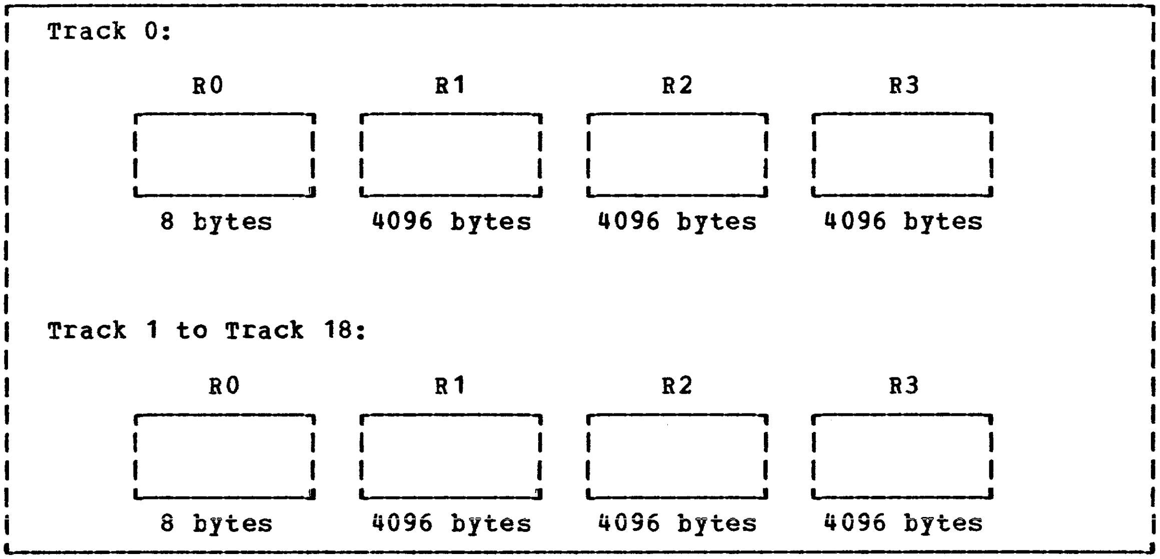


Figure 7. Format of 3330 Cylinders for Use by CP

Format for Cylinder Zero

All volumes containing space for CP use (paging, spooling, and so on) must have a properly formatted cylinder 0. The only service program that can do this is the Format/Allocate program (DMKFMT).

Cylinder 0 is formatted like other cylinders except that the space associated with the first three 4096-byte blocks is reserved for system use. This area is then formatted as illustrated in Figure 8.

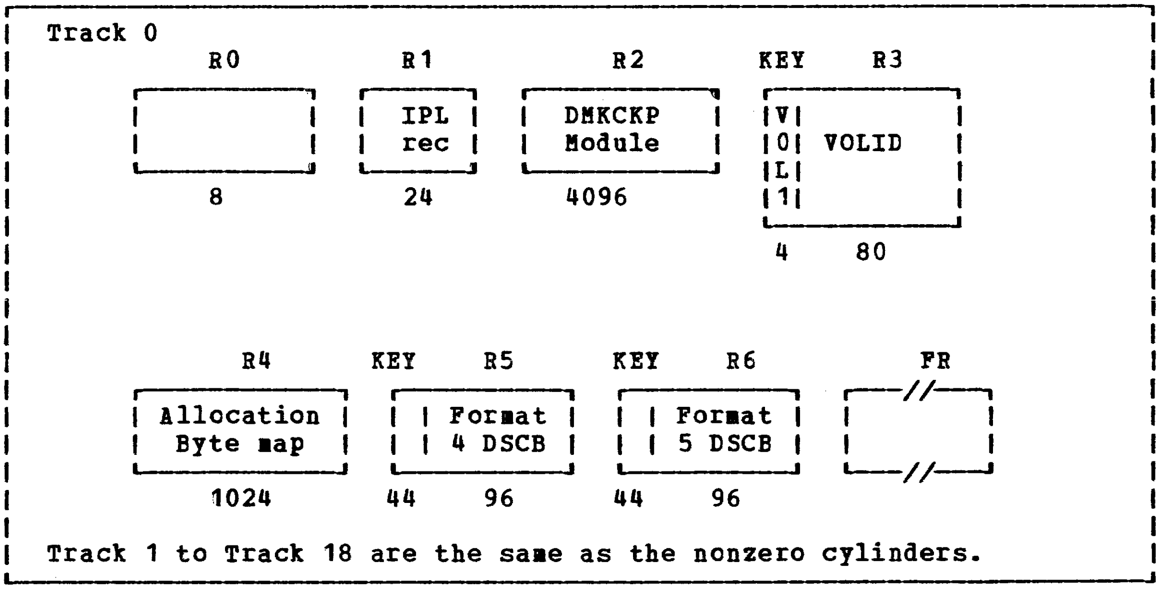


Figure 8. 3330, 3340, or 3350 Cylinder 0 Format

The contents of each record in cylinder 0 track 0 are as follows:

R0 Nothing.

R1 IPL record -- Puts the system into wait state if storage volume is loaded before CP nucleus is built.

R2 Checkpoint record -- Used by CP to save and retrieve information for a warm start.

R3 Volume label -- Same as OS VOL1 label. On CP system residence volume, area in data record marks the beginning of the system directory. A label is automatically written when cylinder 0 is formatted. The owner field of the label record contains "VM/370" if there is allocation data present in R4.

R4 Allocation Byte Map -- Each byte identifies a cylinder and specifies its usage (paging, spooling, directory, and so on). This map is filled in by the ALLOCATE function of the DMKFMT service program.

R5 Format 4 OS DSCB type label -- For compatibility with OS. Also, the IBCDASDI program uses this label to keep a record of how many alternate tracks remain available for assignment on this disk. The Format/Allocate program will preserve this information by first reading it from any existing Format 4 label, and then writing it back in the new label.

R6 Format 5 OS DSCB type label -- For compatibility with OS. Label indicates to OS that no space is available on this volume.

FR Is one or more filler records.

Format/Allocate program control statements may be supplied in card form via a card reader, or may be entered at the system console. All error messages regarding improper specification of control statements are displayed at the console.

FORMAT/ALLOCATE PROGRAM CARD INPUT

Punch control statements for card input start in column 1, and each field is separated from the adjacent field by a comma. Two commas in a row cause the insertion of a default value. Three commas in a row cause the insertion of two default values.

Note: The only default values permitted are those that define the starting and ending cylinders. The defaults are the first and last cylinders of the volume, respectively.

Comments must be preceded by at least three blanks.

The control card entries for the Format/Allocate program must be in the following order:

- Format function:

```
FORMAT,devadr,devtype,volser,startcyladr,endcyladr
```

- Allocate function:

```
ALLOCATE,devadr,devtype,volser
TEMP,startcyladr,endcyladr
PERM,startcyladr,endcyladr
TDSK,startcyladr,endcyladr
DRCT,startcyladr,endcyladr
END
```

- Label functions:

```
FORMAT,devadr,devtype,volser,LABEL
```

FORMAT, ALLOCATE, and LABEL are Format/Allocate program control words and may be abbreviated to one letter.

FORMAT Control Statement

The format of the FORMAT control statement is:

```
FORMAT,devadr,devtype,volser,startcyladr,endcyladr
```

where:

devadr is a three-digit hexadecimal number that identifies the address of the device that the Format/Allocate program is to act upon. Valid device addresses under CMS are X'001' to X'5FF' for ECMODE OFF and X'001' to X'FFF' for ECMODE ON.

devtype is a four-to-seven character field that defines an approved device for the Format/Allocate program. Approved device types are 2314, 2319, 3330, 3330-11, 3340-35, 3340-70, 3350, 2305-1, and 2305-2. Specify 3333 as 3330, and 3340-70F or 3344 as 3340-70. Specify a 3350 device in 3330-1 compatibility mode as a 3330, and a 3350 in 3330-11 compatibility mode as a 3330-11.

volser is a one-to-six character field that represents the volume serial number of the volume you are formatting.

startcyladr is the starting cylinder address on the DASD on which the format function is to be performed. The start cylinder address is entered as decimal digits.

endcyladr is the last cylinder address on the DASD on which the format function is to be performed. The end cylinder address is entered as decimal digits.

Note: FORMAT is a control word and maybe abbreviated to F.

ALLOCATE Control Statements

The formats of the ALLOCATE control statements are:

```
ALLOCATE,devadr,devtype,volser
TEMP,startcyladr,endcyladr
PERM,startcyladr,endcyladr
TDSK,startcyladr,endcyladr
DRCT,startcyladr,endcyladr
END
```

where:

devadr — is a three-digit hexadecimal number that identifies the address of the device that the Format/Allocate program is to act upon. Valid device addresses under CMS are X'001' to X'5FF' for ECMODE OFF and X'001' to X'FFF' for ECMODE ON.

devtype — is a four-or-seven character field that defines an approved device for the Format/Allocate program. Approved device types are 2314, 2319, 3330, 3330-11, 3340-35, 3340-70, 3350, 2305-1, and 2305-2. Specify a 3350 device in 3330-1 compatibility mode as a 3330 and a 3350 in 3330-11 compatibility mode as a 3330-11. Specify a 3333 as a 3330, and 3340-70F or 3344 as 3340-70.

volser — is a one-to-six character field that represents the volume serial number of the volume you are formatting.

startcyladr — is the starting cylinder address on the DASD on which the format function is to be performed. The start cylinder address is entered as decimal digits.

endcyladr — is the last cylinder address on the DASD on which the format function is to be performed. The end cylinder address is entered as decimal digits.

TEMP — indicates that the following operands identify temporary storage space reserved for spooling or paging activity.

PERM — defines an area that can contain the logout area, the CP nucleus, and space that is not used by the system but is available for use by virtual machine users (for example, for user minidisks).

TDSK — defines the pooled space available for virtual machine users after they have logged on the VM/370 system.

DRCT — indicates that the following cylinders are reserved for directory files.

ALLOCATE is a control word and may be abbreviated to its first letter, A.

TEMP, PERM, TDSK, and DRCT are all functions of ALLOCATE. These cards can follow the ALLOCATE control statement in any sequence. Each card in turn overlays the cylinder table, and any space not reallocated remains the same. If an ALLOCATE function overlays the previous cylinder allotment, then the previous cylinder space allotment is truncated to the beginning of the next cylinder allotment. For example:

Disk Storage Allocation	First Cylinder	Last Cylinder
1st Entry TEMP	000	202
2nd Entry PERM	010	050
3rd Entry TDISK	040	050
4th Entry DRCT	000	004
5th Entry END		

The result of this disk volume allocation is:

Disk Storage Allocation	First Cylinder	Last Cylinder
DRCT	000	004
TEMP	005	009
PERM	010	039
TDSK	040	050
TEMP	051	202

Once an ALLOCATE control statement is encountered, all cards following it until an END card is encountered are assumed to be part of a single allocation. The Format/Allocate service functions cannot be performed on another disk volume until the END card is encountered. Any area not allocated will default to "TEMP" space. (See Figure 10.) Note, however that if the allocation extent map does not cover the entire volume (as could be the case if the volume was copied from a smaller volume or minidisk) the remainder of the volume is allocated as PERM space.

Note: Reallocation of a directory cylinder containing an active VM/370 directory deallocates the directory to allow a new directory to be written on the same cylinder. After reallocation, the Directory program must be executed to reinitialize the directory pointer. If this is not done, when you initialize (via IPL), your program will abend (ABEND CPI002).

LABEL Control Statement

The format of the LABEL control statement is:

```
FORMAT,devadr,devtype,volser,LABEL
```

where:

devadr is a three-digit hexadecimal number that identifies the address of the device that the Format/Allocate program is to act upon. Valid device addresses under CMS are X'001' to X'5FF' for ECMODE OFF and X'001' to X'FFF' for ECMODE ON.

devtype is a four-to-seven character field that defines an approved device for the Format/Allocate program. Approved device types are 2314, 2319, 3330, 3330-11, 3340, 3340-35, 3340-70, 3350, 2305-1, and 2305-2. Specify a 3350 device in 3330-1 compatibility mode as a 3330, and a 3350 in 3330-11 compatibility mode as a 3330-11. Specify a 3333 as a 3330, and 3340-70F or 3344 as 3340-70.

volser   is a one-to-six character field that represents the volume serial number.

LABEL   is a keyword designating the label function of the Format/Allocate program.

Note: FORMAT and LABEL are control words and may be abbreviated to F and L, respectively.

