GC33-5378-4
File No. S370-40

Systems

Operator's Library DOS/VS Operating Procedures

Release 34

IBM

IBM Technical Newsletter

This Newsletter No.	GN33-9247
Date	November 28, 1977
Base Publication No.	GC33-5378-4
File No.	S370-40
Previous Newsletters	GN33-9229

Operator's Library:
DOS/VS Operating Procedures

This Technical Newsletter, a part of Release 34 of the IBM Disk Operating System/Virtual Storage, DOS/VS, provides replacement pages for your publication. These replacement pages remain in effect for subsequent DOS/VS releases unless specifically altered. Pages to be replaced are:

137, 138
155, 156
177, 178
199, 200

A technical change to the text or to an illustration is indicated by a vertical line to the left of the change.

Summary of Amendments

This Technical Newsletter contains changes pertaining to the 3895 Document Reader/Inscriber.

Note: *Please insert this page in your publication to provide a record of changes.*

IBM Laboratory, Programming Publications Department, Boeblingen, Germany

Printed in U.S.A.

Device Code	Actual IBM Device	Device Type
1419	1255 Magnetic Character Reader	MICR - Magnetic Ink Character Recognition Devices
1419	1259 Magnetic Character Reader	
1419	1419 Magnetic Character Reader	
1419P	1419 Dual Address Adapter Primary Control Unit	
1419S	1419 Dual Address Adapter Secondary Control Unit	
3895	3895 Document Reader/Inscriber	Inscriber
1017	1017 Paper Tape Reader with 2826 Control Unit Model 1	Paper Tape Readers
2671	2671 Paper Tape Reader	
1018	1018 Paper Tape Punch with 2826 Control Unit Model 1	Paper Tape Punch
1419	1270 Optical Reader Sorter	Optical Readers
1419P	1275 Optical Reader Sorter Primary Control Unit	
1419S	1275 Optical Reader Sorter Secondary Control Unit	
1287	1287 Optical Reader	
1288	1288 Optical Page Reader	
3881	3881 Optical Mark Reader (cannot be used as a SYSIN device)	
3886	3886 Optical Character Reader	
2260	2260 Display Station or 1053 Printer	Display Station or Printer
3277	3277 Display Station or 3284/86 Printer	
3277B	3277 Display Station or 3284/86 Printer running in burst mode on the MPX channel	
2701	2701 Data Adapter Unit	Teleprocessing Lines
2702	2702 Transmission Control Unit	
2703	2703 Transmission Control Unit	
2703	Integrated Communications Adapter (Models 125 and 135)	
2703	3705 Communications Controller in Emulation Mode	
3704	3704 Communications Controller in Network Control Mode	
3705-1	3705 Communications Controller in Network Control Mode with Type 1 Adapter	
3705-2	3705 Communications Controller in Network Mode with Type 2 Adapter	
3791L	3791 Local Communications Controller	
2955	2955 Data Adapter Unit	Data Link for RETAIN
7770	7770 Audio Response Unit	Audio Response Unit
UNSP	Unsupported Device	Note: No burst mode on multiplexor channel
UNSPB	Unsupported Device	Burst mode on multiplexor channel

Figure IV-3. Device codes used in the ADD command (Part 2 of 2)

ALLOC
allocate virtual storage

The ALLOC command allocates storage to the virtual foreground partitions (or modifies the amount of storage allocated to the virtual foreground partitions). All of the virtual address area not allocated to virtual foreground partitions is automatically part of the virtual background partition.

See also Procedure 14

If your system uses a shared virtual area (SVA), only the address space in the virtual address area not allocated to the shared virtual area (SVA), can be allocated to foreground partitions with this command.

Operation	Operands	Type
ALLOC	Fn=mK[,Fn=mK]...	AR, JC

n indicates the partition to which storage is to be allocated, m specifies the amount of storage. The minimum amount of storage allocated to an active (batched) virtual foreground partition is 64K. The value m should be an even number; any odd specification (greater than 64K) is rounded up to the nearest even integer.

A non-active foreground partition may be allocated 0K bytes of virtual storage, for example, if you want to add its size to another partition. The size of the virtual background partition cannot be reduced below 64K.

The ALLOC command must have at least one operand; it may have as many operands as there are foreground partitions in your system. The operands can be specified in any order. The first operand must not be preceded by a comma.

Caution: Modifying the size of active partitions may lead to errors, either in the same or in other partitions. You should therefore obtain precise instructions before issuing the ALLOC command. It is further recommended to UNBATCH all affected partitions before issuing the ALLOC command. Refer to *DOS/VS System Control Statements*, GC33-5376, if you wish to modify active virtual partitions.

addr,addr — specifies the start and end addresses of the storage area to be dumped. The contents of the general registers that are associated with the specified storage area are also printed.

If the specified addresses are within an invalid address area, the command is ignored and a message to this effect is issued.

If the storage area crosses the boundary between a valid and an invalid address space, only the contents of the specified valid address space are dumped, together with the general registers that are associated with the valid address area. A message to this effect is issued.

If the storage area to be dumped crosses partition boundaries, the specified storage area is dumped, together with the general registers that belong to the partition in which the starting address is located.

partition — indicates the partition to which SYSLST is assigned for output of the dump. If it is omitted, SYSLST assigned to the background partition is used.

DVCDN
device down

The DVCDN (Device Down) command serves to inform the system that a device is no longer available for system operation. It is used when a device is to be serviced or when a device becomes inoperative. For the 3895 Document Inscriber, issue DVCDN (or ROD) before taking the device offline, to record the predictor sense information on SYSREC.

If a standard or temporary assignment exists for the device, any logical units assigned to it are unassigned. The DVCDN command does not close files associated with logical units, and after the DVCDN command has been issued, files on a DASD or diskette unit cannot be closed or reassigned to another DASD or diskette unit. Therefore, if the unit is a DASD or diskette unit, first attempt to close any files associated with logical units currently assigned to the device, using the CLOSE command.

If an alternate assignment exists for the device, it is removed when the DVCDN command is issued.

When the device has become operative, you must first issue a DVCUP command before the device can again be put to use.

The DVCDN command makes use of supervisor services that prevent other operator communication during execution of this command.

Operation	Operands	Type
DVCDN	X'cuu'	JC

X'cuu' the channel and unit number (in hexadecimal) of the device to be made unavailable.

RESET
reset I/O assignments

See also Procedure 13

The RESET command resets temporary I/O assignments either to the last preceding permanent assignment, if any, made via an ASSGN command without the TEMP option or to the standard assignment made during system generation. If neither a permanent nor a standard assignment exists, the temporary assignment is reset to UA (unassigned).

Resetting is performed only for the temporary assignments within the partition for which the RESET is given.

Operation	Operands	Type
RESET	{ SYS PROG ALL SYSxxx }	JC

SYS — resets the temporary assignments of all system logical units of the partition to their permanent or standard assignment.

PROG — resets the temporary assignments of all programmer logical units of the partition to their permanent or standard assignment.

ALL — resets the temporary assignments of all logical units of the partition to their permanent or standard assignment.

SYSxxx — resets the temporary assignment of the specified logical unit to its permanent or standard assignment. SYSIN and SYSOUT are invalid for this operand.

ROD
record on demand

See also Procedure 8

The ROD command writes statistical information to the recorder file (SYSREC) that has been accumulated since the last IPL or since the last time the ROD command was given. The ROD command should be issued each time before the system is shut down or before you re-IPL the system. For the 3895 Document Inscriber, issue ROD (or DVCDN) before taking the device offline, to record the predictor sense information on SYSREC.

Operation	Operands	Type
ROD	blank	JC

The ROD command has no operand.

If your system uses the Reliability Data Extractor (RDE) and the ROD command is issued, the system issues the message on SYSLOG:

```
1I90A END OF DAY =
```

You may respond with Y (for YES) or N (for NO). Any other response causes an error message and the same message as before to be issued:

```
1I92I INVALID CODE
1I90A END OF DAY =
```

When the response is Y, the accumulated statistical information is added to the recorder file and an end-of-day (EOD) record is written.

When the response is N, the accumulated statistical information is added, but no EOD record is written.

Linkage Editor Control Statements

This page shows the general formats of the linkage editor control statements. It is unlikely that you will be concerned with these statements; full details are contained in *DOS/VS System Control Statements*, GC33-5376.

PHASE

The PHASE statement indicates the beginning of a phase.

Operation	Operands
PHASE	name,origin[,NOAUTO]

INCLUDE

The INCLUDE statement signals that an object module is to be included.

Operation	Operands
INCLUDE	[modulename][,(namelist)]

ENTRY

The ENTRY statement provides an optional transfer address for the first phase.

Operation	Operands
ENTRY	[entrypoint]

ACTION

The ACTION statement specifies options to be taken.

Operation	Operands
ACTION	{CLEAR,MAP,NOMAP,NOAUTO,CANCEL,BG,Fn,REL,NOREL}

IPL Reason Codes and ID-Codes

For details on how to enter these codes, see Procedure 8 in the section *Procedures*.

Reason Code	Comments
CE	IBM representative has control of the system for maintenance or testing purposes.
DF	Default.
EN	Environmental problems (power, overheating, for example) caused failure.
IE	IBM hardware or IBM-supplied program error that did not require an IBM representative.
IM	IBM hardware or IBM-supplied program error that required an IBM representative.
ME	Media - hardware error caused by faulty disk pack, reel of tape, cards, and so on.
NM	Normal IPL.
OP	Operational problem - operator error or procedural problem.
UN	Unknown - undetermined error.
UP	A user (non-IBM-supplied) program caused the failure.

Always use ID code 00 with reason codes DF, EN, NM, OP, UN, and UP.

ID Code	Comments
00	Unknown - Must be used with Reason Codes DF, EN, NM, OP, UN, and UP. 00 is the default.
10	Processor - CPU, channel (integrated), storage unit, and so on, failure.
20	DASD - A failure occurred in a DASD unit or its associated control unit (such as, 2311, 2314, 2841, 3330, 3340, 3350).
30	Other - A device without an ID code (such as a paper tape unit) caused the failure.
40	Magnetic Tape - A failure occurred in a magnetic tape unit or its associated control unit (such as, 2401, 2803, 3400, or 3540 Diskette).
50	A failure occurred in a card reader/punch, a printer, or in its associated control unit (such as, 2540, 1403, 2821).
60	MICR/OCR - A magnetic ink character reader (such as, 1417, 1419) or an optical character reader (such as, 1285, 1287) failure.
61	Inscriber - A 3895 document reader/inscriber failure.
70	A teleprocessing failure occurred in a teleprocessing control unit (such as, 2701, 2702).
80	Graphic - A video display unit (such as, 2260) or its associated control unit failure.
90	An IBM-supplied program (such as the DOS/VS system or one of its components) failure.
91	An IBM Programming Product failure.

GC33-5378-4
File No. S370-40

Systems

Operator's Library DOS/VS Operating Procedures

Release 34

IBM

Summary of Amendments

Release 34

Edition GC33-5378-4 documents:

- Full support of
  - IBM 3350 Direct-Access Storage (DOS/VS previously supported the device only in 3330-1 compatibility mode).
  - IBM 3330-11
- Support for dynamically changing the blocking factor for a sequential disk file on an IBM 3350 or 3330-11 through the use of a BLKSIZE parameter in the DLBL job control statement.
- Common device class for 3211-compatible printers (IBM 3211 and IBM 3203-4).
- Support of IBM 3277 Display Station as operator console.
- Support of IBM 3540 Diskette Unit as an IPL communication device.
- Improvement of initial program load through the use of an IPL communication device list.
- Integration of support information on System/370 CPUs Model 135-3, 138, 145-3, and 148 and on the IBM 3203-4 printer.
- Inclusion of the List System History utility program.
- Deletion of the Automatic Condense function

The information on POWER/VS has been removed from this edition; it is now contained in *DOS/VS POWER/VS Installation Guide and Reference*, GC33-6408, and *DOS/VS POWER/VS Workstation User's Guide*, GC33-6049.

Release 33

Edition GC33-5378-3 documents:

- Cardless system support
- Second label information cylinder for the IBM 3340
- POWER/VS enhancements
- System enhancements
- Installation improvements

In addition, technical corrections and editorial changes have been made throughout the manual.

Note: Installations using DOS/VS POWER - up to and including Release 30 - should retain the suffix-1 edition of this manual.

Fifth Edition (April 1977)

This is a major revision of, and obsoletes GC33-5378-3.

This edition applies to Version 5, Release 34, of the IBM Disk Operating System/Virtual Storage, DOS/VS, and to all subsequent versions and releases until otherwise indicated in new editions or Technical Newsletters.

Changes are continually made to the information herein; before using this publication in connection with the operation of IBM systems, consult the *IBM System/370 Bibliography, GC20-0001,* for the editions that are applicable and current.

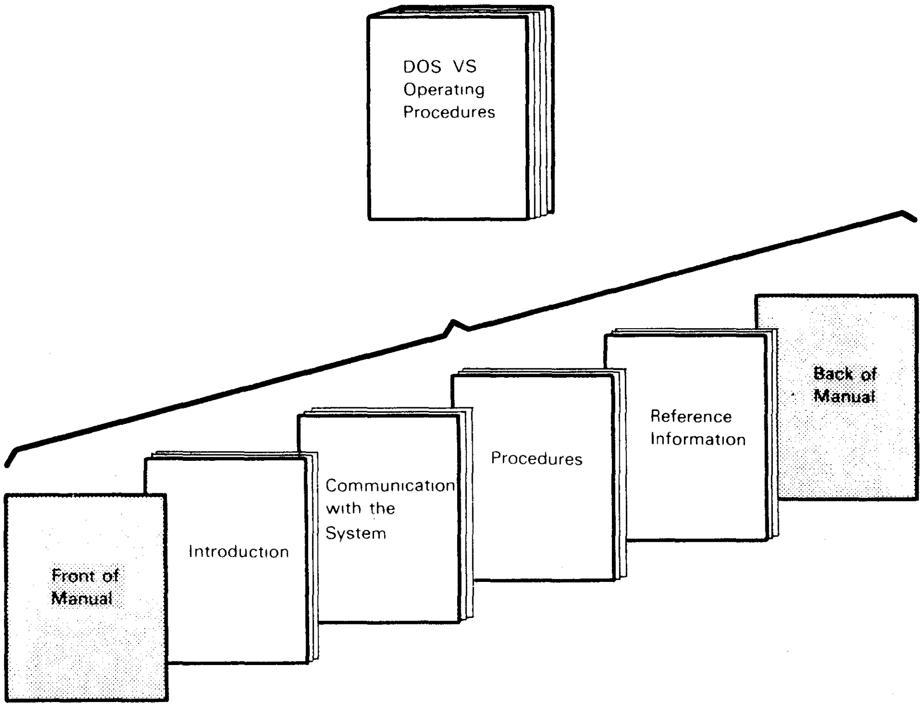
Requests for copies of IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

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 Laboratory, Publications Department, Schoenaicher Strasse 220, 7030 Boeblingen, Germany. Comments become the property of IBM.

Preface

This manual contains the information required by system operators to run jobs under the Disk Operating System/Virtual Storage (DOS/VS). It is intended primarily for system operators who have acquired a basic knowledge of data processing and computer equipment. For a system using DOS/VS Virtual Telecommunications Access Method (VTAM), the operator will need to refer to *DOS/VS VTAM Network Operating Procedures*, GC27-0025.

The manual is divided into four main parts, preceded by front and followed by back matter.



Front of Manual

Front of Manual consists of this preface, the table of contents, instructions on how to use the manual, and a list of abbreviations.

Introduction

Introduction familiarizes the operator with the major concepts and components of DOS/VS. This brief summary provides the basis that is required to understand the information contained in the subsequent sections of the manual.

Communication with the System

Communication with the System describes the forms of system-to-operator and operator-to-system communication, as well as the devices that are used for communication.

Procedures

Procedures tells the operator how to execute jobs under control of DOS/VS. The procedures are illustrated by examples.

Reference Information

Reference Information contains useful information for quick reference. The material in this section is organized in such a way that the desired information can be found easily and quickly.

Back of Manual

Back of Manual contains a glossary, a bibliography, and the index.

The bibliography contains a list of DOS/VS manuals, which enable you to obtain detailed information concerning individual topics. References in this book to other DOS/VS manuals use the abbreviated titles given in the bibliography.

For convenience, each section or major topic begins on a righthand page. The lefthand pages that this practice sometimes leaves blank remain un-numbered.

Contents

Contents

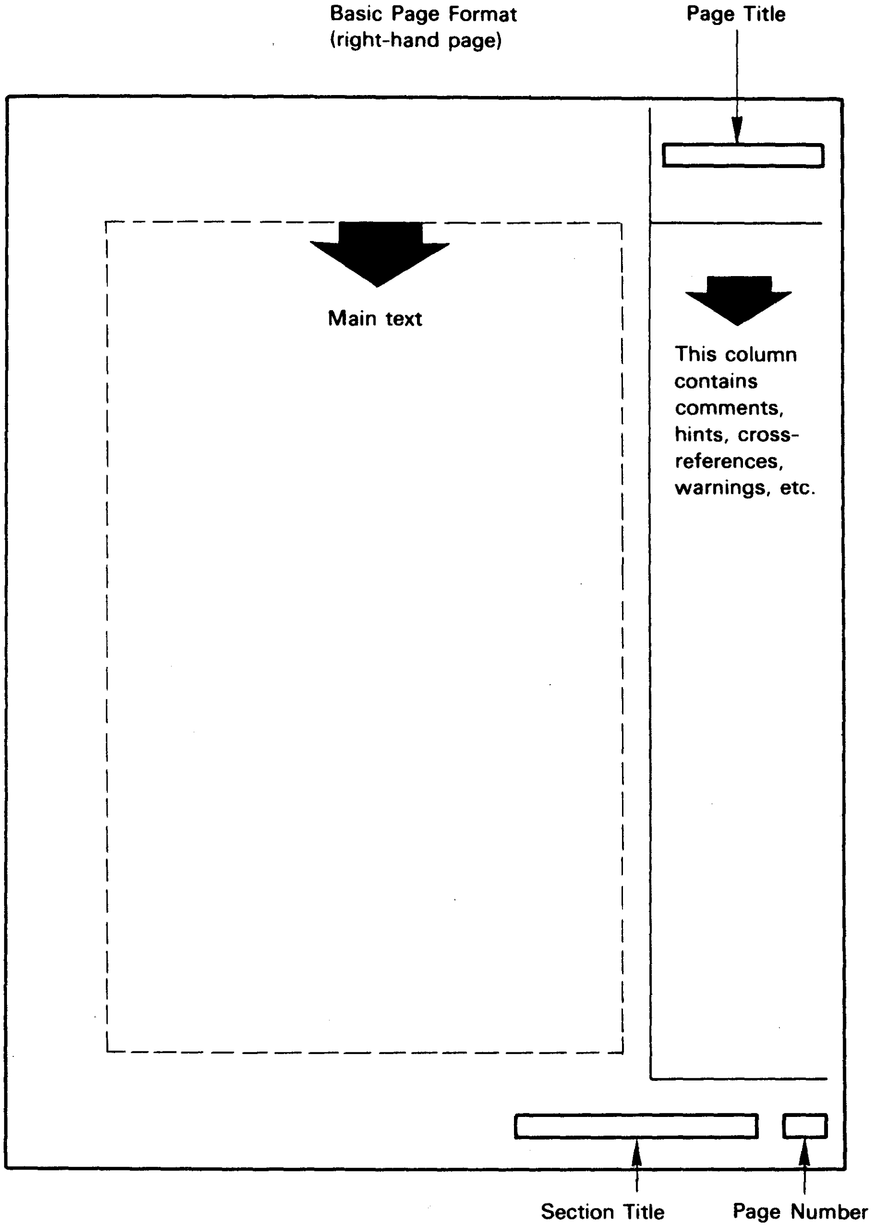
(continued)

List of Figures

How to use this Manual

If you are using this manual for the first time it is advisable to read the preface, which explains the organization of the manual.

An attempt has been made to devote a separate page to each topic, although some subjects had to be spread over several pages.



Abbreviations

CPU	central processing unit
DASD	direct access storage device
DOC	display operator console
DOS/VS	Disk Operating System/Virtual Storage
EBCDIC	extended binary coded decimal interchange code
ECC	error correction code
EFL	error frequency limit
EOJ	end of job
EREP	environmental recording, editing, and printing
FCB	forms control buffer
HIR	hardware instruction retry
IMPL	initial micro-program loader
I/O	input/output
IPL	initial program loader
POWER/VS	Priority Output Writers, Execution processors, and input Readers/Virtual Storage
RDE	reliability data extractor
RJE	remote job entry
RMSR	recovery management support recording
RPS	rotational position sensing
SDL	system directory list
SVA	shared virtual area
TOD	time of day
UCB	universal character set buffer
VSAM	virtual storage access method
VTAM	virtual telecommunications access method
VTOC	volume table of contents

Section 1

This section familiarizes you with the major concepts and components of DOS/VS. It provides the basis that is required to understand the information contained in the subsequent sections of this manual.

The Operator's Responsibilities

A data-processing installation exists for one purpose only: to process data. To fulfill this purpose economically and efficiently, all the equipment at your installation should be kept as busy as possible.

Regardless of its size and complexity, a data-processing installation performs three major functions: it accepts input, processes this input, and produces output. You, as the operator, are directly concerned with the operations that allow the installation's computing system to perform these functions efficiently. The following discussion outlines some of these operations.

Remember, however, that the exact nature and extent of your responsibilities will depend on the organization at your installation. You should therefore regard this discussion as a general introduction to the work of an operator, and not assume that all the duties outlined here will be expected of you.

Preparing the system

Before the system can start performing any of its functions, you have to prepare it for operation. You must, for example, switch on the power, ready devices, and perform other operations in accordance with a set procedure. Once you have prepared the system, it is ready to accept input.

Input

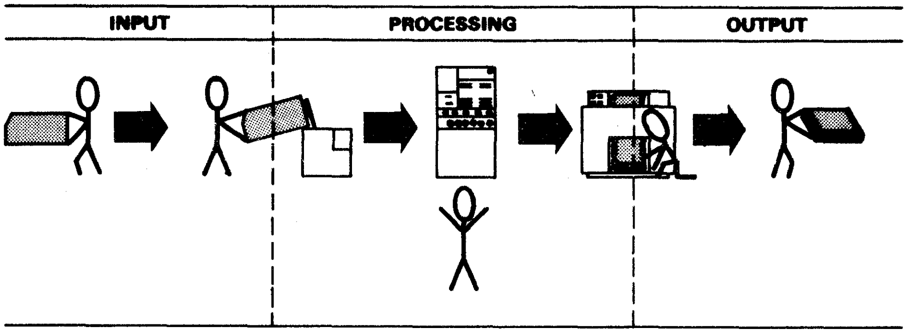
The input usually consists of programs, data, and control information and is normally on punched cards, on tape, on disk, or on diskette. In most cases the input is supplied by the programmer. Sometimes, however, you may have to correct errors in control statements or prepare such statements yourself. Also, certain operations are necessary to make your installation accept the input. Input in the form of punched cards, for example, is read by a card reader. This means that you must place the cards in the reader and ready the device.

Processing

The execution of a program is called processing. Processing takes place in the Central Processing Unit (CPU). Although you are not concerned with the actual processing, you may have to take specific action so as to enable the CPU to continue.

Output

The output may be in the form of a printed report, of punched cards or paper tape, or it may be on diskettes, or on magnetic tape or disk.



While your jobs are running, there are several things you may have to attend to. You may have to place more cards in the hopper of the card reader, or mount tape reels or disk packs. For printed output you may have to reload either a forms control buffer (FCB), which controls the skipping of forms, or a universal character set buffer (UCB), which controls printing with a universal character set print chain (or print train). You may also have to respond to messages that are printed on the printer keyboard or that appear on the screen.

Error handling

Errors, too, may occur during each of the three stages. Cards may get jammed in the card reader, there may be errors in the control statements supplied by the programmer, or one of the devices may break down. You may have mounted a wrong disk pack or tape drive reel for a program. Programs or data that are required for processing may not be found on the specified disk packs or diskettes. Although you can recover from a number of errors by taking specific action, it is sometimes impossible to continue processing. In such cases it is essential that you collect as much information as possible on the status of the program and the system at the time when the error occurred. This helps the programmer or the engineer to analyze and solve the problem.

Restarting a job from a checkpoint

When a programmer writes a program that is expected to run for a long time, he usually ensures that it will not have to be rerun from the start if execution ends abnormally. He arranges for the program to record the status of the job and the system at set intervals on disk or tape. If abnormal termination requires the program to be run again, it may be restarted from one of these checkpoints.

Each time a checkpoint is written during the execution of a program, the following message appears on SYSLOG:

```
0C00I  CHKPT nnnn TAKEN ON SYSxxx=cuu
```

nnnn — is the checkpoint number, which is increased by one each time a checkpoint is taken.

SYSxxx=cuu — indicates the logical unit and physical device on which the checkpoints have been stored.

No action is required for these messages. However, be sure to save the messages because the checkpoint numbers are needed for restarting by the programmer.

If a checkpointed program terminates abnormally, inform your programmer, who will help you to analyze the cause of the failure and give you instructions on how to restart the program from one of the checkpoints.

The restart procedure is described in *DOS/VS System Management Guide*, GC33-5371.

Operating a teleprocessing network

If your installation is running a teleprocessing network under DOS/VS VTAM, you will be involved in additional duties as the network operator. For details see *DOS/VS VTAM Network Operating Procedures*, GC27-6997.

Shutting down the system

The customer's requirements will normally determine if and when the system will be shut down. Occasionally, a shutdown will be necessary for maintenance purposes. The shutdown procedure depends on the size and complexity of your installation. You may, for example, have to see to it that statistical data, accumulated during the day, is recorded or printed for subsequent use by the system programmer or data processing analyst.

Operating systems

The preceding paragraphs, although simplified, apply to all electronic data-processing installations. The work performed by all modern installations is controlled by a set of programs, which are called the operating system. The functions of an operator vary from installation to installation and depend on the operating system that is used and on the kind of work that is to be done.

You are concerned with DOS/VS and, consequently, with the functions that are typical for this system. To help you understand why you should perform certain operations, you should be familiar with the concepts of DOS/VS, which are discussed in the next chapter.

Concepts and Components of DOS/VS

In the early days of electronic data processing, when computer installations were less sophisticated, management of the installation's resources (such as CPU, input/output devices, programs,) was fairly easy. However, as installations grew in size and complexity, a way of simplifying resource management became necessary. Special programs were written to perform operations that occur frequently, and these programs were combined to form a so-called operating system. Because the operating system you are concerned with resides on disk and uses the virtual storage concept, it is called the Disk Operating System/Virtual Storage, or briefly DOS/VS. (The concept of virtual storage is described later in this manual.)

DOS/VS can be divided into two main groups of programs: control programs nd processing programs, which are further broken down as shown in Figure I-1. All these programs are contained in libraries - disk areas that are set aside for them.

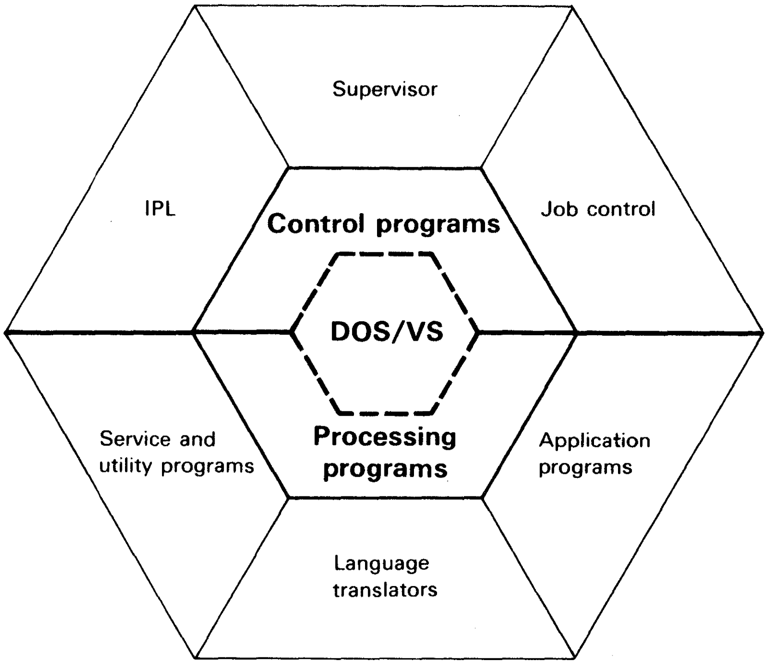


Figure I-1. DOS/VS consists of control programs and processing programs

The following pages describe these two types of programs, and also various concepts and components of DOS/VS that will concern you in your work as an operator.

Control Programs

The control programs ensure proper and efficient operation of the entire system. The three control programs are:

- Initial Program Loader
- Supervisor
- Job Control

Initial Program Loader (IPL)

The DOS/VS IPL program starts the system, which means that this program must be executed each time the system is started. Executing the IPL program - performing the IPL procedure, as it is often called - is one of your responsibilities as an operator.

Supervisor

The supervisor is the control program that supervises execution of all other programs. It coordinates the use of the system resources and maintains the flow of operations, thus taking over a considerable part of the burden which, otherwise, would have to be handled by the individual programs.

Although most of the work done by the supervisor is invisible to you, some of its functions are worth remembering: it loads programs, manages and controls storage and CPU time, and handles much of the communication between system and operator.

When you have performed the IPL procedure, part of the supervisor will always be in storage. The remainder of the supervisor comprises parts that are less frequently used; they are called transient routines. Transient routines are kept on the disk pack on which the system resides, from which they are automatically fetched into the so-called transient area in storage when they are required.

Job Control Program

The job control program is loaded into storage each time a new program must be prepared for execution. This means that it is loaded for the first time after IPL to prepare execution of the first program. Subsequently it is loaded each time a program has completed execution or is terminated. This makes transition from executing one program to executing the next one automatic.

The job control program has many different functions. Not all of them are needed for every program execution. The programmer or - sometimes - the operator must therefore inform the job control program of the specific requirements for each program. Such instructions to the job control program, which have a prescribed form, are called job control statements and job control commands. The difference between these two types of job control information is explained later in this manual. The entire set of job control statements and job control commands is called the job control language.

Job control statements/commands

Job control language

The job control language is one of your chief means of telling the system what you want it to do. The discussion of the job control language therefore plays an important part in this manual.

Processing Programs

The processing programs comprise (1) language translators, (2) service and utility programs, and (3) problem programs, also called application programs. The first two types of processing program are supplied by IBM. They serve to make the application programs run under DOS/VS and provide service functions to make best use of the system.

Language Translators

Programming languages

Source program

Assembler/Compiler

DOS/VS permits the programmer to write his programs in several programming languages: COBOL, PL/I, FORTRAN, Assembler, and RPG. A program written in one of these languages is called a source program. Source programs must be translated into a language that can be understood by the computer. Translation of source programs is performed by special programs, called language translators. Each programming language has its own language translator. In the case of the assembler language, the language translator is called the assembler; the translators of the other programming languages are called compilers.

Object program
Object module

The translation of a source program results in an object program, which may consist of one or more object modules. The object program can be punched or written out, or stored (cataloged) in the library that contains object modules. Object modules cannot yet be executed. They must first be processed by the linkage editor (see below).

Service and System Utility Programs

The DOS/VS service programs consist of (1) the linkage editor program and (2) a set of programs called the librarian.

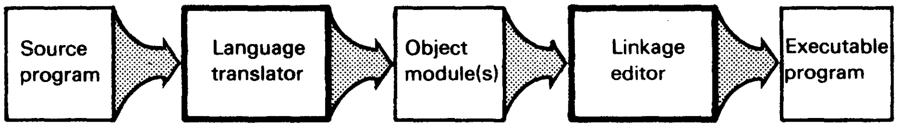
The DOS/VS system utility programs are IBM-supplied programs, designed to perform routine tasks.

Linkage Editor

Executable program

Phases

The linkage editor reads the output produced by a language translator (one or more object modules) and converts it to an executable program. This conversion is called link-editing. The linkage editor is further used to catalog executable programs, or parts of these programs (called phases) in the library that contains executable programs.



Librarian

The librarian consists of a number of programs that are used to maintain the DOS/VS libraries. The librarian performs such functions as:

- Displaying the contents of a library
- Adding elements (phases, modules, etc.) to a library
- Deleting elements from a library
- Condensing a library
- Creating a library and modifying its size

Utility Programs

The following IBM utility programs are designed to perform such everyday tasks as copying data from one volume to another and preparing volumes for the storage of data. Note, however, that the description of these programs given here is very general, and is not intended as a substitute for the information in *DOS/VS System Utilities*, GC33-5381. Your system programmer will advise you when you need to run these programs.

Assign Alternate Track Data Cell: This program analyzes a data-cell track that contains an error and assigns an alternate track to contain the data from the defective track.

Analysis Program-1 (AP-1): This program enables the operator or system programmer to determine if a data error on the IBM 3344 or 3350 Direct Access Storage device is a data module error or a device error. For information on AP-1, see *OS/VS and DOS/VS Analysis Program-1 (AP-1) User's Guide*, GC26-3855.

Assign Alternate Track Disk: This program analyzes a disk track that contains an error and assigns an alternate track to contain the data from the defective track.

Backup and Restore System: These programs generate on tape a device-independent backup copy of the system and/or private libraries and restore this copy to another DASD device.

Clear Disk and Data Cell: These programs clear and preformat one or more areas of a disk pack or a data cell, respectively.

Copy and Restore Disk or Data Cell: This program copies a file or volume from disk to cards, disk, or tape, or from a data cell to tape. It also restores data to disk or data cell.

Copy and Restore Diskette: This program copies the contents of a diskette to another diskette on a different device. It also restores data from one diskette to another diskette on the same device.

Copy File and Maintain Object Module: This multi-purpose program serves primarily as a file-to-file utility used to copy and list card image files. It may also be used to modify object modules, phases, and PTFs as well as to deblock PTF files.

Deblock: This program is used to block and deblock 3440 byte blocked records in IBM distribution files.