Examples:

FORMAT:

```
FORMAT,232,3330,MYDISK,000,006
FORMAT,232,3330,MYDISK,,,
FORMAT,232,3330,MYDISK,,00
FORMAT,232,3330,MYDISK,001,,
```

ALLOCATE:

```
ALLOCATE,232,3330,MYDISK
TEMP,000,050
PERM,055,060
TDSK,100,108
DRCT,110,120
END
```

LABEL:

```
F,232,3330,MYDISK,label
```

FORMAT/ALLOCATE CONSOLE INPUT

The Format/Allocate program can be controlled by control statements entered into the real or virtual console instead of by a deck of cards containing control statements. If the program finds no control statements at the card reader, it issues a prompting message to the console. The proper response causes the prompting message for the next operand to appear until the Format, Allocate, or Label function is completely defined; then the Format/Allocate program is executed.

After execution, the prompting begins again until all DASD allocation requirements are fulfilled.

The sequence for console typewriter processing of the Format/Allocate program (after making the operator's console ready), is as follows:

1. Load the card reader with a loader, followed by the Format/Allocate deck.

2. IPL the card reader.

3. Respond to the first message displayed at the system console.

4. Respond to other messages.

Following are examples of Format/Allocate program execution under CP control. Figure 9 is an example of the label operation, Figure 10 is an example of the allocate operation, and Figure 11 is an example the allocate overlap operation. All responses are entered after the colon; after a function is complete, the program returns and issues again the 'ENTER "FORMAT" OR "ALLOCATE":' statement.

```
VM/370 FORMAT/ALLOCATE PROGRAM VERSION 3.0

ENTER "FORMAT" OR "ALLOCATE":f
FORMAT FUNCTION SELECTED
ENTER DEVICE ADDRESS (CUU):131
ENTER DEVICE TYPE: 2314
ENTER START CYLINDER (XXX) OR "LABEL":l
ENTER DEVICE LABEL:cpdsk2
```

Figure 9. Using the Format Program Label Function

```
ENTER "FORMAT" OR "ALLOCATE":a
ALLOCATE FUNCTION SELECTED
ENTER DEVICE ADDRESS (CUU):131
ENTER DEVICE TYPE: 2314
ENTER DEVICE LABEL:cpdsk2
ENTER ALLOCATION DATA FOR VOLUME CPDSK2
TYPE CYL CYL
.... ... ...
drct 000 001
perm 004 008
tdsk 100 150
end
ALLOCATION RESULTS
DRCT 000 001
TEMP 002 003
PERM 004 008
TEMP 009 099
TDSK 100 150
TEMP 151 202
DEVICE 131 VOLUME CPDSK2 ALLOCATION ENDED
```

Figure 10. Using the Format Program Allocate Function

```
ENTER "FORMAT" OR "ALLOCATE":a
ALLOCATE FUNCTION SELECTED
ENTER DEVICE ADDRESS (CUU):131
ENTER DEVICE TYPE: 2314
ENTER DEVICE LABEL: cpdsk2
ENTER ALLOCATION DATA FOR VOLUME CPDSK2
TYPE CYL CYL
.... ... ...
perm 004 004
temp 000 010
tdsk 000 010
perm 010 202
drct 000 004
end
ALLOCATION RESULTS
DRCT 000 004
TDSK 005 009
PERM 010 202
DEVICE 131 VOLUME CPDSK2 ALLOCATION ENDED
```

Figure 11. Using the Format Program Allocate Overlap Function

Note that before the ALLOCATE function was invoked, cylinder 0 was formatted and labeled CPDSK2. The area associated with the first three 4096-byte blocks on cylinder 0 are not used for spooling but contain system information (page allocation map, label, and so on).

These CP-formatted volumes can be made usable by CP in one of two ways:

1. They may be attached to the system by the VM/370 operator.
2. Their volume serial numbers may appear in the SYSOWN macro in the DMKSYS module. The CP system residence volume's serial number must appear in the SYSOWN macro.

IBCDASDI--General Information

VM/370 uses the OS utility program IBCDASDI for disk initialization, for disk formatting, and for assigning an alternate track. The execution of IBCDASDI is performed from the virtual card reader and the technique is described in "Invoking IBCDASDI" later in this section.

DISK INITIALIZATION - Corresponds roughly to what is referred to as "initialization with surface analysis" or "non-quick DASDI". It is invoked by a DADEF statement when its PASSES operand is specified or defaulted as a value greater than 0 (and for 2314, 2319, and 2305 the FLAGTEST=NO operand must also be specified). Each track is tested and alternate tracks will be assigned to tracks that prove defective. Pre-existing flags and alternate assignments are ignored. Record zero is rewritten on all tracks. For some device types, home addresses are also rewritten on all tracks. Disk initialization can be performed on 2305, 3330, 3340, and 3350 real (dedicated) disks and on 2314 and 2319 minidisks as well as real disks. Initialization will generally be required when a disk (or a 2314 or 2319 minidisk) is in an initial or unknown state. Note: whenever disk initialization is done, disk formatting is also done.

| Note: Disk formatting will not reset (clear) the count field.

DISK FORMATTING - Corresponds roughly to what is referred to as "initialization without surface analysis" or "quick DASDI". It is invoked by a DADEF statement with PASSES=0,BYPASS=YES specified. Formatting consists of writing IPL records 1 and 2 standard OS volume labels on track 0 cylinder 0 and of writing an OS volume table of contents (VTOC). Since a disk's OS Format 4 label always contains a record of how many alternate tracks remain available for assignment, this information must be computed before the Format 4 label can be written. To compute this, IBCDASDI reads the "alternate tracks" cylinders. Disk formatting can be performed on any device, real or minidisk. Disk formatting will generally be required when standard labels must be created.

ASSIGNING AN ALTERNATE TRACK - Invoked with a GETALT statement. This is an explicit request for an alternate to be assigned to the particular track that is specified. (Note that disk initialization may also assign alternate tracks.) Only the specified track is affected, so label and VTOC data is generally preserved. Alternate track assignment can be performed on 3330, 3340, and 3350 real (dedicated) disks and on 2314 and

2319 minidisks as well as real disks.[1] The last cylinder in any 2314 or 2319 minidisk initialized by IBCDASDI is reserved and used as the "alternate tracks" cylinder. Assigning an alternate track explicitly will generally be required when a track becomes defective subsequent to disk initialization or when an intermittently bad track is missed during disk initialization.

IBCDASDI Restrictions

The IBCDASDI program cannot check to see if the 3330, 3340, or 3350 space to be initialized was previously formatted.

If you format only five cylinders of a 5-cylinder virtual disk of a 3330, 3340, or 3350 but specify 20 cylinders to be initialized (CYLNO=20), the IBCDASDI program does not initialize all 20 cylinders but merely updates the format 5 DSCB in the VTOC to indicate available space, without checking for the existence of 20 cylinders and without issuing an error message. Later, any attempt to use the sixth through the twentieth cylinder causes a Seek Check and the channel program abnormally terminates.

When assigning alternate tracks explicitly, IBCDASDI must find a valid OS Format 4 label on the disk. If this label has been destroyed (by CMS Format, for example), or if the label does not apply to the present disk in its present state (for example, a no longer applicable OS Format 4 label might have been restored to the disk by the DDR utility), then IBCDASDI must be used to perform the disk formatting function before alternate tracks can be assigned. The recommended procedure in this case is to:

- Dump the entire disk to tape using the DDR utility
- Format the disk using IBCDASDI
- Assign the additional alternate tracks using IBCDASDI
- Restore the data to the disk from the tape using the DDR utility

[1] On the 2305 drum, defective tracks should be rewired to alternate tracks by the CE rather than being flagged and assigned an alternate in the usual manner. In this case the fact that a track is defective and has an alternate wired in is transparent to all programming. (CP itself has no recovery procedures for "software assigned" alternate tracks on the 2305; however, an operating system running in one of its virtual machines might make use of them.)

Initialization with Surface Analysis (Not applicable to 3330/3340/3350)

The IBCDASDI program:

- Checks for tracks that were previously designated as defective (flagged) and have had alternates assigned. The program automatically assigns alternate tracks for 2314/2319 disk devices. This test must be suppressed when a disk is being initialized with surface analysis for the first time. This test must not be suppressed when a disk is initialized without surface analysis.
- Performs a surface analysis of each track of a 2314 or 2319 and automatically assigns alternates (for 2314/2319s), if necessary. Tracks that are available for use as alternates are checked first.
- Writes a track descriptor record (record 0), and erases the remainder of each track. When initializing a disk with surface analysis, IBCDASDI also writes a standard home address.
- Writes IPL records on track 0 (records 1 and 2).
- Writes volume label on track 0 (record 3) and provides space for additional records, if requested.
- Constructs and writes a volume table of contents (VTOC).
- Writes IPL program, if requested, on track 0 (2314, 2305, 2319, or track 1 (2311).

Initialization without Surface Analysis

The IBCDASDI program:

- Checks for tracks that were previously designated as defective (flagged) and have had alternates assigned. The program automatically assigns alternates (2314/2319 disk devices only). This test must not be suppressed.
- For 3350 devices, HA/RO on all tracks is rewritten (to facilitate changing from ' 3330 compatibility mode to 3350 native mode and vice versa). Tracks that are flagged defective are surface analyzed and reclaimed if no errors are detected. The BYPASS and FLAGTEST options are ignored. The PASSES=0 option causes "QUICK DASDI" to be performed.
- Writes a standard home address, a track descriptor record (record 0), and erases the remainder of each track.
- Writes IPL records on track 0 (records 1 and 2).
- Writes volume label on track 0 (record 3) and provides space for additional records, if requested.
- Constructs and writes a volume table of contents (VTOC).
- Writes IPL program, if requested, on track 0 (2314, 2305, 3330, 3340, or 3350 devices) or track 1 (2311 disks).

Note: The IBCDASDI program can only assign alternate tracks for real 3330/3340 volumes when they are specified by the GETALT statement. Even with the GETALT statement, the IBCDASDI program cannot assign alternate tracks for a 3330/3340/3350 minidisk because CP prevents a virtual machine from writing a home address record on these devices (effectively treating the write as a NO-OP) unless the device is dedicated to that virtual machine. Defective tracks are flagged and alternate tracks are assigned when the 3330/3340/3350 storage volumes are initialized at the factory. An IBCDASDI job that initializes a 3330 performs the Quick DASDI function, which reads alternate tracks, decrementing by 1 the total number of alternates whenever an alternate is found defective or assigned, writes a volume label and VTOC, and writes an IPLTEXT if requested. No surface analysis is performed and no home address or record 0 is written on the primary tracks. The BYPASS and FLAGTEST options of the DADEF statement are ignored.

Disk packs for the 3340 are factory-shipped without flagged tracks and alternate track assignments. IBCDASDI's "Quick DASDI" detects 3340 customer-generated alternate track assignments.

The IBCDASDI program cannot check to see if the 3330, 3340, or 3350 space to be initiated was previously formatted.

| All utility control statements and operands must be preceded and
| followed by one or more blanks.

The IBCDASDI program requires control statements as input. Data on a control statement must not extend beyond column 71 except when a continuation of the statement is required. Then the statement must be broken off after a comma and a non-blank character must be placed in column 72. Then the statement can be continued on the next line, beginning in column 16 (and beginning <u>only</u> in column 16). The statements must appear in the following sequence:

1. JOB Statement -- indicates the beginning of the IBCDASDI job.
2. MSG Statement -- defines the output device for operator messages.
3. DADEF Statement -- defines the DASD device to be initialized.
4. VLD Statement -- labels the volume and allocates space for additional labels.
5. VTOCD Statement -- controls the location of the volume-table-of-contents (VTOC).
6. IPLTXT Statement (optional) -- separates service program control statements from IPL text statements.
7. END Statement -- ends an IBCDASDI job or IPL text.
8. LASTCARD Statement (optional) -- ends series of stacked IBCDASDI jobs.

Initializing a 3350 (3330 compatability mode to native mode)

To initialize a 3350 in native mode after conversion from 3330 or 3340 compatability mode, the device must be attached (dedicated) to a virtual machine and either IEHDASDP or IBCDASDI must be run. This is necessary to allow alternate cylinders to be formatted and alternate tracks assigned.

JOB Statement

The JOB statement indicates the beginning of a job. The format of the JOB statement is:

[name]	JOB	[user information]

MSG Statement

The MSG statement defines an output device for operator messages. It follows the JOB statement and precedes any function-defining statements that are associated with the IBCDASDI program. The format of the MSG statement is:

[name]	MSG	TODEV=xxxx,TOADDR=cuu

where:

TODEV=xxxx is the device type of the message output device (1403, 1052, 2400, 3400, 3210, or 3211).