Fast Copy Disk Volume: This program copies at high speed the entire contents of certain disk storage devices to other devices of the same type, either directly or via intermediate tape storage.

Initialize Data Cell: This program prepares new or expired IBM 2321 Data Cells for use on a data cell drive.

Initialize Disk: This program prepares a disk pack or data module for use or changes the volume label(s) and the VTOC address of a previously initialized disk pack.

Initialize Tape: This program writes IBM standard tape volume labels on EBCDIC tapes to enable standard label checking to be carried out.

Maintain System History: This program is a tool for the application and removal of PTFs (Program Temporary Fixes) to DOS/VS Systems. It also maintains a history of all PTFs applied to the system.

List System History: This program provides a formatted printout of the system history.

Printlog: This program produces a printed copy of messages contained in the hard-copy file.

VTOC Display: This program displays the labels contained in the VTOC of a disk pack or a data cell on a printer, on tape, or on disk.

The two most important utility programs for the operator, VTOC Display and PRINTLOG, are described in Procedures 19 and 24 of this manual. For details of the other programs, see *DOS/VS System Utilities,* GC33-5381.

The DOS/VS Libraries

The DOS/VS libraries contain programs or program parts as well as control information, and their contents are available for use whenever they are required for processing. There are four types of library:

- Core image library
- Relocatable library
- Source statement library
- Procedure library

Core image library
Phases

The core image library contains program phases that have been processed by the linkage editor and are therefore ready to be executed. A phase can be an entire program or part of a program. Phases are placed into the core image library by the linkage editor.

Relocatable library
Object modules

The relocatable library contains the output from language translators: object modules. These object modules can be combined with other object modules and processed by the linkage editor to form a single executable program. The relocatable library is used mainly to store standard routines that are frequently used in larger programs. This makes it unnecessary for the programmer to code the same routine more than once for several programs.

Source statement library

Books

The source statement library contains source programs or parts of such programs (source statements or groups of source statements). The individual elements contained in a source statement library are called books. The source statement library is used to store routines or groups of statements that are frequently used in larger source programs.

Procedure library
procedures

The procedure library is used to store frequently used sets of job control and linkage editor control statements, as well as control statements for system utility and service programs.

To you, as an operator, the first three types of library are of interest only if you use the linkage editor and the librarian. The procedure library, however, is a useful tool to you, since it reduces the number of control statements contained in punched cards and, consequently, the amount of card handling. For cardless system users, the use of the procedure library reduces the number of control statements on diskette.

System/Private libraries

Libraries must reside on a magnetic disk device. The first three types of library are available as system libraries and as private libraries. The procedure library is available as a system library only.

All system libraries together form the DOS/VS system. They all reside on the same disk pack, which is referred to as the system residence pack or SYSRES (see Figure I-2).

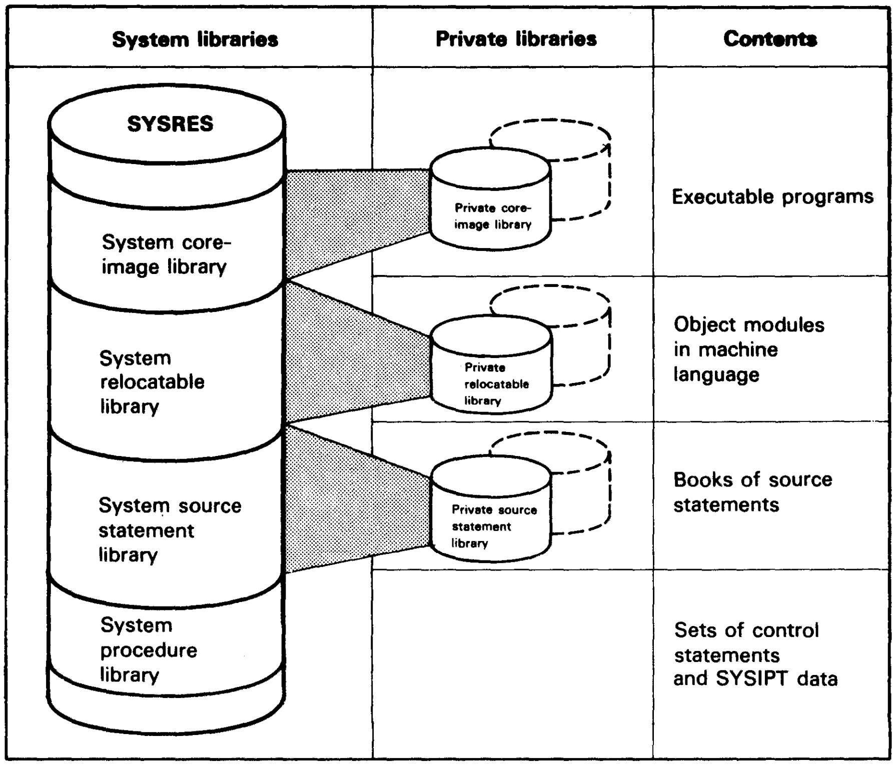
Library directories

Each system library is preceded by a directory, which contains the names of all elements in the respective library, and the disk addresses where they can be found.

The DOS/VS Libraries (continued)

System directory

The DOS/VS system as a whole is also preceded by a directory. This is called the system directory and indicates the location of all libraries and directories belonging to the system.



System libraries	Private libraries	Contents
SYSRES System core-image library	Private core-image library	Executable programs
System relocatable library	Private relocatable library	Object modules in machine language
System source statement library	Private source statement library	Books of source statements
System procedure library		Sets of control statements and SYSIPT data

Figure I-2. The libraries and their contents

Private libraries are extensions of the system libraries. They contain the same types of elements as the system libraries and are used for the same purpose. They, too, are preceded by directories.

Records and Files

The main purpose of this chapter is to introduce some terminology that is frequently used in data processing. In your job as an operator you will not be concerned with the structure of records and files (this is the programmer's responsibility), but you will often hear about them. Also, the terminology introduced in this chapter will help you to understand subsequent chapters of this manual.

A record is a collection of related items of data, treated as a unit. A file is a collection of related records, treated as a unit. These two terms are best explained by an example.

If you think of an invoice, one line of this invoice may be thought of as an item of data, a complete invoice may be thought of as a record, and the complete set of invoices may be thought of as a file.

The collection of libraries on the system residence pack, which together make up DOS/VS, also form a file.

Files can be stored on various media: on punched cards, on diskette, on paper tape, on magnetic tape or disk, and so on.

File maintenance

A very important operation in data processing is file maintenance, which is the activity of keeping files up-to-date by adding, changing, or deleting records in the file.

Volume

It is obvious that, if a file is to be updated, we must have a means of finding it. This is no problem when the file is stored in cards since it can easily be identified. Finding a specific file on disk, diskette, or magnetic tape is more complicated. Disk packs, diskettes, and magnetic tape reels - also called volumes - are normally labeled on the outside for easy identification. However, volumes frequently contain more than one file, and it is therefore necessary that each file on a volume is uniquely identified.

File labels

This unique identification is provided in the form of file labels. Whenever a programmer creates a file, he can specify the contents of the file label, which is then written onto the disk, diskette, or tape just like a data record. When the file is processed as input, the programmer specifies - in job control statements - the contents of the label so that certain routines can compare the specified data with the actual label. If the checking routines detect a mismatch between the actual label and the label information supplied by the programmer, an error message is issued. This is illustrated in Figure I-3.

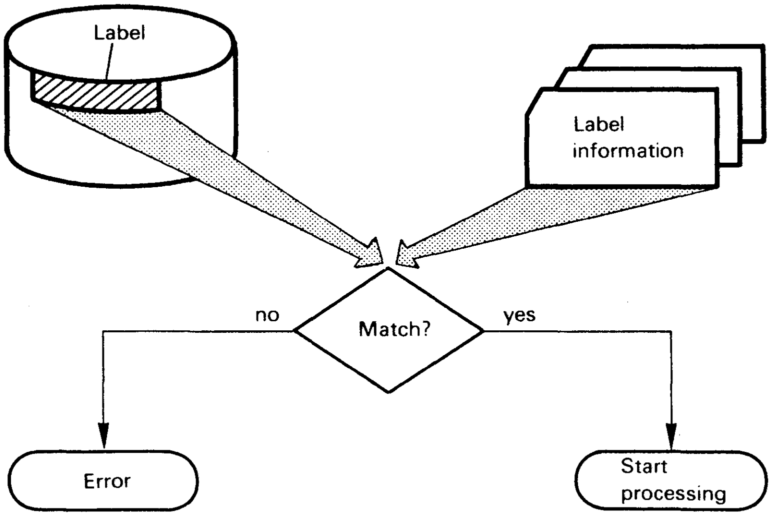


Figure I-3. **Label checking**

File protection

File labels also serve to protect files. They contain information regarding the date of creation and expiration of the file. A file must obviously not be destroyed, that is, overwritten by another file, before it has expired. The label information ensures that this cannot occur.

Volume label

In addition to a physical label - usually in the form of a sticker - each volume (disk, diskette, tape, or data cell) is further identified by the so-called volume label. This label is written by a special utility program when the volume is prepared for use for the first time. For disk and diskette files, labels are mandatory.

VTOC

The file labels of all files on a disk or diskette volume are grouped together in the VTOC (Volume Table of Contents) of that volume.

Tape labels

For tapes, labeling is optional. If a tape is to be labeled, a special utility program - the Initialize Tape Utility - is used to write a volume label on the tape. File labels are written preceding and following each file, using information supplied in job control statements.

On unlabeled tapes, specific files can be located by counting the number of tapemarks preceding the file. A job control command/statement, MTC, is used to position the tape correctly.

Physical and Logical Devices

Physical device address

All input/output devices (I/O devices, for short) are uniquely identified by a three-digit number. These numbers are shown on each I/O device (card reader, disk drive, magnetic tape unit, printer, and so on) and are called physical device addresses. These addresses are very important because the system must know from which device to read input and to which device to write output.

X'cuu'

In job control statements and commands, the physical device address is represented by X'cuu'.

X	indicates that the number is in hexadecimal notation.
c	indicates the channel to which the device is attached (channel address).
uu	indicates the number of the device (unit address).

Typical examples of physical device addresses are X'180', X'182' for magnetic tape units, X'190', X'191' for disk drives, X'00E' for a printer.

Logical unit
Symbolic device name

A programmer, however, does not use physical device addresses in his programs. When he writes his program, he wants to be independent of the actual devices that are eventually used for program execution, for a number of good reasons: the program may not always be executed in the same system, or the devices he needs may be used by a program that runs at the same time as his, or a device may be inoperative when the program must be executed. Programmers, therefore, use logical units, also called symbolic device names. Immediately before a program is executed, these logical units are associated, by means of job control statements or commands, with the physical devices to be used for this particular execution of the program.

System logical units

Programmer logical units

The logical units are divided into two classes: system logical units and programmer logical units. System logical units are used primarily by the control programs and by various IBM-supplied processing programs. The table in Figure I-4 lists all system logical units and indicates what they are used for. Programmer logical units are used primarily by problem programs. They run from SYS000 to SYSnnn. SYSnnn is the highest-numbered programmer logical unit available in your system.

Of the system logical units, user programs may also use SYSIPT and SYSRDR for input, SYSLST and SYSPCH for output, and SYSLOG for communication with the operator.

Two additional symbolic names, SYSIN and SYSOUT, are provided:

SYSIN **Can** be used instead of SYSRDR and SYSIPT when these are assigned to the same card reader or magnetic tape unit. It **must** be used when SYSRDR and SYSIPT are both assigned to disk or to a diskette.

Logical Unit	Device Type	Used for
SYSRDR	card reader, magnetic tape unit, diskette, or disk extent	reading job control statements or commands
SYSIPT	card reader, magnetic tape unit, diskette, or disk extent	input of system data, such as source statements for language translators, or control information for service programs.
SYSPCH	card punch, magnetic tape unit, diskette, or disk extent	punched output of the system
SYSLST	printer, magnetic tape unit, diskette, or disk extent	printed output of the system
SYSLOG	console printer keyboard or display console or printer*	communication between the system and the operator and logging of job control statements.
SYSLNK	disk extent	input to the linkage editor
SYSRES	disk extent	system residence device
SYSSLB	disk extent	private source statement library
SYSRLB	disk extent	private relocatable library
SYSCLB	disk extent	private core image library
SYSREC	disk extent	storing error records collected by the recovery management support recorder (RMSR) functions. For the Model 115 and 125, storing messages to the operator for subsequent printing.
SYSVIS	disk extent	virtual-storage page data set
SYSCAT	disk extent	VSAM catalog

* DOC (Display Operator Console) or 3277

Figure I-4. System logical units and their use

SYSOUT **Must** be used instead of SYSPCH and SYSLST when these are assigned to the same magnetic tape unit. It **cannot** be used to assign SYSPCH and SYSLST to disk or to a diskette because these two units must refer to separate extents if disk, or to separate devices if diskette.

SYSRES

Although you will soon become familiar with the various symbolic device names, it is good to remember at this stage that SYSRES is the symbolic name for the disk drive on which the system residence pack is mounted.

Standard assignments

When your system was generated, the logical units were assigned to certain physical devices. These assignments are called standard device assignments. You can change these standard assignments by means of the ASSGN command or the ASSGN statement, which are discussed later in this manual.

Permanent assignments

Temporary assignments

Device assignments can be either permanent (if made by a job control command) or temporary (if made by a job control statement). A permanent assignment remains in effect until it is overridden by a temporary assignment or until it is changed by a new permanent assignment. A temporary device assignment is valid for only one job or until it is overridden by another assignment.

Both temporary and permanent assignments can be specified in two ways. The programmer can include them in the job stream as job control statements, or you, the operator, can enter them directly from your console printer keyboard or display console.

Generic assignments

To assign a logical unit to a physical device, you only need to specify a generic name, for example DISK or TAPE. The system will then select an available device in that category and inform you of the assigned device address.

It is obvious that the system must have a means of keeping track of the assignments between physical devices and logical units. For this purpose the supervisor contains LUB and PUB tables.

LUB and PUB tables

The LUB table (LUB = logical unit block) contains all the logical units; the PUB table (PUB = physical unit block) contains the addresses of all the I/O devices at your installation. The ASSGN statement (or the ASSGN command) establishes the link between an entry in the LUB table and an entry in the PUB table when it assigns a logical unit to a physical device address.

I/O devices may be added to or removed from your installation. The addresses of such devices must be added to or deleted from the PUB table. The ADD and DEL (delete) operator commands enable you to effect such changes during the IPL procedure, which is described in the section *Procedures.*

Figure I-5 illustrates the link between the LUB and PUB tables and the physical devices.

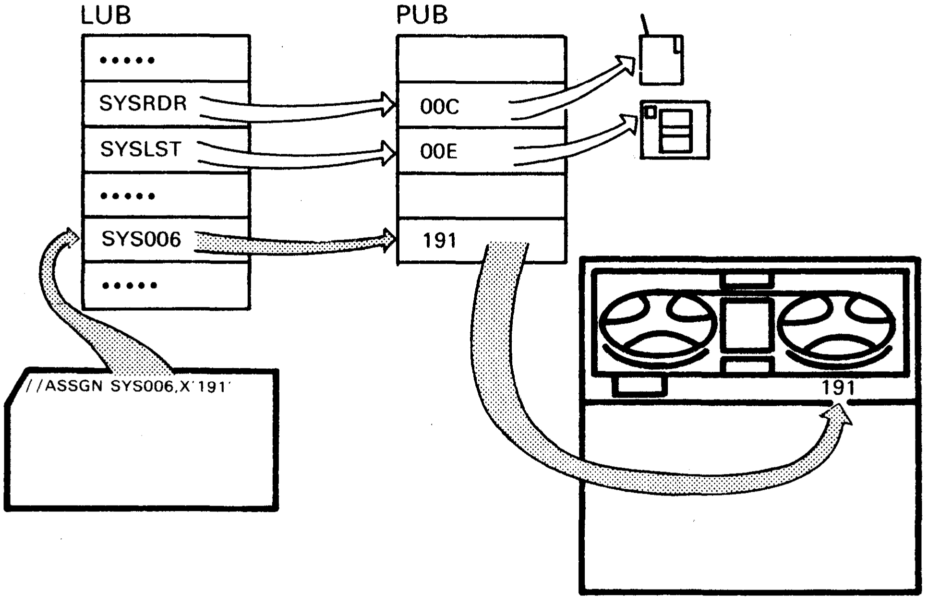


Figure I-5. The ASSGN statement establishes a link between SYS006 and tape drive 191

Jobs and Job Steps

A unit of work that is submitted to the system for processing is called a job. Each job is clearly defined by two statements:

```
// JOB name
    .
    . (other
      cards)
    .
/&
```

- job statement (// JOB name)
- end-of-job statement (/&)

The job statement is the first statement of a job, the end-of-job (EOJ) statement the last.

```
// JOB name
    .
    .
// EXEC PROGA
/&
```

The program to be executed in a job is invoked through the EXEC statement. In the example in the margin, the program PROGA is fetched from the core-image library and executed.

```
// JOB name
    .
    .
// EXEC PROGA
    .
    .
// EXEC PROGB
/&
```

One or more programs can be executed within a job; the execution of a single program within a job is called a job step. Therefore, each job can consist of one or more job steps. The example in the margin shows a job that comprises two job steps. Although the programmer is free to include as many job steps in a job as he wishes, it is not advisable to execute, in one job, several programs that are completely independent of one another. This is because, if one job step terminates abnormally, the job control program will ignore the remaining job steps up to the next end-of-job statement and the programs following the one that failed will not be executed.

A typical example of related job steps that should form one job is assembling, link-editing, and executing a program, where execution of one job step depends on successful completion of the preceding one.

Job stream

The job control program provides automatic job-to-job transition. This means that an unlimited number of jobs can be submitted to the system in one batch, and that the job control program processes one job after the other without requiring your intervention. The job or jobs submitted for execution in one batch are referred to as a job stream.

You can interrupt the processing of a job stream to make last-minute changes to one of the jobs, or to squeeze in a rush job. You do this by means of the PAUSE command, which you enter from the console printer-keyboard or the display console, and which causes processing to halt at the end of the current job or job step, depending on the operand specified in the PAUSE command.

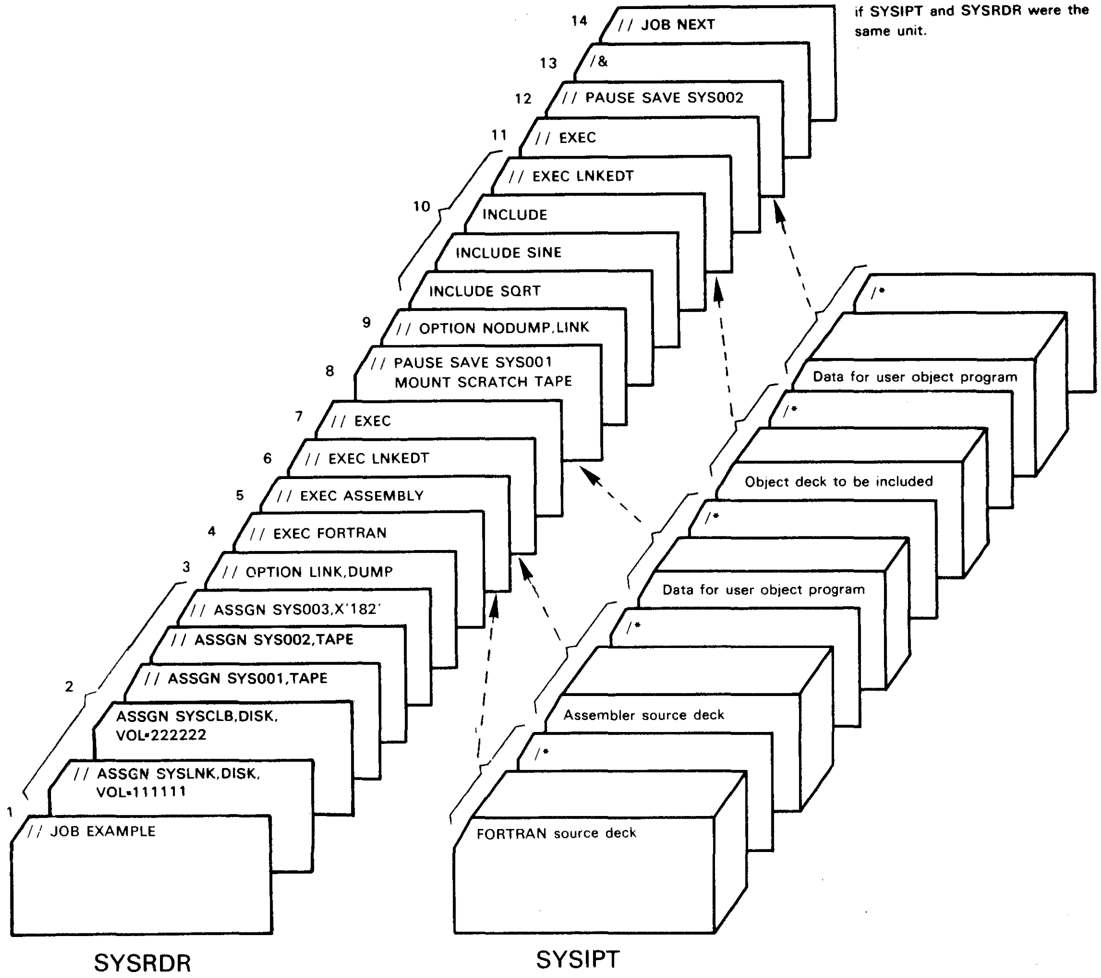
```
// JOB name
    .
    .
// EXEC PROGA
    .
    . (data)
    .
/*
    .
    .
// EXEC PROGB
    .
    . (data)
    .
/*
/&
```

If SYSIN is used (SYSRDR and SYSIPT refer to the same device), each EXEC statement may be followed by input data for the program that is to be executed. The end of the data should be marked by a statement containing /* in positions 1 and 2.

A detailed example of a job is shown in Figure I-6. The numbers on the lefthand side of the statements and commands refer to theexplanatory text on the following pages.

Although the programmer normally supplies the job control statements that are necessary to run a job, this example is a useful aid to familiarize yourself with job control statements and their functions. See also Section II *Communication with the System.*

Example of a Job



Broken lines indicate where the input in SYSIPT would be placed if SYSIPT and SYSRDR were the same unit.

Figure I-6. Example of a job

1. The JOB statement identifies the job and indicates the beginning of the job control statements for that job.

2. These are the ASSGN statements for the various job steps, assuming that these assignments differ from the current assignments. The assignments established by the ASSGN statements remain in effect until the / & statement, when they will be reset to the standard assignments (established during system generation), and/or to assignments specified by operator commands. The assignment for SYSCLB must always be permanent and is therefore given in the form of a command.

3. The OPTION statement specifies that the output of the FORTRAN compiler and the assembler is to be written on SYSLNK for subsequent link-editing, and that a dump is to be produced if the job ends abnormally.

4. This EXEC statement specifies that the program, in this case the FORTRAN compiler, is to be executed. It must be followed by the FORTRAN source deck and by the /* statement if SYSIPT is the same device as SYSRDR.

5. This EXEC statement specifies execution of the assembler program. It must be followed by the assembler source deck and the /* statement if SYSIPT is the same device as SYSRDR.

   Note: If the input stream is entered on an IBM 5425 MFCU, the /* card must be followed by a blank card.

6. This EXEC statement specifies execution of the linkage editor program, which then combines the FORTRAN and assembler object modules on SYSLNK and places the link-edited program in the core-image library.

7. EXEC statement for the link-edited object program in the core-image library. Data for this program must follow this statement if SYSIPT and SYSRDR are assigned to the same device.

8. The PAUSE statement causes the job control program to stop processing temporarily, so that you can act according to the instructions contained in the statement. You may also enter commands at this time. Processing continues when you press END (or ENTER).

9. This OPTION statement specifies that a dump (a printout of storage) is not required and that the object module is to be link-edited.

10. The INCLUDE statement signals that an object module is to be included. Note that this is neither a job control statement nor an operator command. It is one of four linkage editor control statements (PHASE, INCLUDE, ENTRY, and ACTION). The name of the object module is specified as an operand - in this case SQRT and SINE. These two modules are in the relocatable library. If no operand is specified, the module to be included should follow on SYSIPT. The EXEC statement causes the resulting program to be link-edited and placed in the core-image library.

11. This EXEC statement causes the program to be executed. The blank operand indicates that the program to be executed is the one just link-edited and temporarily placed in the core-image library. The data for the program must follow this statement if SYSIPT and SYSRDR are assigned to the same device.

12. This PAUSE statement requests operator action. You may also enter commands at this time.

13. The end-of-job statement causes all temporary I/O assignments to be reset to the standard assignments and/or to the assignments established by operator commands.

14. JOB statement for the next job.

Multi–programming

Normally, the processing time of a program is much shorter than the time that is required by that program to get its input and to produce its output, because the processing speed of the CPU is much greater than the speed at which the I/O devices can operate.

Multiprogramming compensates - to a large extent - for the difference between the high processing speed of the CPU and the relatively low operating speed of the I/O devices by allowing two or more programs to be executed concurrently: while one program is performing I/O operations, the CPU is free to be used by other programs.

I/O request

Assume that two problem programs are in storage for concurrent execution. The supervisor would then ensure efficient processing as follows: One program receives control of the CPU and can remain active until it has to issue an I/O request (that is, it has to read or write a record before it can go on processing). At that moment, the supervisor takes control, starts the necessary I/O operation, and gives control to the second program, which can now use the CPU while the first program waits for completion of its I/O operation. When this operation is complete, the first program is given control of the CPU again.

This method ensures that the CPU is idle only if both programs have to wait for I/O, that is, if the second program issues an I/O request before the I/O operation for the first program is complete. This idle time, however, is much shorter than it would be if the two programs were executed one after the other.

Partitions

One of the prerequisites of multiprogramming is that storage is divided into sections. In DOS/VS these are referred to as partitions. Each partition can contain only one program at a time. The number of partitions in a system, as well as their sizes, are defined at the time when the system is generated. You cannot change the number of partitions during system operation, but you can modify their sizes by means of an operator command.

In DOS/VS, you have one background partition (BG) and up to four foreground partitions (F1, F2, F3, F4). The background partition is always present.

Priorities

Because the programs in a multiprogramming environment take turns with each other in using the CPU, processing must proceed according to priority rules. This means that, if several programs are ready to go on processing, the supervisor must know which of them is to get control of the CPU first. The processing priority of a program depends on the priority of the partition in which it runs. Normally, F1 has the highest priority, followed by F2, F3, and F4; the background has the lowest priority. These priorities are so-called default values, which means that they are valid when nothing else has been specified.

In a single-partition system (BG only), only one program can be in storage at a time. If it needs input or output, it issues an I/O request to the supervisor, which initiates that request and after completion returns control to the program. During the interval in which the I/O operation is performed, the CPU remains idle.

The following table shows the default priorities of the partitions in each of the five possible combinations of partitions in a system.

Priority	5 Partitions	4 Partitions	3 Partitions	2 Partitions	1 Partition
1	F1	F1	F1	F1	BG
2	F2	F2	F2	BG	
3	F3	F3	BG		
4	F4	BG			
5	BG				

In DOS/VS, these default priorities may be changed during system generation or you, the operator, can change them at any time during system operation by means of the PRTY command.

Figure I-7 shows that multiprogramming makes a more efficient use of the CPU. The numbers at the top of the illustration are explained in the following text.

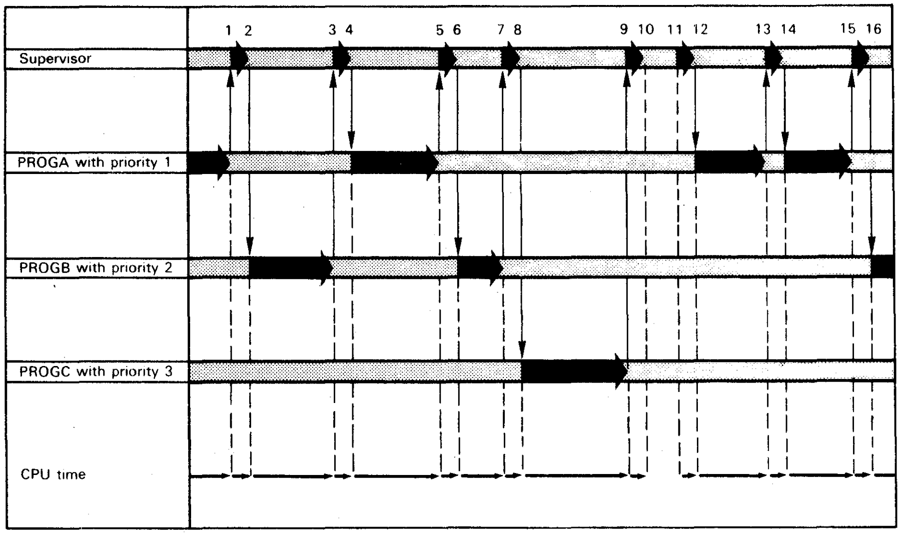


Figure I-7. **CPU-usage in a multiprogramming system**

1. PROGA issues an I/O request to the supervisor.
2. The supervisor, having started that request, gives control to the highest-priority program that is ready to run, in this case PROGB.
3. The supervisor is notified that the I/O operation for PROGA is complete; PROGB loses control.
4. Since PROGA has a higher priority and is ready to continue, it is given control.
5. PROGA issues another I/O request.
6. PROGB was interrupted in the middle of processing; it is ready to continue processing and is given control.
7. PROGB issues an I/O request.
8. Since PROGA is still waiting for completion of its I/O operation, PROGC is now the only program ready to run and it receives control.
9. PROGC issues an I/O request. The supervisor starts that request.
10. Because none of the programs is ready to run (neither I/O operation is complete), the system must go into the wait state, that is, the CPU is idle.
11. The supervisor is notified that the I/O operation for PROGA is complete.
12. The supervisor returns control to PROGA so that it can continue processing.
13. The supervisor is notified that the I/O operation for PROGB is complete.
14. PROGB could continue now, but PROGA has the higher priority, is ready to run, and therefore receives control.
15. PROGA issues another I/O request.
16. PROGB, which is waiting to continue, now receives control.

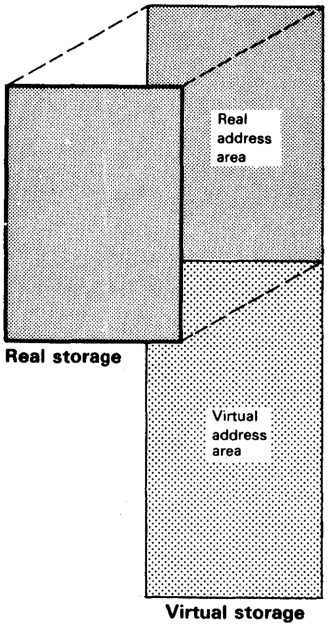
Virtual Storage

Virtual/Real storage

In DOS/VS, programmers do not have to write their programs in such a way that they fit entirely into the computer's storage. They write them for a much larger storage, which does not really exist and is therefore called virtual storage (as opposed to the computer's real storage).

Real address area

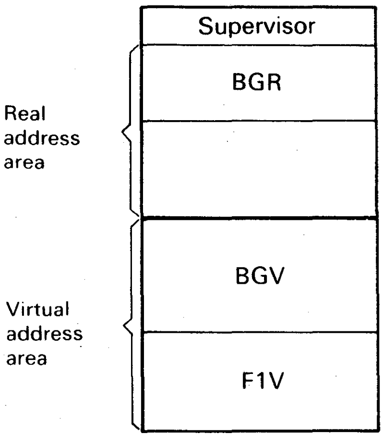
Virtual address area



Virtual storage consists of two separate areas. The first one is called the real address area, because it covers all the addresses that exist in the real storage of your computer. The second one is the virtual address area. Its addresses start just beyond the highest address of the real address area and range up to a value that was specified when your DOS/VS system was generated. Virtual storage is normally considerably larger than real storage.

As explained in the section *Multiprogramming*, storage is divided into a number of partitions. Virtual storage in DOS/VS introduces the concept of 'paired' partitions. Each such pair has one part in the virtual address area; this is called the virtual partition. The other part is in the real address area and is called the real partition. Virtual and real partitions to which no storage has been allocated are set to 0K.

A 3-partition DOS/VS system in which storage has been allocated to the virtual partitions BGV and F1V and to the real partition BGR appears as follows:



Virtual storage

The supervisor always resides in the lower part of the real address area.

No storage has been allocated to F2R and F1R; both are set to 0K.

No storage has been allocated to F2; it is set to 0K.

Virtual/Real partitions

Virtual mode
Real mode

A program to be executed under DOS/VS must run either in a virtual partition (virtual-mode execution) or in a real partition (real-mode execution); it cannot use a combination of both. If - in the above example - a program runs in BGV, no program can run in BGR, and vice versa.

The above description may lead you to believe that programs are executed in virtual storage. This is only what appears to happen. The execution of a program, however, requires real storage. This raises a problem because, unlike the real address area, the virtual address area does not have physical space in real storage that is directly related to it.

Page data set
SYSVIS

As an intermediate step, the system places all virtual programs on an intermediate storage device, an extent on a disk pack, which is called the page data set. This disk pack resides on the disk drive assigned to SYSVIS. The page data set is associated with the virtual address area in the same way as real storage is associated with the real address area (see Figure I-8). However, the programs on the page data set also require real storage.

DOS/VS solves this problem by making all the real storage that is not being used by the supervisor and real-mode programs available to the programs residing on the page data set. The collection of all unused sections of real storage may thus be thought of as a pool on which all programs on the page data set can draw.