TOADDR=cuu is the channel number (c) and unit number (uu) of the message output device.

DADEF Statement

The DADEF statement defines the direct access vclume to be initialized. The format of the DADEF statement is:

```
┌──────────────────────────────────────────────────────────────────┐
|[name]      |DADEF      |TODEV=xxxxxx                            ¹ |
|            |           |,TOADDR=cuu                               |
|            |           |[,IPL=YES]                                |
|            |           |{,VOLID=serial  }                         |
|            |           |{,VOLID=SCRATCH }                         |
|            |           |[,MODEL=n]                                |
|            |           |[,CYLNO=nnn]                              |
|            |           |------------------------------------------|
|            |           |[,FLAGTEST=NO]                          ² |
|            |           |[,PASSES=n]                               |
|            |           |------------------------------------------|
|            |           |[,BYPASS=YES]                           ³ |
|            |           |                                          |
|------------------------------------------------------------------|
|¹Applies to initialization with or without surface analysis.      |
|²Applies to initialization with surface analysis.                 |
|³Applies to initialization without surface analysis.              |
└──────────────────────────────────────────────────────────────────┘
```

where:

TODEV=xxxxxx
is a 4- to 6-character device type of the direct access device. Specify either:

2305	
2314	
3340	for 3340 and 3344
3330	for 3330 Models 1 or 2 and 3350 in 3330-1 compatibility mode
3330-1	for 3330 Model 11 and 3350 in 3330-11 compatibility mode
3350	in native mode

TOADDR=cuu
is the channel (c) and unit address (uu) of the device. Specify the real address of the device if you are executing IBCDASDI on the real machine; specify the virtual address of the device if you are executing IBCDASDI on the virtual machine.

IPL=YES
writes an IPL program on the minidisk. An IPL initialization program must be written on a device to be used for system residence. If IPL is omitted, no IPL program is written.

Note: If you specify IPL=YES, you must also include an IPLTXT statement.

VOLID=serial
is the volume serial number of the minidisk to be initialized. If "serial" matches the volume serial number found on the minidisk to be initialized, the operation proceeds. If it does not match, the operator is notified.

VOLID=SCRATCH
specifies that no volume serial number check is to be made.

MODEL=n is a decimal model number (1 or 2). This operand is only for the 2305 and corresponds to the 2305-1 and 2305-2, respectively.

CYLNO=nnn

is a decimal number that specifies the number of cylinders to be formatted. If the CYLNO parameter is omitted, IBCDASDI intializes the entire real volume specified.

Note: If more cylinders are specified than are available on the minidisk, the message "Command Reject" is returned.

FLAGTEST=NO

(applies to surface analysis) specifies that the program is not to check for previously flagged tracks before surface analysis is attempted on this device. (FLAGTEST=NO applies only to 2314 and 2319 devices, and should be specified when the disk recording surface is initialized for the first time.)

Note: Because no check is made for previously flagged tracks on drum volumes, FLAGTEST=NO is not coded when these devices are initialized.

PASSES=n specifies the number of passes per track to be made in checking for defective tracks. PASSES is valid when surface analysis is to be performed or when a Quick DASDI is to be performed on a 3330, 3330-1, 3340, or 3350 volume. The value n can be 0 through 255. The 0 value indicates that a "Quick DASDI" is to be performed on a 3330, 3330-1, 3340, or 3350 volume. For a 3330, 3330-1, 3340, or 3350 volume, a value other than zero causes record 0 to be written on each track. PASSES=0 should be specified for 3330, 3340, and 3350 (full disk or minidisk) except when deliberately trying to free alternate areas assigned by GETALT in the prime data area, at which time PASSES=1 would have to be specified. No check is made for defective tracks on a 3330, 3330-1, 3340, or 3350. A value of 1 through 255 indicates the number of passes to be made per track for volumes other than a 3330, 3330-1, 3340, or 3350 volume. If PASSES is omitted, one pass is made per track.

BYPASS=YES bypasses the rechecking of any tracks that are already flagged as defective. If BYPASS is omitted, flagged tracks are rechecked and those found defective are automatically assigned alternates. For 3330 or 3350 devices, this operand is ignored and BYPASS=YES is assumed.

VLD Statement

The VLD statement labels the volume. The format of the VLD statement is:

```
┌──────────────────────────────────────────────────────────────────────┐
|[name]       | VLD     | NEWVOLID=serial                              |
|             |         | ┌           ┐                                |
|             |         | |,VOLPASS=0|                                 |
|             |         | |,VOLPASS=1|                                 |
|             |         | └           ┘                                |
|             |         | [,OWNERID=xxxxxxxxxx]                        |
|             |         | [,ADDLABEL=n]                                |
└──────────────────────────────────────────────────────────────────────┘
```

where:

NEWVOLID=serial
is a one-to-six character volume serial number.

VOLPASS=1 sets the volume security bit to 1.

VOLPASS=0 sets the volume security bit to 0.

If VOLPASS is omitted, the volume security bit is set to 0.

OWNERID=xxxxxxxxxx
is a one-to-ten character field that identifies the owner of the volume.

If OWNERID is omitted, no identification is given.

Note: The ownerid CP370 is reserved for use by DMKFMT and cannot be specified.

ADDLABEL=n
is a number between one and seven that indicates the total number of additional labels for which space is to be allocated.

If ADDLABEL is omitted, 0 is assumed.

VTOCD Statement

The VTOCD statement contains information for controlling the location of the volume table of contents. The format of the VTOCD statement is:

```
┌──────────────────────────────────────────────────────────────────────┐
|[name]      |VTOCD     |STRTADR=nnnnn                                 |
|            |          |,EXTENT=nnnn                                  |
└──────────────────────────────────────────────────────────────────────┘
```

where:

STRTADR=nnnnn
is the one to five byte track address, relative to the beginning of the minidisk, at which the volume table of contents is to begin. The VTOC cannot occupy cylinder 0, track 00, or any alternate track.

EXTENT=nnnn

is the length of the volume table of contents in tracks. The number (decimal) of entries per track for each type of device is given below.

Device	VTOC Entries/Track
2301	63
2314/2319	25
2311	16
2305-1	18
2305-2	34
3330-1,2	39
3330-11	39
3340	22
3350 (in native mode)	46

IPLTXT Statement

The IPLTXT statement separates service program control statements from IPL program text statements. It is required only when IPL=YES is coded on the DADEF statement and IPL text is included. The statement consists of the characters IPLTXT, followed by blanks.

When IPL text is included, the END statement must follow it and END must start in column 2.

END Statement

The END statement denotes the end of the job. It appears after the last function-defining statement. The format of the END statement is:

[name]	END	[user information]

LASTCARD Statement

The LASTCARD statement is required only when a IBCDASDI job or a series of stacked IBCDASDI jobs is followed by other statements on the control statement input device. The LASTCARD statement must follow the last END statement applying to an IBCDASDI job. It consists of the operand LASTCARD, followed by blanks.

ASSIGNING AN ALTERNATE TRACK

IBCDASDI: (1) analyzes a track and, if necessary, assigns an alternate or (2) bypasses testing, and assigns an alternate. You must specify the tracks for which you wish alternates with a GETALT statement.

Assigning an Alternate Track (with Testing): An alternate track (if available) is assigned for a track specified for testing and found defective. If the defective track has had an alternate previously assigned, a new alternate track is assigned. If the defective track is an unassigned alternate track, it is flagged to prevent its future use and another alternate track is selected. The alternate track address is made known to the operator.

If a track is tested and found to be "not defective," no alternate is assigned. The operator is notified by a message.

Assigning an Alternate Track (without Testing): The program's defective track checking feature can be bypassed, and an alternate track can be assigned for any track, whether it is permanently defective or not. If the specified track is an alternate, a new alternate track is assigned. If the specified track is an unassigned alternate, it is flagged to prevent its future use.

Note: For 3330, 3340, and 3350 minidisks you must assign the alternate on the real disk. Any references thereafter to that track on the minidisk are referred to the alternate track.

GETALT Statement

Any number of alternate tracks can be assigned in a single job by including one GETALT statement for each track. The GETALT statement can follow the MSG statement. The format of the GETALT statement is:

```
┌──────────────────────────────────────────────────────────────────────┐
|[name]      |GETALT   |TODEV=xxxx                                      |
|            |         |,TOADDR=cuu                                     |
|            |         |,TRACK=ccccchhhh                                |
|            |         |,VOLID=serial                                   |
|            |         |[,FLAGTEST=NO]                                  |
|            |         |[,PASSES=n]                                     |
|            |         |[,BYPASS=YES]                                   |
|            |         |[,MODEL=n]                                      |
└──────────────────────────────────────────────────────────────────────┘
```

where:

TODEV=xxxx
is the device type of the direct access device.

TOADDR=cuu
is the direct channel number (c) and unit number (uu) of the direct access device.

TRACK=ccccchhhh
is the address of the track for which an alternate is requested, where cccc is the cylinder number and hhhh is the head number. These are hexadecimal numbers.

VOLID=serial
is the volume serial number of the disk or minidisk to which an alternate track is to be assigned. If serial matches the volume serial number found on this disk or minidisk, the alternate track assignment proceeds. If it does not match, the operator is notified.

FLAGTEST=NO
(used when testing before assigning an alternate) specifies that the program not check for a previously flagged track before a surface analysis is attempted on this track (disk storage devices only).

PASSES=n (used when testing before assigning an alternate) specifies that the program's defective track checking feature is to make n number of passes (from 1 to 225) when performing a surface analysis. If PASSES is omitted, one pass is made on this track.

BYPASS=YES
is that the program's defective track checking feature is to be bypassed. If BYPASS is omitted, the program assigns an alternate only if it finds that the specified track is defective.

MODEL=n is a decimal model number (1 or 2). This operand is only for the 2305 and corresponds to the 2305-1 and 2305-2, respectively.

Note: A list of defective tracks (if any) is provided with new IBM disk storage volumes. Refer to this list when using the IBCDASDI program for the first time. After initialization, include the GETALT statement in an IBCDASDI job to assign an alternate track for each track on the list. Subsequent IBCDASDI jobs "remember" those defective tracks, unless the FLAGTEST=NO option is specified for those jobs.

Example 1

The example below shows the control statements that might be prepared to initialize a dedicated 3340 disk that is to be initialized by running IBCDASDI in a virtual machine. The disk is to be labeled VMRES and it is mounted on unit address 240.

```
INIT240  JOB
         MSG    TODEV=1052,TOADDR=009
         DADEF  TODEV=3340,TOADDR=240,VOLID=SCRATCH,      C
                FLAGTEST=NO,BYPASS=NO
         VLD    NEWVOLID=VMRES
         VTOCD  STRTADR=1,EXTENT=1
         END
```

Since the PASSES operand was not specified, it defaults to PASSES=1 and a non-Quick DASDI function is performed. All tracks, including any that were previously flagged as defective, are tested. Flagged tracks that test okay are recovered (not flagged). Any tracks that do not pass the test are flagged and an alternate track is assigned.

Example 2

The example below shows the control statements that might be used to explicitly assign alternate tracks to two tracks on the same disk shown in Example 1. It is assumed that these two tracks were found to be defective when the disk was put into use, subsequent to its initialization in Example 1. Once again the disk must be dedicated to the virtual machine in which IBCDASDI will be run.

```
MOREALTS JOB
         MSG     TODEV=1052,TOADDR=009
         GETALT  TODEV=3340,TOADDR=240,VOLID=VMRES,        C
                 BYPASS=YES,TRACK=00FB0003
         GETALT  TODEV=3340,TOADDR=240,VOLID=VMRES,        C
                 BYPASS=YES,TRACK=011C0005
         END
```

The TRACK operands specify the hexadecimal CCHH addresses of the two tracks that get flagged as defective.

Example 3

The example below shows the control statements that might be prepared to initialize the 2314 minidisk LIBRES for OS data set residence. Note that the label of the real volume, CPVOL1, cannot be used as the VOLID operand.

```
JOB
MSG     TODEV=1403,TOADDR=00E
DADEF   TODEV=2314,TOADDR=231,VOLID=SCRATCH,CYLNO=50
VLD     NEWVOLID=LIBRES,OWNERID=OPERATIONS
VTOCD   STRTADR=10,EXTENT=5
END
```

The desired size of the minidisk is specified through the CYLNO operand. The value specified includes one cylinder reserved for alternate track assignment; a user assigned n cylinders has n-1 of these initialized for his use, and the nth cylinder used for any alternate track assignment. The minimum size of a minidisk can be computed from the formula below. In it, N represents the minimum number of cylinders, and K represents the number of recording heads of the device. The SIZE-OF-VTOC value should be in tracks, and the result of the division should be rounded to the next highest integer.