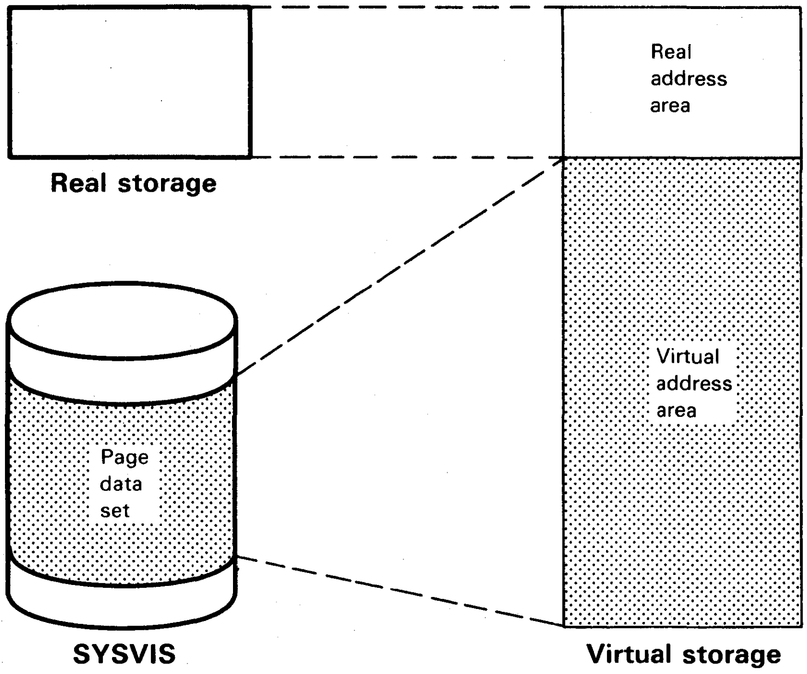


Figure I-8. The page data set is an image of virtual storage

It has already been said that virtual storage is normally much larger than real storage. Consequently, not **all** the virtual programs can be accommodated by real storage at the same time.

Pages

This is no problem, since only parts of a program can be processed in real storage at a time. Virtual programs are therefore brought into real storage in pieces, as and when they are required for execution. Each of these pieces is 2K bytes in size and is called a page. The entire virtual address area is divided into such pages. The page data set, which represents the virtual address area, has a specific 2K-slot for each of these pages.

Page frames

Page pool

Real storage is also divided in 2K-blocks. These are called page frames. All the page frames available for the execution of virtual programs, that is, all those not currently used by the supervisor and for the execution of real-mode programs, form the page pool.

Paging in

Paging out

At any point in time, only some of the pages that form a program are needed in real storage for execution. The system identifies these pages, finds them on the page data set, identifies available page frames in the page pool, and brings the pages into page frames in real storage. Bringing a page into a page frame is called paging in. As program execution proceeds, some pages will no longer be needed and others, which are not yet in a page frame, must be retrieved from the page data set. Sooner or later the page pool will be filled, and yet other pages must be brought in. The system then makes room for new pages by writing pages that are no longer needed back to their slots in the page data set (paging out). Reading in and writing out pages is called paging.

Figure I-9 shows how the pages of Program A are paged in and out between the page data set and real storage. Note that Program A occupies only one partition, and that other programs' pages (in other partitions) may be paged in and out at the same time.

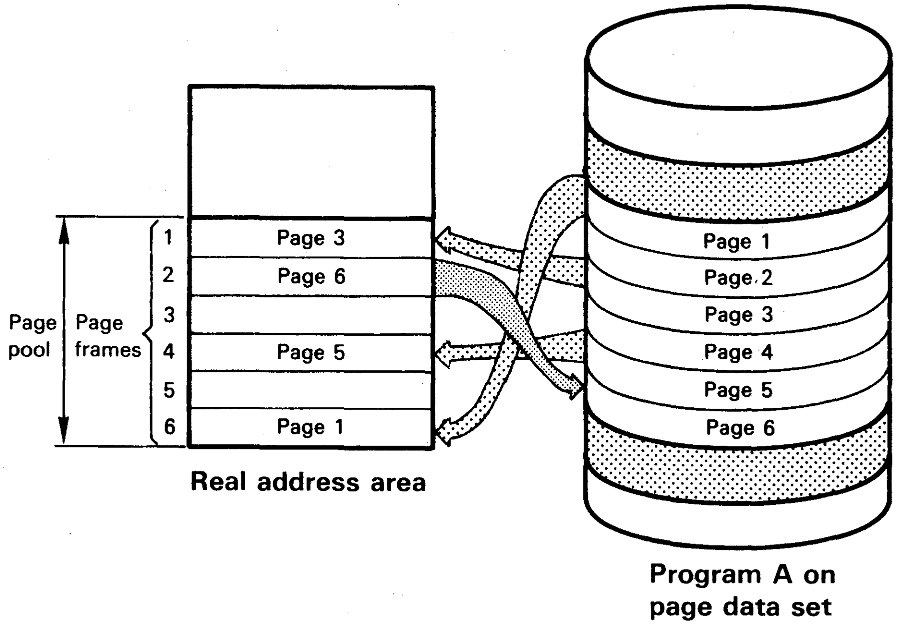


Figure I-9. Paging between the page data set and real storage

The above information on how the system executes virtual-mode programs is not essential to you in performing your job as an operator, but you may find it of interest if you want to understand how DOS/VS operates.

Here is a brief summary of the points that you should know and remember about DOS/VS operation:

1. The number of partitions specified when a system is generated refers to pairs of real and virtual partitions: A 3-partition system, for example, has three real **and** three virtual partitions.

2. When a program runs in real mode in a real partition, no program can run in the corresponding virtual partition, and vice versa.

3. Even when a real partition has been allocated, this partition's page frames belong to the page pool except for the time when a real-mode program is actually running in it. (For example, while a virtual-mode program is running in a virtual partition, the corresponding real partition is part of the page pool.)

4. The job control program always runs in virtual mode and requires 64K bytes. A virtual partition in which a program is to be executed must therefore have at least 64K bytes.

5. A real partition may be as small as 2K bytes. Note, however, that even a real-mode program must be started by job control, which requires 64K bytes for the corresponding virtual partition.

What you have to do for virtual storage is very little, and in each case you will receive detailed instructions from your system programmer:

1. At IPL time, you mount the pack containing the page data set and enter the DPD command.

2. Whenever necessary during system operation, you allocate or reallocate space to real and virtual partitions using the ALLOCR and ALLOC commands. (You can use the MAP command to obtain a survey of current allocations at any time.)

POWER/VS

POWER/VS is a service program designed to increase the productivity of computer installations operating under DOS/VS. POWER/VS runs in a virtual partition. It can service up to four other partitions (a maximum multiprogramming system) for virtual-mode as well as real-mode program execution. POWER/VS's own partition must have a higher priority than any of the partitions it supports.

POWER/VS job

The POWER/VS program reads all SYSIN card and diskette input for the partitions operating under control of POWER/VS, and stores it on disk. This card input on disk represents job stream input and is grouped into units of work upon which POWER/VS can act. Such a unit of work, a POWER/VS job, consists of all the cards contained between the // JOB and the /& cards if the DOS/VS job control language (JCL) is used; a POWER/VS job consists of all the cards contained between the * $$ JOB and the * $$ EOJ cards if the POWER/VS job entry control language (JECL) is used.

JECL

The POWER/VS program transfers the input from disk to the individual partitions as and when required. Conversely, POWER/VS intercepts all printer and punch output, and again stores it on disk or tape. The actual printing and punching is done later. This procedure is shown in Figure I-10.

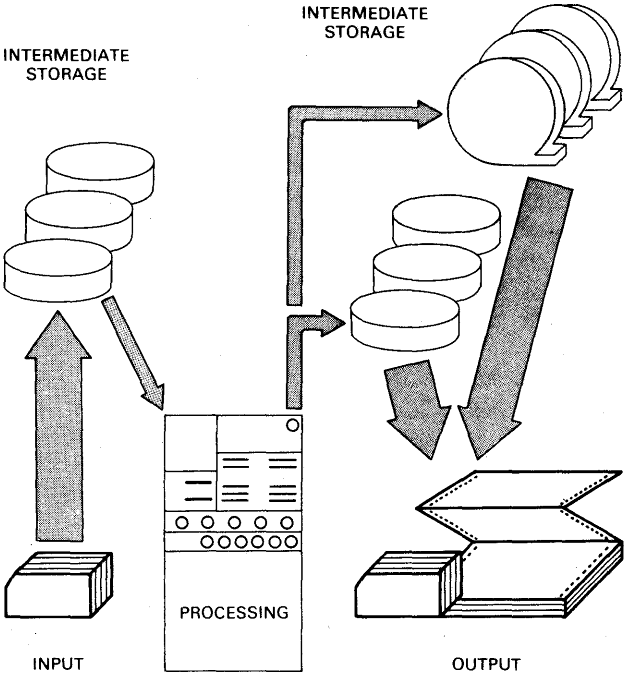


Figure I-10. POWER/VS stores input and output temporarily on disk. Output may also be stored on tape

POWER/VS permits I/O operations for each partition under its control to be handled individually. For example, one partition can have only its input spooled (a reader-only version of POWER/VS), while another can have only its output spooled (a writer-only version).

POWER/VS increases system productivity in two ways:

I/O devices

1. The speed of processing in POWER/VS-supported partitions is not dependent on the pace of the slower I/O devices (card readers, punches, printers, diskettes). Their input and output operations are carried out at disk or tape I/O speed. In combination with multiprogramming this guarantees the most efficient utilization of CPU time. In other words, the system can do more work in a given period of time.

2. Maximum use is made of the system's devices. In a multiprogramming system without POWER/VS, each partition requires its own card reader (or diskette device), printer, and punch. Much of the time these devices are idle, waiting for the program in the partition to read another card or print another line. With POWER/VS, one card reader or diskette device assigned to the POWER/VS partition can handle the input for all supported partitions, and one printer and one punch - also assigned to the POWER/VS partition - are sufficient to produce all output. This ensures that under POWER/VS, the devices assigned to the POWER/VS partition are operating at maximum speed most of the time.

   You, the operator, may start and stop input reading and output printing and punching independently of execution. Whenever a card reader, a punch, a printer, or a diskette device becomes inoperative, the system can continue processing those job entries that are already stored on disk; similarly, the system can store the output of these job entries on disk. When the I/O device becomes operative again, reading, punching, or printing can continue.

POWER/VS is documented in the following manuals:

DOS/VS POWER/VS Installation Guide and Reference, GC33-6408
DOS/VS POWER/VS Workstation User's Guide, GC33-6049.

Communication with the System

Section 2

You, the operator, must see to it that all operations at your data-processing installation can proceed smoothly and that delays are kept as short as possible. You can do this efficiently only if you are kept informed about the status of the system and, in return, tell the system what you want it to do. In other words, the system must communicate with you, and you must communicate with the system. The console printer-keyboard or the display console enables you to communicate with the system and permits the system to communicate with you.

For details on the syntax of statements and commands, refer to the section *Reference Information.*

The system communicates with you by issuing messages on SYSLOG, the logical unit to which the console printer-keyboard or the display console is assigned. Each message is preceded by a partition identifier and a message code, which includes an action indicator. This action indicator indicates what operator response is required.

Grouped according to these action indicators, there are four types of message:

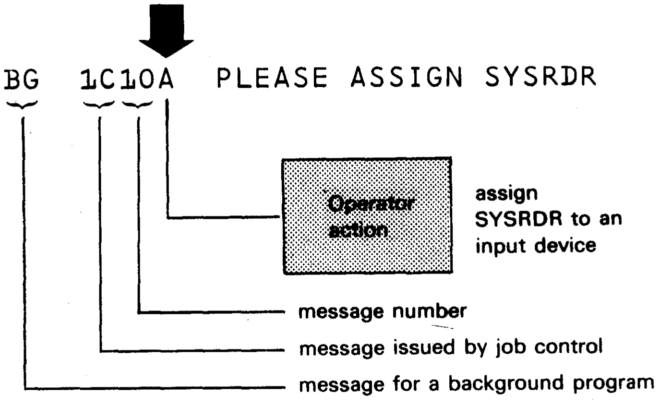
- Action (A) messages
- Decision (D) messages
- Information (I) messages
- Eventual-action (E) messages

Note: Do not rely on your memory but make a habit of looking up each message that is issued by the system. This will save you a lot of time and trouble. Full details of all DOS/VS messages, including operator action and responses, are contained in *DOS/VS Messages*, GC33-5379.

Action Messages

A

The message number is followed by the letter A, indicating that your action is required.



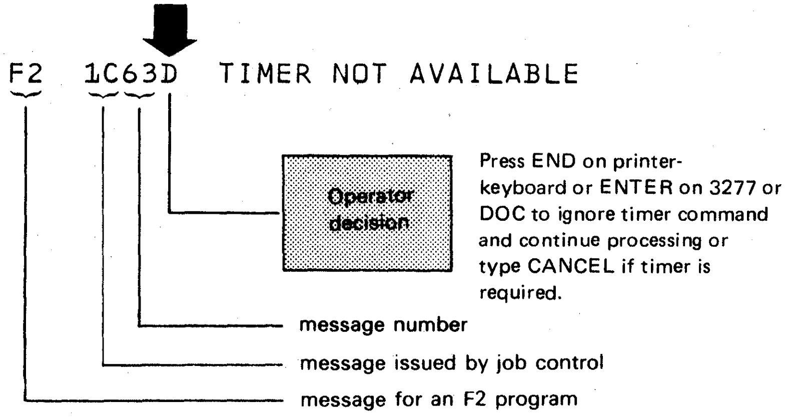
```
BG  1C10A  PLEASE ASSIGN SYSRDR
```

Operator action
assign SYSRDR to an input device
message number
message issued by job control
message for a background program

D

Decision Messages

The message number is followed by the letter D, indicating that you must make a choice between alternative courses of action.



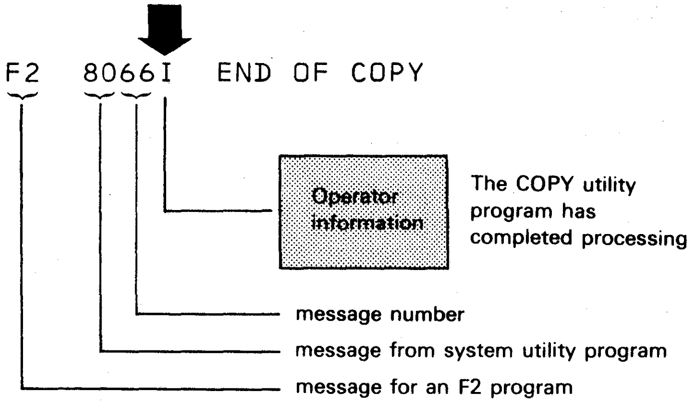
```
F2  1C63D  TIMER NOT AVAILABLE
```

Press END on printer-keyboard or ENTER on 3277 or DOC to ignore timer command and continue processing or type CANCEL if timer is required.

I

Information Messages

The message number is followed by the letter I, indicating that no specific operator action is required.



```
F2  8066I  END OF COPY
```

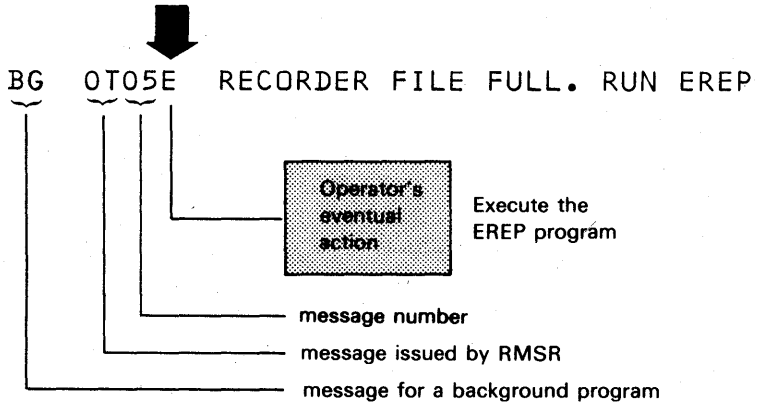
Operator information
message number
message from system utility program
message for an F2 program

The COPY utility program has completed processing

E

Eventual-Action Messages

The message is followed by the letter E, indicating that some action will have to be taken eventually.



```
BG  0T05E  RECORDER FILE FULL. RUN EREP
```

Execute the EREP program

Whenever operator action or a decision is necessary, the program that issued the message waits until you have entered an appropriate response to the message.

Hard Wait Codes

A number of errors may occur that cause the system to enter an uninterruptable wait state. This is indicated by the wait light on the CPU console. At the same time, a code in bytes 0-3 of real storage identifies the type of error. Retrieval of codes from bytes 0-3 is described in detail in *DOS/VS SADP*, GC33-5380.

Operator-to-Syst Communication

You can communicate with the system by means of either job control statements or operator commands. Job control statements have two slashes (//) in positions 1 and 2, whereas operator commands do not have slashes (their operation code may, but need not, start in position one).

Job control statements

Job control statements may either be punched in cards and entered via a card reader (SYSRDR); or be in card-image format on tape, disk, or diskette. They are accepted by the system only between jobs or job steps.

Operator commands

Operator commands are usually entered either from the console printer-keyboard or from a display console, which must be assigned to SYSLOG. If entered from the console printer keyboard, commands must be followed by pressing END; if entered from a display console, they must be followed by pressing ENTER.

There are five types of operator commands:

IPL commands

1. IPL commands, which are accepted only at IPL time (when you perform the IPL procedure). These commands prepare the DOS/VS system for operation.

Attention commands

2. Attention commands, which are accepted after you have pressed REQUEST on the console printer-keyboard. (If 'AR' does not appear, press REQUEST again, after which the message 1I40D REQUEST CANCEL [-ENTER POWER/VS COMMAND] is displayed.) Attention commands from a display console may be entered without pressing REQUEST (PA1 on the 3277), unless the message ENTER RESPONSE is displayed. In this case, press REQUEST/PA1 first.

Job control commands

3. Job control commands, which are accepted only between jobs or job steps.

POWER/VS commands

4. POWER/VS commands, which are treated like attention commands.

VTAM network operator commands

5. VTAM network operator commands, which are needed to monitor and control a VTAM telecommunication network.

All DOS/VS job control statements and operator commands, except those commands used for POWER/VS and VTAM, are described in detail in the section *Reference Information.* POWER/VS commands are described in *DOS/VS POWER/VS Installation Guide and Reference,* GC33-6048, and *DOS/VS POWER/VS Workstation User's Guide,* GC33-6049. VTAM operator commands are described in *VTAM Network Operating Procedures,* GC27-6997.

By using the appropriate DOS/VS operator command you can, for example, perform the following functions:

- **Set system values.** The ADD command adds devices to the PUB table; the DEL command deletes devices from the PUB table. The CAT command identifies the disk drive on which the VSAM catalog is mounted. The SET command sets the values for date and time. The DPD command defines the page data set.

- **Temporarily suspend processing.** The PAUSE command causes a partition to pause between jobs or job steps, enabling you to take action such as mounting the next tape reel or disk pack.
- **Cancel jobs.** The CANCEL command terminates the execution of a job.
- **Change I/O device assignments.** The ASSGN command assigns a physical I/O device to a logical unit name. The DVCDN (device down) command informs the system that an I/O device is inoperative. The DVCUP (device up) command informs the system that an I/O device that was inoperative is now operational again. The RESET command resets temporary I/O assignments - which are valid for only one job - to the permanent assignments or, where no permanent assignments were made, to the standard assignments established when the system was generated.
- **Control magnetic tape operations.** The MTC command controls magnetic tape operations, such as rewinding tapes, skipping files, writing tape marks.
- **Closing files.** The CLOSE command closes a system or programmer logical unit assigned to magnetic tape, or a system logical unit assigned to a disk drive or a diskette input/output unit.
- **Obtain information from the system.** The LISTIO command prints or displays a list of current I/O device assignments in the system. The LOG command prints or displays all job control statements and commands occurring in the partition in which the command is issued. The NOLOG command suppresses logging of the job control statements and commands.
- **Control multiprogramming.** The following commands are valid only in a multiprogramming environment:

  The ALLOC command enables you to allocate storage to virtual partitions; the ALLOCR command enables you to allocate storage to real partitions. The BATCH, START, UNBATCH, and STOP commands are used to initiate, interrupt, and reinitiate processing in a partition. The MAP command prints a map of the current sizes of the partitions, both real and virtual, as well as the size of the page pool. The PRTY (priority) command enables you to modify the priorities of the partition. The START command enables you to resume processing after a STOP command.

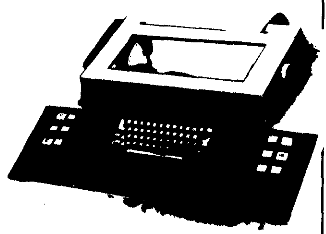
START and BATCH are identical

Although the normal communication device for operator commands is SYSLOG (assigned to a console printer-keyboard or display console), commands may also be entered from SYSRDR, with the exception of attention commands.

Devices Used for Operator/System Communication

This chapter deals with four types of devices that are available for operator-to-system and system-to-operator communication. They are:

1. The IBM 3210 and 3215 Console Printer-Keyboards, which are used in conjunction with IBM System/370 Models 135, 145, and 155-II.
2. The Video-Display Keyboard Console, consisting of a video display unit (screen) and a keyboard, used in conjunction with IBM System/370 Models 115 and 125. This console is referred to in this manual as the DOC (Display Operator Console).
3. The 3277 Display Console.
4. The Console Display-Keyboard used in conjunction with IBM System/370 Models 138 and 148.



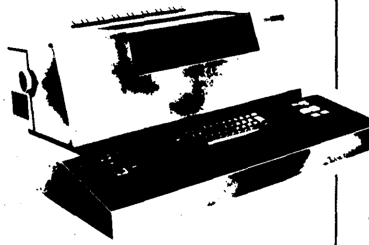
IBM 3210 Console Printer-Keyboard

IBM 3210 and 3215 Console Printer-Keyboards

The console printer-keyboard is an I/O device that allows you to enter operator commands, job control statements, and responses to messages that are printed on its printer.

The console printer-keyboard consists of a printer, a keyboard, indicators, and control keys.

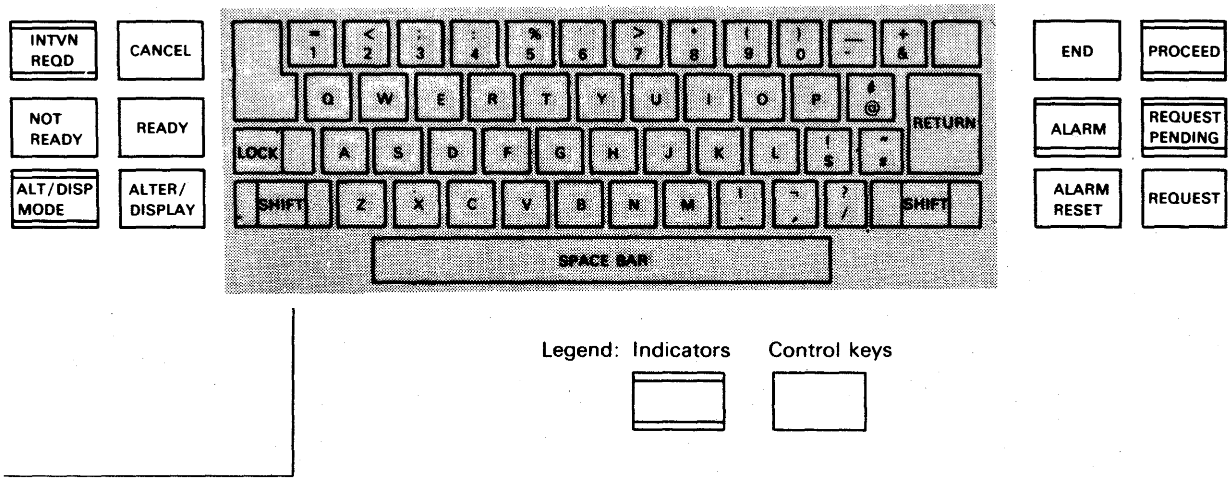
Regardless of the type of printer-keyboard selected for your installation, you have available to you the same set of keys, indicators, and control keys. Both printer-keyboards have the same arrangement as the IBM 1052 Printer-Keyboard Model 7, used in System/360, except that the 3210 and 3215 keyboards have a 'cancel' key and do not have an 'alternate coding' key.



IBM 3215 Console Printer-Keyboard

Indicators and Control Keys

The indicators and control keys that are provided on both sides of the keyboard show status or provide certain control functions. The layout of the keyboard below shows how these indicators and control keys are arranged and Figure II-1 summarizes their functions.



Devices Used for Operator/System Communication (continued)

Indicator	Indication When On
ALARM	Alarm issued - operator action required
ALT/DISP MODE	Request for alter/display mode accepted
INTVN REQD	Printer has no paper, or printer-keyboard is not ready for use
PROCEED	Keyboard unlocked and ready to accept keyboard entry (turned on by pressing REQUEST [program-dependent], by pressing ALTER/DISPLAY, or by keyboard entry)
REQUEST PENDING	Request operation has been initiated but has not yet been serviced

Control Key	Function When Pressed
ALARM RESET	Turns off alarm and ALARM indicator
ALTER/ DISPLAY	Requests or ends alter/display operation (when used to end an alter/display operation, alter/display mode is retained).
CANCEL	Deletes the current keyboard entry. It can be used if you notice a mistake in the current entry. The input can then be re-entered. This key is not active during an alter/display operation.
END	Terminates keyboard entry, printout, or alter/display (when used to terminate alter/display, exit is made from alter/display mode)
NOT READY	Places the printer-keyboard in a not-ready state. It should, for example, be used before opening the printer-keyboard cover.
READY	Places the printer-keyboard in a ready state, if forms are in the printer and the cover is closed.
REQUEST	Requests the CPU to accept keyboard entry

Figure II-1. Functions of indicators and control keys of the IBM 3210 and 3215 Console Printer-Keyboards

Display Operator Console (DOC)

The DOC is an integral part of the IBM System/370 Models 115 and 125 and serves two purposes:

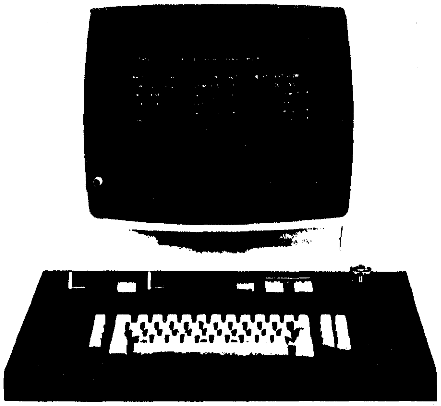
1. It replaces the conventional CPU console panel with its numerous controls and indicators.

2. It is used to enter operator commands, job control statements, and responses to messages that are displayed on its screen.

The DOC offers the following advantages over the console printer-keyboard:

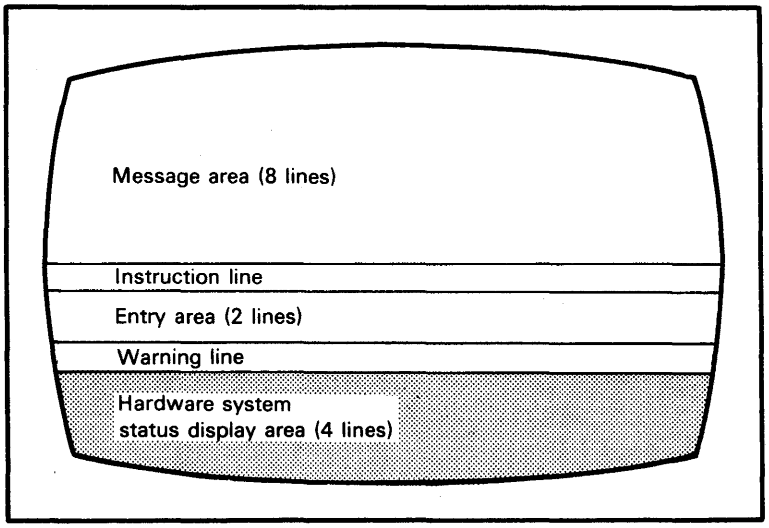
- input can be entered immediately; there is no need to press REQUEST.
- messages are delivered faster, and
- errors in statements and commands can be corrected more easily.

The DOC consists of two parts: the display unit and the operator keyboard with control panel.



Display Unit

The display unit can display a maximum of 16 lines, 12 of which are available to you, the operator. The last four lines serve to display the status of the hardware; these lines are used for service purposes (see Figure II-2).



Note: the solid lines and words do not appear on the screen.

Figure II-2. Format of the DOC screen

The **message area** is reserved for messages from the system and from user-written programs.

The **instruction line** serves to display messages that inform you of incorrect usage of the control command (K command - discussed in a subsequent chapter) or to draw your attention to operating conditions you should be aware of.

The **entry area** serves to enter commands.

The **warning line** displays messages related to problems you must resolve.

The **hardware system status display area** is reserved for use by the IBM customer engineer.

Each line can contain a maximum of 56 characters. Messages longer than 54 characters are continued on the next line. The format of the lines in the message area is shown in Figure II-3.

Blank	Line number	Blank or *	Partition indicator	Blank	Message text	
1	2	3	4	6	7	56

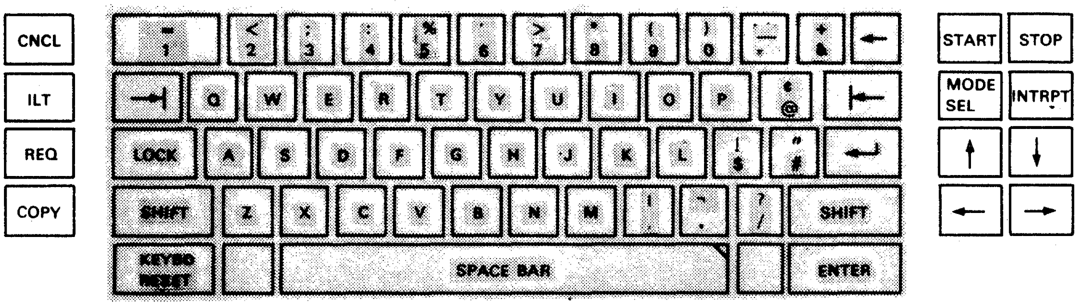
Figure II-3. Format of the DOC message lines

Devices Used for Operator/System Communication (continued)

Audible alarm

The DOC has an audible alarm, which sounds when you have to respond to a message, when you have made an error while entering the control command (K command), or when the message 'MESSAGE WAITING' is displayed. The audible alarm can be switched off.

Operator Keyboard

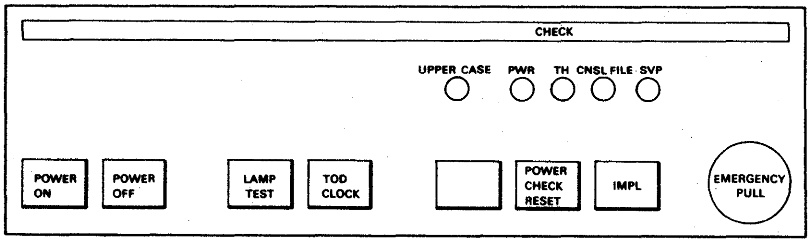


The functions of the keys are described in detail in *Operator's Library System/370:*

Model 115 Procedures, GA33-1514.
Model 125 Procedures, GA33-1509.

Control Panel

The control panel, which is located at the top of the keyboard, contains a number of switches, keys, and lights. The keys are used for basic tasks such as making the system operational. The lights alert you to check conditions in the system.



Full details of the switches, keys, and lights are given in *Operator's Library System/370:*

Model 115 Procedures, GA33-1514.
Model 125 Procedures, GA33-1509.

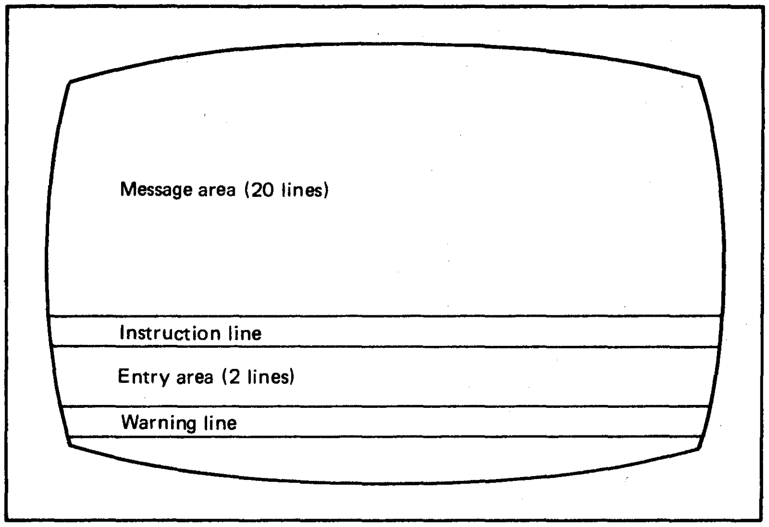
3277 Display Console

The 3277 CRT console device is supported as operator console on all System/370 models to which it is attachable. The supervisor must have been generated with the proper support, otherwise when the 3277 is assigned to SYSLOG, the message

1A2nD INVALID DEVICE TYPE

is displayed.

Like the DOC, the 3277 display console is used to enter operator commands, job control statements, and responses to messages that are displayed on its screen. The upper 20 lines of the screen are used as message area, and lines 21-24 are used like lines 9-12 of the DOC screen, that is, as instruction line, entry area, and warning line (see Figure II-4). However, unlike the DOC, the 3277 has no hardware system status display area.



Note: The solid lines and words do not appear on the screen.

Figure II-4. Layout of the 3277 screen

The **message area** is reserved for messages from the system and from user-written programs.

The **instruction line** serves to display messages that inform you of incorrect usage of the control command (K-command - discussed in a subsequent chapter) or to draw your attention to operating conditions you should be aware of.

The **entry area** serves to enter commands.

The **warning line** displays messages related to problems you must resolve.

Each line can contain a maximum of 80 characters. Messages longer than 77 characters are continued on the next line. The format of the lines in the message area is shown in Figure II-5.

Blank	Line number	Blank or *	Partition indicator	Blank	Message text	
1	2	3	4	6	7	80

Figure II-5. **Format of the 3277 message lines**

Model 138/148 Console Display-Keyboard

The Models 138 and 148 are operated and controlled through a system control panel on the CPU and a standard display console consisting of a display unit (cathode ray tube) and an operator keyboard. Operator input is entered through the alphameric keyboard (and, at the same time, displayed on the display unit), and messages are displayed on the screen. The physical capacity of the screen is 24 lines with 80 characters each.