N = 2 + (SIZE-OF-VTOC / K)

Note: For 2305 Models 1 and 2, 3330, 3340, and 3350 devices, all n cylinders of a minidisk are assigned for your use. No allowance is made for alternate track assignment.

INVOKING THE IBCDASDI PROGRAM

The IBCDASDI program is invoked for minidisks by specifying the operand

```
...,CYLNO=nnn
```

on the DADEF control statement (discussed previously). This control statement is passed to the IBCDASDI utility for processing. This operand specifies the size, in cylinders, of the minidisk to be initiated.

The IBCDASDI program, which is distributed as a CMS file with a filename of IPL and a filetype of IBCDASDI, should be spooled to your own virtual card reader. Control statements for the program can follow the last card or card image for the program, or can be entered via a separate input device.

To execute the IBCDASDI program:

1. Make sure ECMODE is off for your virtual machine; use the QUERY command to determine this. Use the SET command to modify if required.

2. Spool a copy of the IBCDASDI object module to your virtual card reader, or mount and attach the tape containing the object program.

3. Load the object program from the virtual reader or tape by issuing the CP IPL command for the appropriate virtual device address. When the program is loaded, an enabled wait state is entered with the address field of the PSW containing the hexadecimal value FFFF.

4. When the program is loaded and waiting for input, signal attention from the virtual console device. The message

   DEFINE INPUT DEVICE

   is sent to your virtual console. Enter the following response from the virtual console:

   INPUT=type,cuu

   where:

   type is the device type of the device containing the control statements. Valid device types are 2400, 2540, 3410, 3420, and 3505.

   cuu is the device address of the device containing the control statements.

Control statements are printed on the message output device. At the end of job, the END OF JOB message is printed on the message output device and the program enters the wait state.

VMFDUMP Service Program and How To Use It

CREATING DUMP FILES

Conditions can occur during VM/370 processing that cause an abnormal program ending (abend). The system operator can cause this abend condition by pressing the SYSTEM RESTART key on the system console. If VM/370's control program (CP) recognizes an error condition or program situation where program recovery is not possible or desirable, abend occurs. The abend procedure creates a historical record of the incident by dumping register and processor storage contents to a previously designated device.

The extent of the dump operation as well as the device that captures this information is specified by operands of the CP SET command. The format of the CP SET DUMP command is:

```
┌──────────────────────────────────────────────────────────────┐
│       │              ┌   ┐                                   │
│  SET  │  DUMP ⎧raddr⎫ │CP │                                   │
│       │       ⎩AUTO ⎭ │ALL│                                   │
│       │              └   ┘                                   │
└──────────────────────────────────────────────────────────────┘
```

raddr (real device address) designates either the system printer or a readied nine-track tape drive (three hexadecimal digits). If the printer address is specified, the data is printed online and system operations as well as virtual machine operations are suspended until the dump operation finishes and the subsequent system automatic restart operation occurs. This dump is unformatted. The printed dump is the end result of specifying the system printer in raddr. No further processing or copying of this file is possible. If raddr refers to a tape drive, the dump data must fit on one reel of tape. VM/370 does not support multiple tape volumes. The historical data on the tape is in print line format. You can further process this historical file by user-created programs or CMS commands. See "How To Print a CP Abend Dump from Tape" later in this section for an example of how CMS can do this.

AUTO spools abend dumps to a virtual card reader with a class of D for a specific userid. This userid either is assigned during system generation to a specific virtual machine user, or defaults to the userid of OPERATNS. Thus, only the specified user with a privilege class of C or E can invoke VMFDUMP and achieve desired results. The dump spool file can be manipulated by the user just like any other spool file, except that it can be interpreted correctly only by the VMFDUMP program. The device type and address of this virtual card reader can be determined by issuing a QUERY DUMP command. The system then responds with the device type and the channel and unit address of the system dump device.

CP dumps only CP storage.

ALL causes <u>all</u> of processor storage (that is, all data that relates to CP as well as all virtual machines) to be dumped to the selected device. If the ALL option is not specified, then the system dumps those areas that pertain to CP, and not those areas that pertain to user virtual machine operations.

Only dumps created by SET DUMP AUTO can be used by the IPCS VMFDUMP service program

USING THE VMFDUMP COMMAND

The IPCS VMFDUMP command uses the DMMEDM program to read the CP spool reader file that contains the system dump and write it on the CMS A-disk. The DMMEDM program then reads the dump information from the CMS disk, selecting and formatting various control blocks, which it directs to the spooled printer (per VMFDUMP operands). The program also prints a hexadecimal dump, then erases or retains the CMS file, depending on the options chosen. Only users with privilege class C or E can invoke VMFDUMP. In addition to producing formatted dump output, VMFDUMP automatically generates a problem report file from abend dump information and operator-prompted responses which aids in problem analysis, is described fully in the <u>VM/370 Interactive Problem Control System (IPCS) User's Guide</u>. The format of the VMFDUMP command is:

```
┌──────────────────────────────────────────────────────────────────┐
│           │  ┌           ┐  ┌           ┐                        │
│ VMFDUMP   │  | PRBnnnnn  |  |ERASE      |                        │
│           │  └           ┘  |NOMAP      |                        │
│           │                 |NOHEX      | ...                    │
│           │                 |NOFORM     |                        │
│           │                 |NOVIRT     |                        │
│           │                 └           ┘                        │
└──────────────────────────────────────────────────────────────────┘
```

<u>where</u>:

PRBnnnnn specifies the dump number previously assigned for this problem. This dump exists as a CMS file PRBnnnnn DUMP. No problem report is produced. If no number is specified, the dump is read from the spool file and assigned the next sequential problem number (PRBnnnnn) as the dump number.

ERASE specifies that the dump being processed is to be erased.

NOMAP specifies that a load map is not to be printed.

NOHEX specifies that the hexadecimal portion of the dump is not to be printed.

NOFORM specifies that no formatted control blccks are to be printed.

NOVIRT specifies that only the real machine control blocks are to be formatted. This option is ignored if NOFORM is also specified.

Usage Notes:

1. The *current* CP load map (NUC MAP) is *required* for VMFDUMP function to execute correctly. This file must reside on the A-disk of the IPCS virtual machine used to invoke VMFDUMP. If VMFDUMP does not find NUC MAP or finds that it is invalid, you are asked to enter the name of the current load map. You may add the load map to the dump by using the MAP subcommand of DUMPSCAN.

2. The dump specified by PRBnnnnn will be handled according to the options used. When a dump number is specified in the VMFDUMP command, no data extraction occurs. The fcllowing example is the suggested command for processing a new dump:

```
VMFDUMP  NOFORM  NOHEX
```

where:

NOFORM inhibits the printing of formatted control blocks

NOHEX inhibits the printing of the hexadecimal portion of the dump.

VMFDUMP will:

- Read the dump from the spool file.
- Assign the next sequential problem number as the dump number.
- Perform the data extraction.
- Create a problem report.

The filenames of dumps created by the VMFDUMP are of the form PRBnnnnn DUMP.

Upon completion of the problem report generation function of the extended VMFDUMP, a search is made of the symptom summary file for a possible duplicate problem. Up to ten possible duplicate problems are displayed with their associated status. Whether or not the problem was a duplicate, an entry is made in the symptom summary file for this problem.

SAMPLE OF INITIATING THE VMFDUMP OPERATION

Following typical logon procedures, the user can take the following steps to initiate a VMFDUMP operation.

```
query rdr
ORIGINID FILE CLASS RECDS  CPY HOLD
SYSTEM   0126 D DMP 000065 00  NONE
SYSTEM   0088 D DMP 000061 00  NONE
```

Entering QUERY READER discloses that these files are system generated class D files. This alerts you to the fact that the files are CP Dumps. These dumps, although not usable in their present state, are translated by subsequent VMFDUMP processing.

```
ipl CMS
CMS 08/27/75 WED 09.45.20
vmfdump noform nohex
DO YOU WISH TO USE THE 'IPCS' VERSION OF VMFDUMP? (YES OR NC)
yes
VM/370 DUMP, EXTRACT, EDIT, AND PRINT
VM/370 SYSTEM ABEND CODE PRG005 DATE 08/26/75 TIME 13.42
DO YOU WANT THIS DUMP?  (YES OR NO)
yes
CREATING FILE PRB00013 DUMP A1
```

(data extraction process)

```
ASSIGNED PROBLEM NUMBER PRB00013
CMS 08/29/73 WED 09.45.20
3
ENTER FN FT FM OF SUPPORTING DATA AND DESCRIPTION
E.G. PROB00013 CONSOLE A1
```

(User enters pertinent data)

```
ENTER ADDITIONAL TEXT 80 CHARS. PER LINE
ENTER TEXT OR NULL
```

(User enters needed text or presses return key or its equivalent)

```
THE FOLLOWING PROBLEM(S) ARE POSSIBLE DUPLICATES,
00012 OPENUSER 08/19/75
R;
```

CMS is loaded into the user's virtual machine and VMFDUMP is invoked. In this example, he elects to create a CMS dump file. The processing messages related to creating the file follow. If the file is processed and saved using the VMFDUMP program, the CP spool file is erased, and the newly created CMS file becomes PRBnnnnn DUMP, where nnnnn is a number from 00001 through 99999.

<u>Note</u>: If VMFDUMP terminates or enters a loop condition because of an unrecoverable or invalid situation encountered while attempting to format a dump, VMFDUMP should be invoked again with the NOFORM operand.

HOW TO PRINT A CMS DUMP FILE

Use the VMFDUMP command to print a previously created dump file under CMS by entering:

```
     VMFDUMP  PRBnnnnn [options]
```

at the terminal. The message:

```
     PRINTING FILE 'PRBnnnnn Dnn DUMP'
```

then occurs at the terminal. When the specified file has been formatted and printed, one of two completion messages is displayed:

```
     DUMP FILE 'PRBnnnnn Dnn DUMP' PRINTED AND KEPT

     -- or --

     DUMP FILE 'PRBnnnnn Dnn DUMP' PRINTED AND ERASED
```

HOW TO PRINT A CP ABEND DUMP FROM TAPE

When the CP abend dump is set to a tape, the records are 132 characters long, unblocked, with a blocksize of 132 and carriage control characters.

If the CP dump unit has been specified as a tape drive, and one or more dumps have been placed on the tape, use the following procedure to print the dumps.

1. Log on to the VM/370 system with any userid that has the capability of running CMS. No other special privilege classes or options are required.

2. Attach a tape drive to the virtual machine as address 181.

3. Mount the tape containing the CP abend dumps.

4. IPL the CMS system and perform the usual access requirements.

5. Issue the following CMS commands:

```
        FILEDEF ddname1 PRINTER (RECFM UA LRECL 132)
        FILEDEF ddname2 TAP1 (9track DEN nnnn RECFM U LRECL 132)
        MOVE ddname2 ddname1
        cp close prt
```

Step 5 can be repeated for as many dumps as are on the tape. Note that the CP dump routines write two tape marks at the end of each file. Therefore, to process the next dump, the TAPE FSF command line must be issued to position the tape for reading the next dump file.

NETWORK DUMP OPERATIONS

If 3704/3705 operations are erratic, fatal hardware errors occur, or some other internal error appears, the Communications Controller's storage should be dumped. The NETWORK DUMP command dumps the contents of 3704/3705 storage for EP 3704/3705 control programs, if unit check or IPL required conditions are detected.

The format of the NETWORK command with the DUMP operand is:

```
┌──────────────────────────────────────────────────────────────────┐
│            │                ┌     ┐                              │
│  NETWORK   │   DUMP raddr   │IMMED│                              │
│            │                │AUTO │                              │
│            │                │OFF  │                              │
│            │                └     ┘                              │
└──────────────────────────────────────────────────────────────────┘
```

where:

raddr is the real hexadecimal address of the 3704/3705.

IMMED is the default operand; it forces an immediate dump. The IMMED operand, if specified, does not reload the control program. Before 3704/3705 resources can be used again, the control program must be reloaded. To reload the control program after the "NETWORK DUMP raddr IMMED" command has executed, use the "NETWORK LOAD raddr ncpname" command.

If the IMMED operand is specified, a check is made to determine whether the "IPL required" sense status is present. If it is not, the following message occurs:

```
DMKNET461R CTLR raddr IPL NOT REQUIRED; ENTER YES TO
           CONTINUE:
```

This pause in operations allows the operator an opportunity to check the NETWORK DUMP command line before engaging or terminating the operation.