Console support is provided in three modes:

- Printer-keyboard mode
- 115/125 display mode
- 3277 display mode (equivalent to 3277 support)

The required mode can be selected before IPL by changing the microprogram configuration with the aid of the 'IOC Load' facility. For further details of this, see the appropriate hardware description.

Printer-Keyboard Mode

This console mode is functionally equivalent to the 3215 support, that is, the display console and the 3286-2 printer are treated like a 3210/3215 console printer-keyboard. In this mode, all 24 lines available on the screen are used: 23 as output lines, and the last line as input line. The output area is filled sequentially with messages and operator input (transferred from the input line).

Each message requires two lines: the partition identifier occupies the first line and the message text the second line. When the output area (23 lines) is full, the top six lines are automatically deleted (by the hardware) and the remaining lines are moved to the top of the screen, thus leaving space at the bottom for fresh information. The lines that are deleted cannot be recovered, since a disk hard-copy file is not supported in printer-keyboard mode. However, all lines that appear on the screen are also printed on the 3286-2 printer to provide hard-copy output.

115/125 Display Mode

This console mode is compatible with the Model 115/125 Display Operator Console (DOC) support, except for the display of hardware status for service purposes. Since the hardware information will be available on the system control panel, only the upper 12 lines of the DOC display will appear on the screen:

8 message lines
1 instruction line
2 entry lines
1 warning line

The line length of 56 characters will be approximately centered across the screen width, with every second physical display line utilized to spread the information vertically on the screen.

All system messages and operator entries are automatically stored in the hardcopy disk file and can later be retrieved with the Printlog utility program (see Procedure 24). With the optional 3286-2 printer installed, all lines that appear on the screen are also printed on the console printer, thus providing you with an immediate hardcopy of all messages and operator commands.

Screen Operation

The screen operations described in this chapter apply to both the Display Operator Console (DOC) and the 3277 Display Console, except where explicitly stated otherwise. Screen operation consists of deleting and redisplaying messages on the screen. Since there are only eight lines available for messages on the DOC and 20 for messages on the 3277, you will have to delete messages from time to time.

Hard-copy file

All lines that you delete from the screen are automatically stored on a direct-access storage device, which must be assigned to SYSREC. This message file is called the hard-copy file.

When the hard-copy file is almost full, the following message appears on the screen:

```
HARD COPY FILE SHOULD BE PRINTED
```

If you wish to save the messages, you must print them on SYSLST directly after this message. How this is done is described in the procedure 'Printing the Hard-Copy File'.

If you do not need a copy of the old messages, continue processing; the messages on the hard-copy file will be overwritten by new messages when the message

```
HARD COPY FILE IN OVERLAY MODE
```

is displayed.

It is also possible to redisplay messages from the hard-copy file on the screen. This chapter describes how to delete and redisplay messages.

IBM 5213 Console-Printer

If your installation uses the IBM 5213 Console-Printer, which is optional, all lines that appear on the screen are also printed on the console printer, thus providing you with a 'hard' copy of messages, statements, and commands.

Note that the hard-copy file is mandatory if a console-printer is not available; it is optional if a console-printer is attached to the Model 115 or 125.

Deleting Messages

Whenever the message area on the screen is full, you must delete some or all of the messages to make room for new messages. The K control command is used to delete messages. It has the following general format:

For syntax notation refer to Reference Information

$$\text{K} \qquad \begin{bmatrix} \text{S} & \\ \text{E} & [\text{,options}] \\ \text{D} & \end{bmatrix}$$

The operands S, E, and D have the following functions:

S - establishes and displays message deletion modes

E - deletes messages and line numbers of messages

D - controls the display of message line numbers

Note that, in the following description of the operands of the K command, default values are underlined.

A table with examples of the K command is contained in the section *Reference Information.*

Message Deletion Modes

The S operand of the K command is used to establish and display message deletion modes. It has the following format:

$$\text{S}\left[\begin{array}{l}\left[\text{,DEL} = \begin{array}{c}\underline{\text{Y}}\\ \text{N}\end{array}\right]\left[\text{,CON} = \begin{array}{c}\underline{\text{N}}\\ \text{Y}\end{array}\right]\left[\text{,ALM} = \begin{array}{c}\underline{\text{N}}\\ \text{Y}\end{array}\right]\left[\text{,SEG} = \begin{array}{c}\underline{\text{6}}\\ \text{n}\end{array}\right]\\ \underline{\text{,REF}}\end{array}\right]$$

All operands are optional; if they are omitted, the default values apply.

Automatic Deletion Mode

K S,DEL=<u>Y</u>

In automatic deletion mode, the deletable messages that start in the first four lines on the DOC (or 15 lines on the 3277) are automatically removed from the screen when the screen is full and a message is waiting to be displayed. Deletable messages are messages that require no action or for which action has already been taken. Note that A- and D-type messages from POWER/VS must be deleted manually.

Manual Deletion Mode

K S,DEL=N

In manual deletion mode you must delete messages yourself. If the screen is full - in either manual or automatic deletion mode - and a message is waiting to be displayed, the message

MESSAGE WAITING

appears in the warning line and the alarm is sounded (provided that you specified ALM=Y). You must then make room for the waiting message by deleting messages from the screen. This is done by the K command with the E operand (see *Deleting Messages Manually*).

Non-Conversational Deletion Mode

K S,CON=<u>N</u>

In non-conversational mode, all messages selected for manual deletion are deleted immediately, without verification.

K S,CON=Y

Conversational Deletion Mode

Conversational deletion mode allows you to verify all messages selected for manual deletion before they are removed from the screen (for details see *Deleting Messages Manually*).

K S,ALM=Y

K S,ALM=N

Audible Alarm

The audible alarm is switched on when you have specified ALM=Y. Note that the default value is ALM=Y.

K S,SEG=n

K S,SEG=6

Setting the Deletion Range

The SEG option allows you to specify the number of message lines to be deleted at a time by the K command. The range is 1-8 for the DOC and 1-20 for the 3277. (See also *Deleting Messages Manually*).

K S,REF

Displaying and Changing the Deletion Modes

You can use the REF option to display the deletion modes currently in effect. For example, if you wish to modify the default deletion modes, enter the command

K S,REF and press ENTER.

This displays the default deletion modes in the entry area in the format of the K command:

K S,DEL=Y,CON=N,ALM=Y,SEG=6

You can now change the default deletion modes either by the cursor or by entering a K command with the S operand. For example, to change from automatic to manual deletion mode and to specify a deletion range of four lines

- move the cursor to the characters to be changed in the entry area, Y (DEL=Y) and 6 (SEG=6) and enter the new characters, N and 4, or
- enter the command K S,DEL=N,SEG=4

Note that the other default parameters remain unchanged: nonconversational mode (CON=N) is still in effect, and the audible alarm (ALM=Y) remains switched on.

Deleting Messages Manually

If manual deletion mode (DEL=N) is in effect, you must either use the K command with the E operand, or the cursor, to delete messages from the screen. The E operand has the following format:

```
    ┌,SEG ┐
    │,n   │
E   │,n,n │
    │,N   │
    └     ┘
```

Deleting a Predefined Number of Messages

K E,SEG or K

The SEG option of the E operand indicates deletion of a number of lines as specified in the SEG option of the S operand. Since E and SEG are default values, you need to specify only K to delete a predefined number of lines.

Deleting a Single Line

K E,n

To delete a single line, specify its line number n, where n may be

- a digit from 1 to 8 (for the DOC)
- one digit (or two digits) from 1 to 20 (for the 3277)

Deleting Several Lines

K E,n,n

To delete more than one line, specify the number of the first and last line to be deleted. For example, to delete the first four lines from the screen, enter

```
K E,1,4
```

The line numbers must be entered in ascending order. A-, D- and E-type messages within the specified range are not deleted. Messages of this type are deleted only when directly pointed to by a command or by the cursor.

Deleting Line Numbers of Messages

K E,N

The N option of the E operand specifies that the message line numbers in character position 2 (for DOC) or 2 and 3 (for 3277) are to be deleted.

Deleting in Conversational Mode

The procedure to delete messages in conversational mode is as follows:

1. Enter the deletion request with a K command. This causes:

   - the message DELETION REQUESTED to be displayed in the instruction line

   - line numbers to be displayed if they had previously been deleted

   - the deletion request to be displayed in the entry area in the form K E,n,n

2. If the indicated lines are the ones you wanted to delete, press ENTER; the lines are then deleted.

3. If you made a mistake when entering the K command, move the cursor to the character(s) in error in the entry area and enter the correct value(s). Press ENTER to delete the messages.

4. If you decide that you must do something else before you can delete the messages, press CANCEL to cancel the deletion request.

Example

Assume that the default deletion range (SEG=6) applies and you enter K to delete the first six lines from the screen. The message

DELETION REQUESTED

is displayed in the instruction line, and the deletion request is displayed in the entry area in the form

K E,1,6

If you wish to delete only the first four lines, move the cursor to the character 6, type in the character 4, and press ENTER. If you are satisfied that you do want to delete the first four lines, press ENTER again and the lines will be deleted from the screen.

Deleting with the Cursor

The cursor can be moved to any position on the screen - except in the system status area - by means of the cursor positioning keys. To delete messages from the screen, move the cursor to any position in a message line, and press ENTER. The result is that this message is deleted, together with all deletable messages above it. The messages remaining on the screen are repositioned sequentially from the top of the message area.

If the cursor is not properly positioned when you press ENTER, the entry area is blanked and the cursor is repositioned to the left of the entry area. The message

ILLEGAL CURSOR OPERATION

appears in the instruction line. If you want to continue deletion by means of the cursor, reposition it to a valid message line and press ENTER again.

When you delete messages with the cursor in conversational mode, the procedure is the same as described in the preceding section, that is, the message DELETION REQUESTED is displayed and the deletion request appears in the entry area in the form K E,n,n.

Displaying Message Numbers

You can control the display of message numbers in character position 2 (for DOC) or 2 and 3 (for 3277) of the message lines by means of the D operand of the K command. The D operand has the following format:

```
D   [,N [,HOLD]]
```

Temporary Display

If you specify the D operand with the N option, message numbers are displayed until the next K command with an E operand is given.

K D,N

canceled by K E,...

Permanent Display

If you specify the D operand with the N **and** HOLD options, message numbers are displayed all the time; they can only be deleted by the command K E,N.

K D,N,HOLD

Redisplaying Messages

You can redisplay messages that were previously deleted from the screen by means of the D command. There are three versions of the D command, all of which are necessary to redisplay messages.

A table with examples of the D command is contained in the section *Reference Information.*

Version 1 enables you to enter the redisplay mode.

Version 2 enables you to control redisplay operations.

Version 3 enables you to terminate redisplay mode.

Entering Redisplay Mode

To enter redisplay mode, you use the D command, which has the following format:

```
    ┌   ┌ ,ALL ┐ ┐
D   │ L │ ,AR  │ │
    │   │ ,BG  │ │
    └   └ ,Fn  ┘ ┘
```

ALL - specifies redisplay of all messages in the hard copy file.

AR - specifies redisplay of attention-routine messages.

BG - specifies redisplay of messages from the background partition.

Fn - specifies redisplay of messages from a foreground partition. n identifies from which foreground partition, for example F3.

All operands are optional. You can specify D L or just D; the result will be D L,ALL.

When you have typed in the command and pressed ENTER, the command is displayed in the entry area, and the following text appears on the screen:

```
                                          ┌ Fn=nnnn ┐
*** MESSAGE REDISPLAY BWD     ALL=nnnn    │ BG=nnnn │ ***
                                          └ AR=nnnn ┘
```

BWD - indicates that the direction of redisplay is backward; when you change the direction to 'forward' (see following section), BWD is replaced by FWD.

nnnn - is the line count for all messages and for the messages pertaining to a partition.

Controlling Redisplay Operations

If the message area is not full when you enter the redisplay mode, the messages that are on the screen are moved to the bottom of the message area and the free lines at the top are filled with messages most recently stored on the hard copy file.

Whenever you then press ENTER, another eight (DOC) or 20 (3277) message lines are displayed, going backward in the hard copy file. If you wish to change the direction of redisplay, skip a number of messages, or redisplay messages for a partition other than the one originally specified, you must use version 2 of the D command; it has the following format:

```
    ┌     ┌,ALL  [,R]   ┐┐
    │     │,AR   [,R]   ││
    │     │,BG   [,R]   ││
D   │  L  │,Fn   [,R]   ││
    │     │,B    [,nnn] ││
    │     │,F    [,nnn] ││
    │     │,R           ││
    └     └,nnn         ┘┘
```

R - causes the screen to be reset to its initial status in which you started redisplaying. If preceded by either ALL, AR, BG, or Fn, the screen is reset to its initial status in accordance with the prefix.

B - changes direction of redisplay from forward to backward.

F - changes direction of redisplay from backward to forward.

nnn - specifies the number of messages to be spaced forward or backward; if nnn is not specified, 8 is assumed for the DOC, and 20 for the 3277.

The other operands are described on the preceding page.

All operands are optional. If you just specify D, the default is D L,8 (for the DOC) or D L,20 (for the 3277) plus the redisplay direction and the partition currently in effect. Specifying D without any operands has the same effect as pressing ENTER.

Examples

Assume that you entered redisplay mode with the command D L,BG:

1. To display eight (DOC) or 20 (3277) background messages at a time, press ENTER; direction is backward.
2. To change to forward specify: D L,F
3. To space forward 20 messages, specify: D L,20.
4. To display messages for F1, for example, specify: D L,F1
5. To reset the screen to its initial status when you started redisplay, specify: D L,R. You are still in redisplay mode.

Terminating Redisplay Mode

To terminate the redisplay mode, enter the D command with the E operand; the E operand has no options:

```
D E
```

The screen returns to the status which it had prior to entering the redisplay mode.

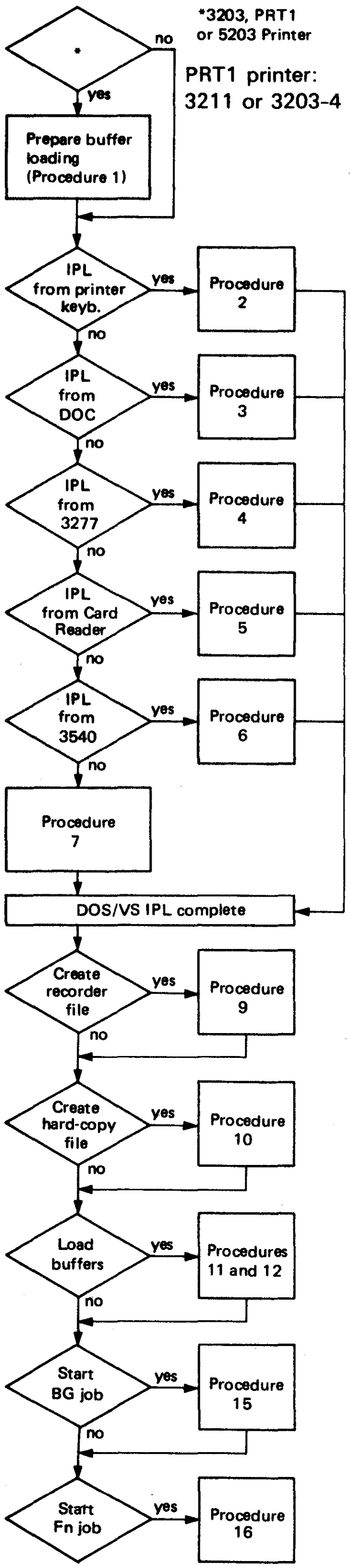
Section 3

This section tells you how to execute jobs under control of DOS/VS. The procedures are illustrated by examples. Each procedure is numbered for quick reference.

The individual procedures are listed on the next page.

Procedures
(continued)

Starting the System



Before you can run any jobs, you must start the system by performing the IPL procedure. This is also necessary after a system breakdown, or when a different DOS/VS system is to be used (when the SYSRES pack is replaced).

If your installation uses an IBM 3203, PRT1, or 5203 Printer, buffer loading for this device must be prepared prior to the IPL procedure. Loading the buffers of these printers is described in Procedure 1.

Procedures 2 - 7 describe the different methods of performing the IPL procedure:

- From the printer-keyboard. This method applies to all System/370 CPU models, except Models 115 and 125.
- From the DOC. This method applies to System/370 Models 115 and 125, whether they are cardless or not.
- From the 3277 Display Console. This method applies to all System/370 CPU models to which a 3277 is locally attached.
- From a card reader. This method applies to all System/370 CPU models with physical card readers.
- From the IBM 3540 Diskette. This method applies to all System/370 CPU models with a 3540 device.
- From the Model 158 display console. This method applies to the System/370 CPU Model 158.

If the supervisor at your installation was generated to support the RDE (reliability data extractor), you must furnish the system with additional information immediately after IPL. Procedure 8 tells you how to enter this information.