AUTO causes a dump if VM/370 subsequently detects a unit check condition or "IPL required" condition. If AUTO is specified, each time a dump is taken, the Communications Controller is reloaded with the 3704/3705 control program that was previously active.

OFF resets a previously set AUTO (automatic dump) status.

Note: The dumps produced by the NETWORK command cannot be processed by the VMFDUMP service program. NETWORK-initiated dumps are processed by the NCPDUMP (Network Control Program DUMP) service program created for this task.

NCPDUMP Service Program and How To Use It

NCPDUMP is a CMS command. It processes CP spool reader files created by 3705 dumping operations, that is, dump files that are produced as a result of the CP NETWORK command specified with the DUMP operand and either automatic or immediate mode.

The NCPDUMP file processing operation can include:

- Erasing a specific CMS NCPDUMP file after printing it.
- Formatting the dump.
- Printing the dump.
- Assigning an identifier to the CMS NCPDUMP file.
- Creating the CMS NCPDUMP file from the spool file.

Although NCPDUMP is a CMS command, its use is restricted to the user identified by the SYSDUMP operand of the SYSOPER macro in DMKSYS during VM/370 system generation. The operation of NCPDUMP is similar to VMFDUMP operations. A general description of the NCPDUMP operation follows the command description.

The NCPDUMP command has the following format:

```
| NCPDUMP | [DUMPxx] [ ([ERASE][NOFORM][MNEMONIC][NCPBUFF][) ]] |
```

where:

DUMPxx is the filename of a CMS file containing a 3704/3705 Communications Controller program dump. This dump was created by a previously invoked NCPDUMP command with the ERASE operand not specified.

ERASE erases the current CP DUMP file or a specified DUMPxx (filename), saved CMS file.

NOFORM specifies that a formatted control block is not desired.

MNEMONIC includes 3705 Assembler mnemonic operation codes in the printed output.

NCPBUFF specifies that a formatted listing of the NCP buffer pool is desired.

The NETWORK command invoked with the DUMPxx operand, as stated previously, produces CP files that contain the contents of a designated 3704/3705 Communications Controller unit buffer. These CP files reside as a spooled reader input assigned to a system-designated user. The CMS NCPDUMP command invoked by this user formats (if requested) and prints the contents of these files.

The NCPDUMP program creates a CMS file with a filename DUMPxx and a filetype of NCPDUMP, and erases the original spooled NETWORK initiated dump reader file. The created CMS file is erased if you specify ERASE; otherwise it is kept.

A maximum of ten dumped spooled files can be processed and saved, and later recalled, if necessary, by the system assignment of an xx identifier suffix to the CMS DUMPxx filename. The "xx" is a decimal number from 00 to 09, depending on any existing files of a similar name. For example, if the files DUMP00 NCPDUMP and DUMP01 NCPDUMP already exist, the new file would be called DUMP02 NCPDUMP. The file thus created is retained for later use unless the ERASE option is specified, in which case the file is erased immediately after the dump is printed.

ZAP Service Program and How To Use It

ZAP is a CMS command that modifies or dumps MODULE, LOADLIB, or TXTLIB files. It may be used to modify either fixed or variable length MODULE files. It is for use by system support personnel only.

Input control records control ZAP processing. They can be submitted either from the terminal or from a disk file. Using the VER and REP control records, you can verify and replace data or instructions in a control section (CSECT). Using the DUMP control record, you can dump all or part of a CSECT, or an entire member of a LOADLIB or TXTLIB file, or an entire module of a MODULE file.

The format of the ZAP command is:

```
┌──────────────────────────────────────────────────────────────────────────┐
│          │ (MODULE )                                                     │
│  ZAP     │ <LOADLIB > [libname1 ... libname3][(option...[)]]             │
│          │ (TXTLIB )                                                     │
│          │                   options:                                    │
│          │                   ┌               ┐┌        ┐                  │
│          │                   │TERM           ││PRINT   │                  │
│          │                   │INPUT filename ││NOPRINT │                  │
│          │                   └               ┘└        ┘                  │
└──────────────────────────────────────────────────────────────────────────┘
```

where:

MODULE
LOADLIB
TXTLIB   indicates the type of file that is to be modified or dumped.

libname   is the library name containing the member to be modified or dumped. You can specify one to three library names. The libname is valid only for LOADLIB and TXTLIB files.

Options:

```
     ┌        ┐
TERM │PRINT   │
     │NOPRINT │
     └        ┘
```

indicates that input to the ZAP service program is submitted through the terminal. If you specify TERM, the prompting message ENTER: is issued, and you can then enter input control records up to 80 characters long. If you specify PRINT with TERM, all output prints on the printer, but only error messages display at the terminal. If you specify NOPRINT with TERM, nothing prints on the printer. All output except control records displays at the terminal.

```
               ┌        ┐
INPUT filename │PRINT   │
               │NOPRINT │
               └        ┘
```

specifies that input is submitted from a disk file, filename.

This file must have a filetype of ZAP, and must be a fixed 80-byte sequential file residing on any accessible device. If you specify PRINT with INPUT filename, all output produced by the ZAP service program prints on the printer. In addition, commands and control records in error and error messages display at the terminal. If you specify NOPRINT with INPUT filename, nothing prints on the printer. All output displays at the terminal.

Figure 12 shows the resulting output resulting from valid option combinations:

OPTIONS	PRINT	NOPRINT
INPUT	Commands and control records in error and error messages on the terminal. Everything to printer.	Everything on the terminal. Nothing on the printer.
TERM	Only error messages on the terminal. Everything on the Printer.	Everything except control records on the terminal. Nothing on the printer.

Figure 12. Valid Options and Their Output

ZAP INPUT CONTROL RECORDS

Eight types of ZAP control records exist: DUMP, NAME, BASE, VER or VERIFY, REP, LOG, COMMENT, and END.

ZAP control records are free-form and need not start in position one of the record but the ZAP program can accept only 80 characters of data for each control record. Separate all information by one or more blanks. All address fields including disp (displacement) fields in VER and REP control records must contain an even number of hexadecimal digits, to a maximum of six digits (0D, 02C8, 014318). Data fields in VER and REP control records must also contain an even number of hexadecimal digits, but are not limited to six digits.

If you wish, you may separate the data anywhere by commas (for example, 83256482 or 8325,6482). The commas have no effect on the operation.

The program sets the NOGO switch on if a control record is found to be in error. A file cannot be modified once the NOGO switch is turned on. The next valid NAME record turns the NOGO switch off. This means that if the control record is the NAME record, all succeeding records are ignored until the next NAME, DUMP, or END record. For any other error, only REP control records that follow are ignored.

<u>DUMP Control Record</u>

The DUMP control record resets the NOGO switch off. The DUMP contrci record must not immediately precede a BASE, VER, or REP control record. A NAME control record must precede the BASE, VER, and REP control records (if any) that follow a DUMP control record.

The DUMP control record allows you to dump a portion or all of a specified control section, or the complete member or module. The format of the output of the dump is hexadecimal with an EBCDIC translation of the hexadecimal data.

The DUMP control record is optional. The format of the DUMP control record is:

```
┌──────────────────────────────────────────────────────────────────────┐
|                                                                      |
|                  ┌                                       ┐           |
|DUMP {membername} |csectname [startaddress [endaddress]]  |           |
|     {modulename} |ALL                                    |           |
|                  └                                       ┘           |
|                                                                      |
└──────────────────────────────────────────────────────────────────────┘
```

<u>where</u>:

membername
: is the name of the member to be dumped, or the member that contains the CSECT(s) to be dumped. This member must be found in one of the libraries specified in the ZAP command line. However, if the library is a CMS TXTLIB, its directory does not contain member names. Therefore, the program ignores the member name (although you must specify it), and the program searches for the csectname (which you must specify).

modulename
: is the name of the module to be dumped, or the module that contains the CSECT(s) to be dumped. If you specify a module that has no loader table, the program dumps the entire module.

csectname is the name of the control section that is to be dumped. If you do not specify csectname, the program dumps only the first CSECT. The csectname is required for CMS TXTLIBs, optional for OS TXTLIBs, LOADLIBs, and MODULE files. (See the discussion of csectname under "Name Control Record.") You must not specify csectname for a module created with the NOMAP option.

ALL specifies to the program to dump all CSECTs within the specified member or module. You can specify ALL for MODULE files, LOADLIBs, and OS TEXTLIBs, but not for CMS TXTLIBS. If you wish to dump all the CSECTs in a member of a CMS TXTLIB, you must issue a separate DUMP control record for each CSECT.

startaddress
: is the location within the specified CSECT where the dump is to begin. This must be two, four, or six-hexadecimal digits. The start address is the displacement from the beginning of the CSECT. For example, if you wish to start dumping at address 08 in a CSECT that begins at location 400, you specify start address 08, not 0408.

endaddress
: is the last address to be dumped. This must be two, four, or six-hexadecimal digits. If you specify no address, the program dumps from the start address of the CSECT to the end of the CSECT. Note that start and end addresses apply only when you specify a csectname.

NAME Control Record

The NAME control record specifies the member or module and CSECT that contain the data to be verified or replaced by the ZAP operation. The format of the NAME control record is:

```
NAME  { membername } [csectname]
      { modulename }
```

where:

{membername}
{modulename}
is the member or module that you want to be searched for the desired CSECT.

csectname is the name of the desired control section. You must specify csectname if the CSECT you wish to modify is in a CMS TXTLIB (that is, TXTLIB created by the TXTLIB command from CMS TEXT decks that do not have a NAME card following the END card). The directory of a CMS TXTLIB contains only CSECT names and no member names. The CSECT name specified in the NAME record is compared with CSECT names in the directory. If a CSECT match is found and no member name match is found, the member selected is the one that contains the CSECT name. The csectname is optional if the CSECT you wish to modify is a LOADLIB or an OS TXTLIB (that is, a TXTLIB created by the TXTLIB command from CMS TEXT decks that have a NAME card after the END card). The dictionaries of the specified libraries are searched for the member name and the member is then searched for the CSECT name, if you specified one. If you do not specify csectname for a LOADLIB or an OS TXTLIB, the program uses the first control section. The csectname is optional for a MODULE file. The module named in the NAME control record is located and, if you specified csectname, the first record is read to determine the number of records in the module and the availability of a lcader table, which the program can then search for the csectname. If you do not specify csectname, the program uses the beginning location of the module. You are not allowed to specify csectname if the module was created with the NOMAP option. The NAME control record must precede the BASE, VER, and REP control records. If it does not, the program sets the NOGO switch on.

BASE Control Record

The BASE control record adjusts displacement values for subsequent VER or REP control records for a CSECT whose starting address is not location zero in an assembly listing. The format of the BASE control record is:

```
BASE    address
```

where:

address is the starting address of the CSECT. The address must be two, four, or six hexadecimal digits. For example, for a CSECT starting at location 400, you would specify the BASE 0400 in the BASE control record. If a subsequent VER card requests verification of location 0408, the BASE of 0400 is subtracted from 0408, and the program verifies location 08 in the CSECT. This example applies if you specify TXTLIB, LOADLIB, or MODULE and the module map is present. However, if nc module map is present for a MODULE file (that is, the module was generated with the NOMAP option), then all operations are performed as if the BASE address is location 0. For example, if you specify a BASE of 400 and the address you wish to inspect or modify is 408, then you must specify 08 and not 408 in REP and VER control records. The address in this case is from the start of the module. If you do not specify csectname in the NAME control record, you cannot specify any BASE value other than 00. The BASE control record is optional. See the discussion under "VER or VERIFY Contrcl Record." If specified, the BASE control record must follcw the NAME record, but it need not follow the NAME record immediately. For example, you could have the following sequence of control records: NAME, VER, REP, BASE, VER, REP.

VER or VERIFY Control Record

The VER control record requests verification cf instructions or data within a CSECT. If the verification fails, the program does not perform a subsequent REP operation until it encounters another NAME control record.

The VER control record is optional. More than one VER record can follow a single NAME record.

The format of the VER control record is:

```
{ VERIFY }    disp    data
{ VER    }
```

where:

disp is the hexadecimal displacement of the data to be inspected from the start of the CSECT, if you did not submit a BASE control record for this CSECT. If you did submit a BASE control record, then disp is the actual location of the data. The disp must be two, four, or six hexadecimal digits. This displacement does not have to be aligned on a fullword boundary. If this displacement value is outside the limits of the CSECT specified by the preceding NAME control record, the VERIFY control record is rejected.

data is the data against which the data in the CSECT is to be compared. This must be an even number of hexadecimal digits. For example, if the location you wish to verify is 3CC, and the CSECT begins at location 2B0, you can either issue:

```
BASE 02B0
VER 03CC data
```

or you can omit the BASE control record, subtract the CSECT start address from the address of the data, and issue:

```
VER 011C data
```

This also applies to the disp operand of the REP control record.

Note: If the VER control statement references a TXTLIB file that contains ORG statements causing more than one occurrence of an address, only the first occurrence is tested when the displacement on the VER statement references the ORG statement addresses.

REP Control Record

The REP control record modifies instructions or data at the specified location within the CSECT that you specified in a preceding NAME control record. The data specified in the REP control record replaces the data at the CSECT location specified by the disp operand. This replacement is on a "one-for-one" basis; that is, one byte of data defined in the control record replaces one byte of data at the location that you specified. If the replacement fails, the program does not perform additional REP operations until it encounters another NAME control record.

The REP control record is optional. More than one REP record can follow a single NAME record.

The format of the REP control record is:

```
REP    disp    data
```

where:

disp is the hexadecimal displacement of the data to be replaced from the start of the CSECT, if you did not submit a BASE control record for this CSECT. If you did submit a BASE control record, then disp is the actual location of the data. The disp must be two, four, or six hexadecimal digits. This displacement need not address a fullword boundary. If this displacement value is outside the limits of the CSECT being modified, the program does not perform the replacement operation.

data is the data that is to replace the data in the CSECT. This must be an even number of hexadecimal digits.

Note: Although you do not have to verify a location before replacing data, you should do so to make sure that the data being changed is what you expect it to be.

LOG Control Record

The LOG control record allows the user to specify a unique fix number for the module specified in the NAME control record. Fix numbers are recorded in a log file associated with the named module. The format of the LOG control record is:

```
┌─────────────────────────────────────────────────────────────────────┐
│                          ┌                        ┐                 │
│  LOG        {fixnum}     │ filetype [user data]   │                 │
│                          │ ZAPLOG                 │                 │
│                          └                        ┘                 │
└─────────────────────────────────────────────────────────────────────┘
```

where:

fixnum specifies the number associated with the fix. Its length may vary from one-to-eight alphameric characters.

filetype specifies the file type of the log. If not specified, filetype defaults to ZAPLOG.

user data specifies any data that the user wishes to enter into the log. If user data is specified, the filetype operand is mandatory.

Note: The LOG control record is optional and is allowed only if valid NAME and REP control records are found. The file name iobtained by the log routine from the module named in the NAME control record.

Comment Control Record

The ZAP program ignores comment control records. If the PRINT option is in effect, the program prints the comments. The format of a comment record is:

```
┌─────────────────────────────────────────────────────────────────────┐
│                                                                     │
│  *  comment                                                         │
│                                                                     │
└─────────────────────────────────────────────────────────────────────┘
```

The asterisk must be followed with at least one blank before you enter the text.

END Control Record

The END control record ends ZAP processing. The END record is required and must be the last control record. The format of the END control record is:

```
┌─────────────────────────────────────────────────────────────────────┐
│                                                                     │
│  END                                                                │
│                                                                     │
└─────────────────────────────────────────────────────────────────────┘
```

SPECIAL CONSIDERATIONS FOR USING THE ZAP SERVICE PROGRAM

Before you use the ZAP command against MODULE files, you can use the MODMAP command to determine whether a module map exists and what it contains.

When a ZAP input file has more than one pair of VER and REP control records and a VER control record (other than the first) fails, you must remove the records prior to the failing record and correct the error before you issue the ZAP command again. Otherwise, the file being modified returns to its original status.

If you issue a REP control record against a file that contains an undefined area (for example, a Define Storage area) within the REP data field and do not issue a VER control record prior to the REP control record, the bytes prior to the undefined area, if any, are modified and all the bytes after the undefined area are not modified. The program prints warning message DMSZAP248W.

If the file tc be dumped contains undefined areas (such as a DS or ORG statement in a TXTLIB member), the hexadecimal portion of the dump contains blanks to indicate that the corresponding positions are undefined.

Section 5. Operator Spooling Functions

Spooling

Input and output files for use and access by virtual machines through unit record devices are maintained by CP as disk data files using a mechanism called spooling. Individual files can be identified and manipulated using various console functions. The disk records are chained to form a logical file from dynamically assigned areas on specially formatted CP disk areas. Data records from disk are read into available page space obtained through the CP paging mechanism. The data records contain the actual data to be used and the CCWs to properly control the format. These CCWs are directly executed to perform the actual unit record operations on the real hardware. CP can support any number of virtual and real unit record devices given sufficient system resources.

The data is placed in the spool buffers through the virtual machine unit record simulation routines in CP. Certain spool files have a special data format (system dumps, for example) and are accessed using a special interface.

SPOOLING CONSIDERATIONS

VM/370 spooling facilities allow several virtual machines to share one or more unit record devices. Since virtual machines controlled by CMS ordinarily have modest requirements for unit record I/O, such device sharing is quite advantageous, and it is the standard mode of system operation.

Each user has, as a general rule, a virtual reader, a virtual punch, and a virtual printer as his spooling devices. In addition, the virtual console can also be classified as a spool file generator as all input and output to the console can be logged on a spool file. This console log, and the files created by the user's virtual spooling devices, can be processed by the real unit record devices that attach to the system.

CP controls and schedules the operation of the real unit record devices via spooling techniques. Virtual machine SIO instructions directed to those unit record devices designated as spool devices in the user directory entry are intercepted and modified by CP. CP generates another I/O operation, transparent to the virtual machine, which replaces the one specified. The new operation is directed to a CP spooling disk area which acts as intermediate storage between the real unit record device and the virtual machine. The data transfer operation between a spooled unit record device and the virtual machine is, in reality, between a CP spool file and the virtual machine. Spool file records are page size (that is, 4096 byte blocks), and are transferred between storage media via the CP paging mechanism.

SPOOLING FUNCTIONS

CP spooling support performs the following five functions for virtual machines:

1. It simulates, with software routines, the operation of the virtual unit record devices that are attached to each user's virtual machine. The simulation makes it appear that the program in the virtual machine controls real unit record devices. Unit record device simulation involves the interception and interpretation of user Start I/O (SIO) instructions, the movement of data to and from the user's virtual storage space, and the reflection of interruption codes and ending conditions in the virtual machine.

2. It operates the real unit record equipment attached to the real machine. Spooling moves print-image and punch-card-image files to a real printer and card punch, and creates spool files from data read in from the real card reader.

3. It provides an easy-to-use interface between the virtual machine users, the system operator, and the spooling system to allow flexible and easy switching of system resources between many users. A set of general user and operator commands can request these functions.

4. It allows keyed-in CP commands and responses as well as virtual machine console input and output to be placed on disk instead of, or in addition to, being displayed at the terminal. You can initiate or terminate virtual console spooling at any time during a terminal session.

5. It provides the ability to spool files across the RSCS teleprccessing network.

Spooling commands can be separated by type into those that affect virtual devices, those that affect real devices, and those that affect queued spool files within the system. The commands that affect virtual devices are available to all general users; a user may affect the status of devices that attach to only his virtual machine. For a discussion of the Class G spooling commands, see _VM/370 CP Command Reference for General Users_.

Only the spooling or resource operator can use commands that affect the status of the real spooling devices. Commands affecting the user's virtual machine closed spool files that are awaiting processing are available to all users, with some additional capabilities available to the spooling operator. For example, a user can alter characteristics of only those files that have the same userid as his, but the spooling operator can change the status of any spool file in the system. The "CP Commands" section of this publication describes these commands.

You can create input spool files (that is, data available at a specific user's virtual card reader), when you place cards in the real card reader that are preceded by a special VM/370 card that identifies the virtual machine userid of the user requesting spool files.

When the virtual machine operating system writes to a virtual printer or card punch, it creates output spool files and stores them on direct access devices. Real output is scheduled for a real printer or card punch, or for remote output, whenever a user logs off the system or issues a CP spooling command to close the file.

You can transfer specific files from the spooled printer or card punch of a virtual machine to the card reader of the same or another virtual machine. (A virtual card reader is not limited to 80-character records.) Files are not physically printed or punched when transferred between virtual unit record devices by the spooling routines. With this method, you can make files available to multiple virtual machines, or to different operating systems executing at different times in the same virtual machine.

The Remote Spooling Communications Subsystem, with the CP spooling system, provides support for spooling across a teleprocessing network. For detailed information about RSCS, see the _VM/370 Remote Spooling Communications Subsystem (RSCS) User's Guide_.

VM/370 spooling includes many options for the virtual machine user and the real machine operator. These include printing multiple copies of a single spool file, backspacing any number of printer pages, and defining or reordering spooling classes for real output scheduling.

Real printers with the Universal Character Set feature must have the block data check set when using VM/370 spooling. The LOADBUF command automatically sets the block data check. For more details on spooling, see "Spooling Considerations."

Virtual Console Spooling

The terminal user can spool virtual console output and virtual console input keyed in by him in addition to typing or displaying it at the terminal. Furthermore, if the virtual console is disconnected and the virtual machine is active, the console spool file acquires all console output that would normally print or display at the terminal. Virtual console I/O consists of all initiated CP commands and responses, as well as the communication to and from the operating system running in the virtual machine.

You start virtual console spooling when you issue the command:

```
SPOOL  CONSOLE  START
```

You close the console spool file at logoff time or when you stop and close the file. When you close the console spool file, it goes on the printer spool file and then the operator can manipulate it in the same way as any other printer spool file.

The display output of local or remote 3270s in DISPLAY mode (every 16 lines) is written automatically on the spool file if the operator is the system operator and the device was specified as a graphic device supported as a virtual machine operator's console.

Console spooling for the system operator continues even if the operator disconnects from one terminal and logs onto another. To stop the automatic console spooling, the operator must issue the SPOOL CONSOLE STOP command. If the system fails, up to 16 lines of output may be lost from the system operator's spool file, but the system closes the console spool file when the failure occurs. However, virtual machine operators may lose up to one page of spooling data in a system failure because their spooling buffer is one page (4096 bytes) in size.
In regard to DASD I/O, errors occurring on spool and paging devices produce error messages that appear on the system operator's terminal.

A user can start or stop virtual console spooling any time during a terminal session with the SPOOL CONSOLE command. For a description of the CP SPOOL and CLOSE commands, see the _VM/370 CP Command Reference for General Users_.

Note: The user cannot use console spooling to stack commands for subsequent execution. The console spool file is for historical purposes only.

I/O Error Handling

If a permanent I/O error occurs during a real I/C operation, you restart the spooled files as follows:

- Printer files--from the beginning of the current page (the last skip to channel 1).
- Punch files--from the beginning of the spool file.
- Reader files--from the beginning of the real deck (the operator must normally place the card deck back in the reader).
- Console files--do not restart. CP ends console spooling and purges the console file. The virtual machine continues without further console spooling.

Note: If an I/O error occurs while punching accounting cards, CP repunches only the error card and not the whole file as in the case of normal punch spool files.

SPOOL BUFFERS

The buffers used in virtual machines while collecting and writing spool data are each one page (4096 bytes) long, and contain both the data to be transcribed and all CCWs necessary for operating the unit record devices that perform the transcription. System failures that end system operation can cause virtual machine operators to lose up to one page of spooling data.

Buffers used for the temporary storage of spool data on its way between auxiliary storage and the user's virtual machine are allocated from a pool of virtual pages that belong to CP. Direct access devices provide the auxiliary storage necessary for CP spool buffering.

CP uses unit record devices to spool input and output operations. However, certain features or functions that pertain to this group of machines are not supported. Consult the VM/370 restrictions listed in the _VM/370 Introduction_ and the list of devices suitable for spooling devices (both unit record and direct access) listed in "Appendix B: System Requirements." Also see the _VM/370 Planning and System Generation Guide_.

SPOOL FILES

Each spool file in the system has a number of attributes that are assigned to it, either explicitly or by default, at the time that it is created. These attributes and their values are as follows:

Filename and Filetype: Each of these consist of alphameric fields of up to eight characters. The CHANGE or CLOSE command can change these fields.

Spoolid: This is a system-assigned number between 1 and 9,900. It is automatically assigned when the file is closed, and is unique to that spooled file. To identify a given file, it is necessary to specify the userid of the file's owner, the device type, and the spoolid number. In most instances, the userid defaults to the ID of the user issuing the given command. Since the ID number, rather than the filename and filetype, is used as an identifier, duplicate user-assigned names do not present an identification problem.

Originating User ID: This is the ID of the file's creator, if the file has been internally transferred from the originator's virtual printer or punch to a new owner's virtual card reader.

Number of Copies: This is the number requested for an output spool file; it This is a number between 1 and 99. Unless specified by the user or operator, it defaults to 1.

Date and Time: This is the date and time that the file was created. Users create most files from virtual spool devices and the date and time indicate when the originator of the file closed the device.

Number of Records: This is a six-digit number indicating the number of logical record (printer or card images) in the file and is an indication of the size of the file.

OUTPUT CLASSES

In addition to the attributes described above, a file queued for output on a real unit record device always has an output class associated with it. A single alphameric character (A through Z, 0 through 9, and the special character *) controls what real device the file is to be printed or punched on, and the relative priority and sequence of output on the device. Although each file is assigned a single class, you can designate each real spooling output device to handle from one to four classes. The device processes only files that have a class code that corresponds to one of its own, and processes these files in the order you specify. For example, if you assign a printer the classes A, D, and 2, it always processes any printer file with class A before it searches the printer output queue for a file with class D, and all class D files are printed before any file with class 2. You can assign a class of * to a real output unit record device to allow that device to process any class spool file.

The output class for a file is assigned at the time of the file's creation and is the class that is associated with the virtual device that created it. When a user logs on to the system, the class associated with the device is the one defined in his user directory entry for that device; however, he may alter this class at any time with the SPOOL command. After files are closed and are awaiting output, their class can be altered with the CHANGE command either by the file's owner or by the spooling operator. The spooling operator can alter the standard assignments of output classes of a real output device with the START command.

Output priorities can also be rearranged by altering the hold status of a file. The system operator can hold or release the files of specific users, all files of a given type, or all files in the system.

Through judicious combination of the class and hold functions, the operator can exercise flexible control of the spooling functions. He can use the class function to direct output to specific printers containing special forms or special print train arrangements, and to force a specific priority scheme. The hold function can ensure the immediate processing of high priority output files, while allowing high volume or low priority jobs to be deferred for output until periods of low system usage.

SPOOLING COMMANDS

The commands shown in Figure 13 control VM/370 spooling operations.

	User Classification	
Command	General User	Spooling Operator
BACKSPAC		X
CHANGE	X	X
CLOSE	X	
DRAIN		X
FLUSH		X
FREE		X
HOLD		X
LOADBUF		X
ORDER	X	X
PURGE	X	X
REPEAT		X
SPACE		X
SPOOL	X	
START		X
TAG	X	
TRANSFER	X	X

Figure 13. CP Spooling Commands

This list represents all the CP commands that pertain to spooling operations. Users having privilege classes D (the spooling operator) <u>and</u> G (the general user) can issue these commands. Users with only the G privilege class cannot invoke class D commands.

The spooling operator has responsibility for the following unit record spooling operations. He must:

1. Verify that the special VM/370 user identification (ID) card precedes a user's physical deck of cards to associate the cards with the correct user's virtual machine (see "Spooled Card Input.")

2. Make sure that sufficient blank cards are available for the punch unit, and that the paper in the printer is the proper size, width, and number of copies, and that the proper print train is mounted and the print buffer is properly loaded.

3. Separate punched and printed output by user identification (userid) code.

4. Start, stop, restart, or rearrange the sequence schedules of spool files to be printed or punched.

The spooling commands are discussed (along with other operator commands) in the "Operator Commands" section.

NOLOG Password

The NOLOG password is reserved for users who are not assigned any privilege class. These users cannot log on to a virtual machine. The user with the NOLOG password can submit jobs to the CMS Batch facility via the real card reader. The user with the NOLOG password can also be the recipient of spool files sent by any other logged-on user only if the other user issued a SPOOL FOR command to the userid having the NOLOG password.

SPOOLED CARD INPUT

Spooled input from a real card reader requires a CP ID (identification) card before the VM/370 system can accept it for processing.

Note: Load cards for only one user at a time. VM/370 does not look for ID cards in the middle of the card deck.

The CP ID card must be inserted in front of the card deck. Then place the deck in the empty feed hopper. Then ready the reader device and push the EOF button. The reader immediately processes the cards (if no errors are encountered) and place the file on a direct access device. The file now resides in the virtual machine card reader that was identified by the CP ID card. Note that a file is transferred from the real reader to the virtual reader regardless of whether that user is logged on or off. The user may log on at any time to read the file.

CP ID Card Format

```
┌────────────────────────────────────────────────────────────────────────┐
│          │                    ┌                             ┐          │
│   ID     │  userid [CLASS n]  │NAME {filename filetype}│    │
│ USERID   │                    │     {dsname           }│    │
│          │                    └                             ┘          │
└────────────────────────────────────────────────────────────────────────┘
```

1. Begin punching in column 1.
2. The first field must be one of the following:

   ID
   USERID

3. Separate all fields by one or more blanks.
4. The second field must be the user's identification (userid), limited to eight characters.
5. The third field (optional) is CLASS n (if class is assigned); one or more spaces must separate the word CLASS and the class designation (n). (A through Z and 0 through 9 are valid class designations.) If the class option is not coded, the file defaults to class A. If the class option is coded, it must immediately follow userid.
6. The fourth field (optional) is the keyword NAME, followed by filename and filetype (or dsname alone). If only a dsname is specified, it can be 24 characters long. If both filename and filetype are coded, they are each restricted to a maximum of eight characters and must be separated by a blank.

An invalid or missing user CP ID card on the front of a card deck, when read, causes an error message on the system console.

Spool Printer and Punch Output

Printed and punched output from virtual machine users is directed to the appropriate real unit record device. The operator (class D) can control this spooling output by combinations of output class, hold status, and other spool command operands as discussed earlier.

Unless you used the NOSEP option to "START" a device, VM/370 prints a two-page separator between all output on printers and punches a four-card separator between all output on punch devices.

The printer separator gives the pertinent spool file characteristics as well as real device data. The separator prints the owner userid and distribution code in block letters on two pages. The separator does not print between multiple copies of the file.

The card separator gives the owner's userid and the distribution code. The distribution code is expanded to punch each character in four columns with two separating blanks so that visual interpretation of the punched data is easier.

Because output for a user file starts on a real device, the following message occurs:

```
{PRT} raddr {OUTPUT OF} userid FILE = file RECDS = norecs COPY = nn a typ
{PUN}       {REPEATED }
```

where:

raddr   is the real device address.

userid   is the identification of the owner of the spool file.

file   is the spool file spoolid number.

norecs   is the logical record count (lines or cards) for the file.

nn   is the current copy number being produced where 01 is the last copy.

a   is the spool file class.

typ   is the originating device typ (PRT, PUN or CON).

The OUTPUT OF message occurs when the file output first starts and the REPEATED message occurs for subsequent copies of the same file as they start.

As the card deck starts reading through the real device, the operator receives the following message:

```
RDR raddr INPUT FOR userid FILE = file
```

where:

raddr   is the real address of the card reader.

userid   is the user identification of the owner of the file from the ID card at the front of the deck.

file   is the spool file spoolid number assigned.

An invalid or missing user identification card on the front of a card deck, when read, causes an error message on the system console.

SPOOL FILES FOR VIRTUAL MACHINES RUNNING BATCH JOBS

Usually, when an installation has only one real reader, printer, or punch, the unit record devices must be shared by all virtual machines. Before VM/370 can process any virtual machine CP spool files and direct them to a real device, those files must be closed. For virtual machines executing batch jobs and using operating systems other than CMS, usually the operator who initially program loaded the virtual machine operating system must intervene to issue the CP CLOSE command.

A feature that closes CP spool files is available for OS/VS1. When the VM/VS Handshaking feature is active, VS1 closes the CP spool files by issuing the CP CLOSE command when the VS1 Job output from its DSD, terminator, and output writer is completed. Once output files are closed, they can be processed by VM/370 without operator intervention.

Without the VM/VS Handshaking feature, CP spool files are not sent to the real printer or punch until the virtual machine operator intervenes.

Using the CMS Batch Facility

The Batch Facility is a VM/370 programming facility that runs under CMS. It allows a VM/370 user to run jobs in batch mode by sending jobs from either his own virtual machine or the real card reader to a virtual machine dedicated to running batch jobs under the Batch Facility. This dedicated machine is generally set up at a terminal in the installation's computer room and is controlled by the system operator.

The Batch Facility virtual machine runs continuously, executing all jobs spooled to its virtual card reader from other virtual machines or from the real card reader. The batch operator need pay no attention to the batch machine once he has started, and disconnected it.

STARTING THE BATCH VIRTUAL MACHINE

The system operator starts the batch virtual machine by logging on with a batch userid and loading CMS using the CP IPL command.

The Batch Userid

Every installation in which the CMS batch facility is available should establish one or more common userids for the Batch Monitor virtual machine. Users can then spool their files for execution to the card reader for that batch userid.

It is the operator's responsibility to log on the VM/370 system using the batch userid that has been established for his installation.

The userid established for the batch facility virtual machine must have a read/write work disk in its directory at virtual address 195. The 195 disk is erased when it is accessed as the A-disk at the beginning of each job.

Invoking the Batch Facility

The batch facility virtual machine is invoked by the batch operator when he issues the CP IPL command with PARM BATCH or CP IPL followed by the CMSBATCH command. The latter method follows:

```
ipl cms
CMS mm/dd/yy WED 17.58.48
cmsbatch
Y/S (19E) R/O.
THE FOLLOWING NAMES ARE UNDEFINED:
 BATEXIT1 BATEXIT2
R; T=0.14/0.39 08:47:40
WAITING FOR THE READER
```

The operator may now disconnect the batch machine terminal, if he wishes, using the CP DISCONN command. The Batch Facility will IPL itself after each job is executed.

All virtual machine (CMS) console output is automatically spooled to a file to be printed after the program output at the real system printer. All commands entered through the virtual reader are displayed on the console to allow them to appear in the console output file. If the batch terminal is disconnected, only CP and batch initialization messages are displayed at the terminal.

If an installation wishes to use a saved system in running batch jobs, the operator must enter the name of the saved system in the CMSBATCH command line.

CONTROLLING THE BATCH FACILITY

The batch facility permits CP and CMS commands to be issued from the batch job stream submitted by the user.

The following CP commands, and only these commands, control the batch virtual machine:

- CHANGE - May not be used to change the virtual card reader.
- CLOSE - May not be used to change the virtual card reader.
- DETACH - May not be used to change the virtual spool devices or the system, IPL, or 195 disks.
- DUMP
- DISPLAY
- LINK - A maximum of 10 links may be in effect at any one time; and, LINK must be used as shown in the following format:

  ```
  CP LINK userid vaddr vaddr mode password
  ```

- MSG
- QUERY
- SET - Only the following SET command operands are allowed under CMS batch:

ABBREV	NONSHARE
DOS	RDYMSG
IMPEX	SYSNAME
LDRTBLS	UPSI

- SPOOL - May not be used to change the virtual card reader.
- STORE
- TAG

When CP commands are used to control the CMS batch virtual machine, they must be preceded by the CMS CP command, because the implied CP function is disabled when batch initializes a job.

The following CMS commands are disabled under the Batch Monitor:

```
READCARD
DISK LOAD
```

The FILEDEF and ASSGN commands are also disabled for use in defining the virtual card reader.

A more complete description regarding control of the CMS Batch virtual machine is contained in the *VM/370: CMS User's Guide*. It describes the user control cards and suggests control techniques as well as how to control the batch machine using EXEC procedures.

The batch virtual machine spools output resulting from program execution to the system printer. Output is printed under the submitting userid, with the submitting userid's distribution code, a spool filename of CMSBATCH, and a spool filetype of JOB (unless a job name was specified on the /JOB card).

The console output is always spooled. Therefore, if the console is disconnected, the CMS console output is spooled to a file that is printed following the user's program execution output at the real system printer, with the submitting userid as distribution code, a spool filename of BATCH, and a spool filetype of CONSOLE.

If the CP TAG command has been used to identify spool files or to direct these files to other virtual machines or remote work stations, BATCH resets the spooling devices for the next job.

PURGING, REORDERING, AND RESTARTING BATCH JOBS

When required, the spooling operator can control the execution of batch virtual machine jobs by purging, reordering, and restarting them; by the same token, because all the closed printer files are queued for system output under the submitting userid, the submitting user can change, purge, or reorder these files prior to processing on the system printer.

To purge a job executing under the Batch Monitor, follow the procedure below:

1. Signal attention and enter the virtual machine environment.
2. Enter the HX (halt execution% immediate command.
3. Disconnect the virtual machine using the CP DISCONN command.

The HX command causes the Batch Facility to abnormally terminate. This provides the user with an error message and a CP dump of the Batch Facility virtual machine. The Batch Monitor then loads itself again and starts the next job (if any).

To purge an individual input spool file that is not yet executing, issue the CP PURGE command:

```
PURGE READER spoolid
```

In the previous format, spoolid is the spool file number of the job to be purged from the batch virtual machine's job queue. For example, the statement:

```
PURGE READER 123
```

would purge 123 from the batch virtual machine's job queue.

To reorder individual spool files in the Batch Facility's job queue, use the CP ORDER command:

```
ORDER READER spoolid1 spoolid2...
```

In this format, spoolid1 and spoolid2 is the assigned spool file identification of the jobs to be reordered.

The operator can determine which jobs are in the queue by using the CP QUERY command:

```
QUERY READER ALL
```

This QUERY command lists the filenames and filetypes of all the jobs in the batch virtual machine's job queue. The operator can then reorder them, using the ORDER command.

STOPPING THE BATCH VIRTUAL MACHINE

When you want to stop batch virtual machine execution after completion of the current job, you can issue the HB immediate command and press the attention key or equivalent to cause an attention interruption at any time during the job. This causes the batch virtual machine to be logged out at job completion.

When Batch Facility virtual machine execution is to be stopped immediately, but current files must be saved, you can use the CP SPOOL command, in the form SPOOL READER HOLD, and then issue the CP LOGOFF command.

The HOLD option causes CP to retain the virtual machine's current card reader file, so that when the Batch Facility is logged on again, execution resumes at the beginning of the held reader file.

If an emergency shutdown should occur, all jobs in the batch reader and all spool files are saved.

SUPPORT OF THE 3800 PRINTER AS A REAL VM/370 SPOOLING DEVICE

Virtual machines can print spool files on the 3800 printer. However, the use of more than one translate table for printing a given file notes an exception. Once the user starts to print a spool file with specific character sets, the entire file must be printed with these character sets. At any time, the user may change the characteristics of a spool file <u>before</u> it actually starts printing.

Appendix: Summary of CP Commands

This section is provided as a quick-reference summary of the VM/370 CP commands. Additional information concerning the commands can be obtained from the _VM/370 CP Command Reference for General Users_.

Be aware that all commands are not available to every virtual machine user. Rather, the commands belong to different classes that represent different areas of computer activity.

The class of commands that a user can issue is contained with the user's identification code in the system directory. The system directory is controlled by the system operations personnel.

The CP commands can be arranged in five categories:

- Program analysis and monitor functions.
- Device control functions.
- Virtual console functions.
- Spooling and file control functions.
- Miscellaneous functions (those commands that do not belong in any of the above categories).

Many command names and operands can be truncated. Where truncation is permitted, the shortest acceptable version of the command is represented by uppercase letters, with the optional part represented by lowercase letters. (Remember, however, that any VM/370 command can be entered with any mixture of uppercase and lowercase letters.

PROGRAM ANALYSIS AND MONITOR FUNCTIONS

Command	Explanation
ACNT	Render accounting information for and to the virtual user (Class A).
ADSTOP	Specify an instruction address location where the user's program is to be halted (Class G).
DCP	Display real processor storage locations (Class E).
Display	Display virtual console registers, plus the current CAW, CSW, and PSW (Class G).
DMCP	Dump the real storage locations to a user's virtual printer (Class E).
DUmp	Dump virtual machine registers and storage to the virtual printer (Class G).
LOCate	Provide the starting location of the CP control blocks associated with a user or his virtual (or real) devices (Class E).

Command Explanation

MONitor Initiate or terminate the recording of events that occur in the real machine (Classes A and E).

SAVESYS Provide a storage copy of virtual machine storage, registers, and PSW contents as they currently exist (Class E).

STCP Change the contents of real processor storage (Class C).

TRace Trace virtual machine activity at the terminal, spooled printer, or both (Class G).

DEVICE CONTROL FUNCTIONS

Command Explanation

ATTach Attach a real device for VM/370 control (Class B).

COUPLE Connect virtual channel to channel adapters (Class G).

| DEFine Alter the device feature of a 3330V (Class B).

DETach Remove a real or virtual device or channel from a virtual machine or the real system (Class B and G).

DISAble Disable direct or switched communication lines from the VM/370 system (Class A or B).

DISConn Disconnect the user's terminal from his virtual machine (Class Any).

ENable Connect specified communication lines to the system. (Class B).

Link Provide access to a specified DASD device by a user's virtual machine (Class G).

LOADBUF Load the 1403 universal character set (UCS) buffer with a specified train image. This command also has options that control a 3211 printer (Class D).

LOADVFCB Load the virtual forms control buffer for a virtual spooled 3211 printer (Class G).

NETwork Allow the enabling and disabling of all or specific 3704/3705 resources. In addition, the Network command is used to vary resources offline or online (Class A and B).

VARY Allow or disallow the availability of a device to a virtual machine or the VM/370 control program (Class B).

VIRTUAL CONSOLE FUNCTIONS

Command Explanation

Begin Continue or resume execution of the virtual machine at a specified storage location or at the address specified in the current PSW (Class G).

Command	Explanation
EXTernal	Simulate an external interruption condition for a virtual machine and return control to that machine (Class G).
Ipl	Initiate a program load for a virtual machine (Class G).
RESET	Clear all pending interruptions on a specified device (Class G).
SPool	Allow the starting or stopping of virtual console spooling as well as the alteration of other spooling control options (Class G).
STore	Alter registers and storage of a virtual machine (Class G).
SYStem	Initiate the virtual VM/370 console functions of clear, System Reset, and PSW Restart (Class G).
TERMinal	Redefine the user's terminal input characteristics and attention interruption handling capabilities (Class G).

SPOOLING AND FILE CONTROL FUNCTIONS

Command	Explanation
BAckspac	Restart or backspace the output of a unit record spooling device (Class D).
CHange	Alter one or more attributes of a closed spooled file (Class D and G).
Close	Terminate a spooling operation on a virtual console, reader, printer, or punch (Class G).
DRain	Stop spooling activity on the specified device after the current file in operation reaches termination (Class D).
Flush	Eliminate and halt the current file output on the specified real unit record device (Class D).
FRee	Release spool HOLD status from previously held files for real RDR, PUN, and PTR devices (Class D).
HOld	Defer file output to the real RDR, PTR, and PUN devices (Class D).
ORDer	Redefine the order of closed spool files (Class D and G).
PUrge	Remove spooled file(s) before reading, punching, or printing (Class D and G).
REPeat	Add to the number of copies of an output printer or punch file on a real unit-record device (Class D).
SPAce	Force single space on the printer, regardless of carriage control codes contained in the file (Class D).
SPool	Change spooling control options on a given device or group of devices and also starting or stopping of console spooling (Class G).

Command Explanation

STart Start a unit record device, or restart a drained device, or restart and change output classes that it may service (Class D).

TRANsfer Direct an input spool file to or retrieve it from a specified user's virtual card reader (Class G).

MISCELLANEOUS FUNCTIONS

Command Explanation

* Provide a means of inserting commentary in the stream of virtual console input and output messages (Class Any).

#CP Allow CP command execution while remaining in the virtual machine environment without having to signal attention (Class Any).

ATTN Create an interruption pending from the virtual console (Class G).

AUTOLOG Provides a means to automatically log on and load any virtual machine with IPL defined in its directory (Class A or B).

CP Allow the user to key in CP commands following "CP" while in the CMS or virtual machine environment (Class Any).

DIAL Logically attach a user terminal to a multi-access virtual system (Class Any).

DEFine Reconfigure one's own virtual machine (Class G).

ECho A terminal test facility that allows keyboard entered data to be displayed back at the user's terminal (Class G).

FORCE Force the logoff of any virtual machine (Class A).

HALT Stop any active channel program on a specified device (Class A).

INDicate Displays use of and contention for major system resources (Class E and G).

Logon
Login Gain access to a virtual system from a user terminal (Class Any).

LOGoff
LOGout Terminate user activity on a virtual machine (Class Any).

LOCK Lock a user's pages in processor storage (Class A).

Message
Msg Send a specified message from one virtual machine to another (Class A, B, or Any).

MONitor Records system counters needed for performance measurement (Class A and E).

| MSGNOH Send a specified message, without the standard MESSAGE
| header, from one virtual machine to another (Class B).

Command	Explanation
NETwork	Provide loading, dumping, tracing, and other functions for the 3704/3705 Communications Controller.
NOTReady	Simulate the not-ready status on a virtual spooled unit record or virtual console device (Class G).
Query	Provide status information on the real or virtual machine, and miscellaneous CP functions (Class A, B, C, D, E, F, and G).
Ready	Simulate a ready (device end interruption) condition for a specified virtual device (Class G).
REQuest	Makes an interruption pending from the virtual console (Class G).
REWind	Rewind a tape to the load point on a real device attached to a user's virtual machine (Class G).
Set	Establish system parameters for virtual and real machines, as well as other VM/370 values (Class A, B, C, E, F, and G).
SHUTDOWN	Terminate VM/370 activity in an orderly manner and checkpoint the system (Class A).
SLeep	Place the virtual machine in a dormant state, with the keyboard locked (Class Any).
UNLOCK	Release previously locked page frames of real storage (Class A).
Warning Wng	Transmit high priority messages or warnings to a user or all users (Class A or B).

Index

Trim Along This Line

READER'S COMMENT FORM

Title: IBM Virtual Machine Facility/370: Operator's Guide

Order No. GC20-1806-9

Please check or fill in the items; adding explanations/comments in the space provided.

Which of the following terms best describes your job?

□ Customer Engineer
□ Engineer
□ Instructor
□ Manager
□ Mathematician
□ Operator
□ Programmer
□ Sales Representative
□ Student/Trainee
□ Systems Analyst
□ Systems Engineer
□ Other (explain below)

How did you use this publication?

□ Introductory text
□ Reference manual
□ Student/ □ Instructor text
□ Other (explain) ______

Did you find the material easy to read and understand? □ Yes □ No (explain below)

Did you find the material organized for convenient use? □ Yes □ No (explain below)

Specific criticisms (explain below)

Clarifications on pages ______
Additions on pages ______
Deletions on pages ______
Errors on pages ______

Explanations and other comments:

Thank you for your cooperation. No postage necessary if mailed in the U.S.A.

Reader's Comment Form

Cut or Fold Along Line

Fold and tape Please Do Not Staple Fold and tape

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE:

International Business Machines Corporation
Department D58, Building 706-2
PO Box 390
Poughkeepsie, New York 12602

Attn: VM/370 Publications

Fold and tape Please Do Not Staple Fold and tape

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

IBM VM/370 Operator's Guide Printed in U.S.A. GC20-1806-9

IBM Technical Newsletter

This Newsletter No.	GN25-0834
Date	April 1, 1981
Base Publication No.	GC20-1806-9
File No.	S370-40 (VM/370 Release 6 PLC 17)
Prerequisite Newsletters/ Supplements	GN25-0491

IBM Virtual Machine Facility/370: Operator's Guide

This Technical Newsletter contains replacement pages ror VM/370 Operator's Guide to support Release 6 PLC 17 of IBM Virtual Machine Facility/370.

Before inserting any of the attached pages into the VM/370 Operator's Guide read carefully the instructions on this cover. They indicate when and how you should insert pages.

Pages to be Removed	Attached Pages to be Inserted*
Title, Edition Notice	Title, Edition Notice
Contents vii-x	Contents vii-x
Summary of Amendments xi-xiv	Summary of Amendments xi-xiv
17-18	17-18.2
21-24	21-24
35-36	35-36.2
39-40	39-40
45-46	45-46
65-66	65-66
85-90	85-90
105-116	105-116
147-148	147-148
159-160	159-160
171-172	171-172
177-180	177-180
183-184	183-184
197-198	197-198
217-218	217-218.4
Reader's Comment forms	Reader's Comment forms

*If you are inserting pages from different Newsletters/Supplements and identical page numbers are involved, always use the pages with the latest date (shown in the slug at the top of the page). The page with the latest date contains the most complete information.

Changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

Summary of Amendments

This Technical Newsletter incorporates changes reflecting minor technical and editorial modifications.

Note: Please file this cover letter at the back of the publication to provide a record of chanqes.

IBM Corporation, Programming Publications, Department G60,
PO Box 6, Endicott, New York 13760

Printed in U.S.A.

GC20-1806-9

IBM VM/370 Operator's Guide Printed in U.S.A. GC20-1806-9

IBM

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