After successful IPL, the system issues the message DOS/VS IPL COMPLETE. If your system uses the RMSR (Recovery Management Support Recorder), you may have to create the system recorder file. If you operate a display console without a console printer, you have to create the hard-copy file. Procedures 9 and 10 describe how these files are created. Buffer loading of the IBM 1403, 3203, PRT1, and 5203 printers is described in Procedures 11 and 12. Once you have completed these preparations, you are ready to run jobs in the background or foreground partitions (Procedures 15 and 16) or to initiate POWER/VS.

Preparing the Printer for Buffer Load

1

Applies to all buffer-controlled IBM printers except the 1403 UCS printer.

If your computing system includes an IBM 3203, 3211, 3203-4, or 5203 Printer, its Forms-Control and Universal-Character-Set buffers (FCBs and UCBs) are loaded automatically during IPL. An FCB is loaded with the standard FCB image, while a UCB is loaded with the buffer image for an A11 (3211) or AN (3203, 3203-4 and 5203) print train. Each printer must be prepared as follows before IPL:

1. Restore the carriage to align the FCB to channel 1.
2. If you perform IPL after power on, adjust the paper in the carriage to the line-1 print position, that is, align the upper margin of the first form with the upper margin of the print train.
3. Ready the printer.

Note that for a 5203 printer without the UCS feature, only the FCB is loaded.

IPL with the IBM 3210 or 3215 Printer-Keyboard

2

If you intend to create an SDL, see the IPL procedure with a card reader (Procedure 5). IPL with the card reader is also faster than using the printer keyboard if you have many ADD and DEL commands.

1. Switch the power on as described in the appropriate hardware manual.

2. Ready the device containing SYSRES. If the device has removable disks, you must first mount the pack containing SYSRES.

3. Ready the device assigned to SYSVIS that contains the page data set. If the device assigned to SYSVIS has removable disks, you must first mount the pack containing the page data set. If the standard assignment for SYSVIS does not exist or is not to be used, any disk drive can be chosen; the physical address of the drive must then be specified in the DPD command. If the page data set resides on a 3340 Disk Storage, this device must be ready before you start the IPL procedure.

4. Set the load-unit switches on the system control panel to the physical device address of the disk drive that holds the SYSRES pack.

5. Press LOAD on the system control panel.

6. When the WAIT light comes on, press REQUEST on the printer-keyboard. The system will respond with the following information message identifying the SYSRES file and CPU:

   ```
   0104I IPLDEV=devaddr,VOLSER=volserno,CPUID=CPU-id
   ```

   Then the following message is printed:

   ```
   0I03A SPECIFY SUPERVISOR NAME
   ```

   If you wish to use the default supervisor ($$A$SUP1) press END; otherwise, enter the name of the required supervisor and then press END.

7. When the WAIT light comes on again, press REQUEST on the printer-keyboard.

   One of the following sets of messages will then be printed:

   A.
   ```
   0I30I DATE=date,CLOCK=time,ZONE=difference
   0I10A GIVE IPL COMMANDS
   ```

   B.
   ```
   0I31A DATE REQUIRED, CLOCK REQUIRED,
         ZONE=difference
   0I10A GIVE IPL COMMANDS
   ```

   C.
   ```
   0I32I TOD CLOCK INOPERATIVE; NO TOD SUPPORT
   0I31A DATE REQUIRED, CLOCK REQUIRED,
         ZONE=difference
   0I10A GIVE IPL COMMANDS
   ```

8. Enter DEL and ADD commands, if necessary.

9. Depending on the messages that were printed on SYSLOG (see step 7), take the following action:

A. 1. If all values are satisfactory, enter the SET command without parameters.

2. If the date or time of day is not satisfactory, enter the SET command with both DATE and CLOCK parameters, and press the TOD CLK switch to the ENABLE SET position.

3. If the zone is not satisfactory, enter the SET command with the ZONE parameter.

4. If none of the values is satisfactory, enter the SET command with all parameters and press the TOD CLK switch to the ENABLE SET position.

B. 1. If the zone value is satisfactory, enter the SET command with DATE and CLOCK parameters, and press the TOD CLK switch to the ENABLE SET position.

2. If the zone value is not satisfactory, enter the SET command with all parameters and press the TOD CLK switch to the ENABLE SET position.

C. Take the same action as in B above.

10. Enter the CAT command, if required, to indicate on which physical device the disk pack containing the VSAM catalog is mounted.

11. Enter the DPD command to define the page data set. DPD is mandatory; all operands are optional.

12. Press END on the printer-keyboard. The system then issues the message

```
0I20I IPL COMPLETE FOR DOS/VS REL xx.x ECLEVEL=nn
      [SUPVR USERID is: id]
```

in which case you can go to steps 13 and 14; or it issues the messages

```
0I20I IPL COMPLETE FOR DOS/VS REL xx.x ECLEVEL=nn
      [SUPVR USERID is: id]
1T00A WARM START COPY OF SVA FOUND
```

There are three possible responses:

A. Enter KEEP, if you wish to keep the current copy of the SVA (Shared Virtual Area); in this case, steps 13 and 14 cannot be executed.

B. Press END on the printer-keyboard. This has the same effect as A, above.

C. Enter REJ, if you do not wish to keep the current copy of the SVA; in this case you can go to steps 13 and 14.

13. If the SVA option was not specified during system generation, or if you wish to change the size of the existing SVA, enter the SET SVA=(nK,nK) job control command.

14. If you wish to use one of the standard procedures provided by IBM, do one of the following:

    a. If you do not need VSAM modules, enter the command

    ```
    EXEC PROC=SDL
    ```

    This procedure builds a system directory list containing the names of frequently used non-VSAM modules and phases.

    b. If you need VSAM modules, enter the command

    ```
    EXEC PROC=VSAMSVA
    ```

    This procedure creates a system directory list of the VSAM modules, in addition to those phases otherwise entered by the procedure SDL. It also loads these VSAM modules into the SVA.

    c. If you need RPS modules, enter the command

    ```
    EXEC PROC=RPS
    ```

    This procedure creates a system directory list of the RPS modules, in addition to those modules otherwise entered through SDL. It also loads the RPS modules into the SVA.

    d. If you need VSAM **and** RPS modules, enter

    ```
    EXEC PROC=VSAMRPS
    ```

    This procedure creates a system directory list of both the VSAM and the RPS modules (together with those entered through SDL) and also loads these modules into the SVA.

Does your system use RDE?

If so, turn to Procedure 8.

1. Perform the power on and load microprogram procedures as described in the appropriate hardware manual and wait until PROGRAM LOAD appears on the screen.

2. Ready the device containing SYSRES. If the device has removable disks, you must first mount the pack containing SYSRES.

3. Ready the device assigned to SYSVIS that contains the page data set. If the device assigned to SYSVIS has removable disks, you must first mount the pack containing the page data set. If the standard assignment for SYSVIS does not exist or is not to be used, any disk drive can be chosen for the pack; the physical address of the drive must then be specified in the DPD command. If the page data set resides on a 3340 Disk Storage, this device must be ready before you start the IPL procedure.

4. Type in the physical device address of the disk drive that holds the SYSRES disk pack.

5. Type in character C in order to clear storage. Only if during a hard wait you want to draw a stand-alone dump type in N instead to conserve storage contents.

   If you do not want to use any emulation press ENTER.

   If you are using 2311, or 2314 emulation (only with Model 125) specify the number of buffers needed.

6. When WAIT appears on the screen, press REQUEST. The system will respond with the following information message identifying the SYSRES file and CPU:

   ```
   0I04I IPLDEV=devaddr,VOLSER=volserno, CPUID=CPU-id
   ```

   Then IPL displays the following message:

   ```
   0I03A  SPECIFY SUPERVISOR NAME
   ```

   If you wish to use the default supervisor ($$A$SUP1), press ENTER; otherwise, enter the name of the required supervisor and then press ENTER.

7. When WAIT appears on the screen again, press REQUEST.

   One of the following sets of messages will then be displayed:

   A.
   ```
   0I30I  DATE=date,CLOCK=time,ZONE=difference
   0I10A GIVE IPL COMMANDS
   ```

   B.
   ```
   0I31A DATE REQUIRED, CLOCK REQUIRED,
         ZONE=difference
   0I10A GIVE IPL COMMANDS
   ```

   C.
   ```
   0I32I  TOD CLOCK INOPERATIVE; NO TOD SUPPORT
   0I31A DATE REQUIRED, CLOCK REQUIRED
   ```

When using the string switch feature on the models 115 or 125, neither CPU should be IPL'ed without first stopping the other CPU. (Press the stop button and wait for stop light 'on' before you begin the IPL.)

If you intend to create an SDL, see the IPL procedure with the card reader (Procedure 5). IPL with the card reader is also faster than using the DOC if you have many ADD and DEL commands.

0I10A GIVE IPL COMMANDS

8. Depending on the messages that were displayed on SYSLOG (see step 7), take the following action:

   A. 1. If all values are satisfactory, enter the SET command without parameters.

      2. If the date or time of day is not satisfactory, enter the SET command with both DATE and CLOCK parameters, and press TOD CLK.

      3. If the zone is not satisfactory, enter the SET command with the ZONE parameter.

      4. If none of the values is satisfactory, enter the SET command with all parameters and press TOD CLK.

   B. 1. If the zone value is satisfactory, enter the SET command with DATE and CLOCK parameters, and press TOD CLK.

      2. If the zone value is not satisfactory, enter the SET command with all parameters and press TOD CLK.

   C. If the message is 0I31A, then take the same action as in B above.

9. Enter the CAT command, if required, to indicate on which physical device the disk pack containing the VSAM catalog is mounted.

10. Enter the DPD command to define the page data set. DPD is mandatory; all operands are optional.

11. Press ENTER. The system then issues the message

    0I20I IPL COMPLETE FOR DOS/VS REL xx.x ECLEVEL=nn
    [SUPVR USERID IS: id]

    in which case you can go to steps 12 and 13 or it issues the messages

    0I20I IPL COMPLETE FOR DOS/VS REL xx.x ECLEVEL=nn
    [SUPVR USERID IS: id]
    1T00A WARM START COPY OF SVA FOUND

    There are three possible responses:

    A. Enter KEEP and press ENTER if you wish to keep the current copy of the SVA (Shared Virtual Area); in this case, steps 12 and 13 cannot be executed.

    B. Press ENTER. This has the same effect as A, above.

    C. Enter REJ and press ENTER if you do not wish to keep the current copy of the SVA; in this case you can go to steps 12 and 13.

12. If you wish to change the size of the existing SVA, enter the SET SVA=(nK,nK) job control command.
13. If you wish to use one of the standard procedures provided by IBM, do one of the following:

    a. If you do not need VSAM modules, enter the command

    ```
    EXEC PROC=SDL
    ```

    This procedure builds a system directory list containing the names of the frequently used non-VSAM modules and phases.

    b. If you need VSAM modules, enter the command

    ```
    EXEC PROC=VSAMSVA
    ```

    This procedure creates a system directory list of the VSAM modules, in addition to those phases otherwise entered by the procedure SDL. It also loads these VSAM modules into the SVA.

    c. If you need RPS modules, enter the command

    ```
    EXEC PROC=RPS
    ```

    This procedure creates a system directory list of the RPS modules, in addition to those modules otherwise entered through SDL. It also loads the RPS modules into the SVA.

    d. If you need VSAM **and** RPS modules, enter

    ```
    EXEC PROC=VSAMRPS
    ```

    This procedure creates a system directory list of both the VSAM and the RPS modules (together with those entered through SDL) and also loads the RPS modules into the SVA.

Does your system use RDE?

If so, turn to Procedure 8.

IPL with the IBM 3277

4

If you intend to create an SDL, see the IPL procedure with a card reader (Procedure 5). IPL with the card reader is also faster than using the 3277 if you have many ADD and DEL commands.

1. Switch the power on as described in the appropriate hardware manual.

2. Ready the device containing SYSRES. If the device has removable disks, you must first mount the pack containing SYSRES.

3. Ready the device assigned to SYSVIS that contains the page data set. If the device assigned to SYSVIS has removable disks, you must first mount the pack containing the page data set. If the standard assignment for SYSVIS does not exist or is not to be used, any disk drive can be chosen; the physical address of the drive must then be specified in the DPD command. If the page data set resides on a 3340 Disk Storage, this device must be ready before you start the IPL procedure.

4. Set the load-unit switches on the system control panel to the physical device address of the disk drive that holds the SYSRES pack.

5. Press LOAD on the system control panel.

6. When the WAIT light comes on, press ENTER on the 3277. The system will respond with the following information message identifying the SYSRES file and CPU:

   ```
   0104I IPLDEV=devaddr,VOLSER=volserno,
   CPUID=CPU-id
   ```

   Then the following message is displayed:

   ```
   0103A SPECIFY SUPERVISOR NAME
   ```

   If you wish to use the default supervisor ($$A$SUP1) press ENTER; otherwise, enter the name of the required supervisor and then press ENTER.

7. When the WAIT light comes on again, press ENTER on the 3277.

   One of the following sets of messages will then be displayed:

   ```
   A.  0130I DATE=date,CLOCK=time,ZONE=difference
       0110A GIVE IPL COMMANDS

   B.  0131A DATE REQUIRED, CLOCK REQUIRED,
             ZONE=difference
       0110A GIVE IPL COMMANDS

   C.  0132I TOD CLOCK INOPERATIVE; NO TOD SUPPORT
       0131A DATE REQUIRED, CLOCK REQUIRED,
             ZONE=difference
       0110A GIVE IPL COMMANDS
   ```

8. Enter DEL and ADD commands, if necessary.

9. Depending on the messages that were displayed on SYSLOG (see step 7), take the following action:

A. 1. If all values are satisfactory, enter the SET command without parameters.

2. If the date or time of day is not satisfactory, enter the SET command with both DATE and CLOCK parameters, and press the TOD CLK switch to the ENABLE SET position.

3. If the zone is not satisfactory, enter the SET command with the ZONE parameter.

4. If none of the values is satisfactory, enter the SET command with all parameters and press the TOD CLK switch to the ENABLE SET position.

B. 1. If the zone value is satisfactory, enter the SET command with DATE and CLOCK parameters, and press the TOD CLK switch to the ENABLE SET position.

2. If the zone value is not satisfactory, enter the SET command with all parameters and press the TOD CLK switch to the ENABLE SET position.

C. Take the same action as in B above.

10. Enter the CAT command, if required, to indicate on which physical device the disk pack containing the VSAM catalog is mounted.

11. Enter the DPD command to define the page data set. DPD is mandatory; all operands are optional.

12. Press ENTER. The system then issues the message

```
0I20I IPL COMPLETE FOR DOS/VS REL xx.x ECLEVEL=nn
      [SUPVR USERID IS: id]
```

in which case you can go to steps 13 and 14; or it issues the message

```
0I20I IPL COMPLETE FOR DOS/VS REL xx.x ECLEVEL=nn
      [SUPVR USERID IS: id]
1T00A WARM START COPY OF SVA FOUND
```

There three possible responses:

A. Enter KEEP, if you wish to keep the current copy of the SVA (Shared Virtual Area); in this case, steps 13 and 14 cannot be executed.

B. Press ENTER. This has the same effect as A, above.

C. Enter REJ, if you do not wish to keep the current copy of the SVA; in this case you can go to steps 13 and 14.

13. If the SVA option was not specified during system generation, or if you wish to change the size of the existing SVA, enter the SET SVA(nK,nK) job control command.

14. If you wish to use one of the standard procedures provided by IBM, do one of the following:

    a. If you do not need VSAM modules, enter the command

    ```
    EXEC PROC=SDL
    ```

    This procedure builds a system directory list containing the names of the frequently used non-VSAM modules and phases.

    b. If you need VSAM modules, enter the command

    ```
    EXEC PROC=VSAMSVA
    ```

    This procedure creates a system directory list of the VSAM modules, in addition to those phases otherwise entered by the procedure SDL. It also loads these VSAM modules into the SVA.

    c. If you need RPS modules, enter the command

    ```
    EXEC PROC=RPS
    ```

    This procedure creates a system directory list of the RPS modules, in addition to those modules otherwise entered through SDL. It also loads the RPS modules into the SVA.

    d. If you need VSAM **and** RPS modules, enter

    ```
    EXEC PROC=VSAMRPS
    ```

    This procedure creates a system directory list of both the VSAM and the RPS modules (together with those entered through SDL) and also loads the RPS modules into the SVA.

> Does your system use RDE?
>
> If so, turn to Procedure 8.

IPL with the Card Reader

5

When using the string switch feature on the models 115 or 125, neither CPU should be IPL'ed without first stopping the other CPU. (Press the stop buttom and wait for stop light 'on' before you begin the IPL.)

If you have many ADD and DEL commands, or if you must create an SDL, this procedure is faster than the one using the printer keyboard or the display console.

1. Perform steps 1 to 6 of procedure 2 or 3, depending on the CPU model of your system. The supervisor name must always be transmitted via the system console (SYSLOG).

2. Place DEL, ADD and CAT cards (optional) and SET and DPD (mandatory) in the hopper of the card reader.

3. Start the card reader in one of the following ways:

   a. If the card reader is assigned to SYSRDR:

      - For CPU models 115 and 125 press the interrupt key when WAIT appears on the screen.
      - For other models press the interrupt key when the WAIT light goes on.

   b. If the card reader is not assigned to SYSRDR:

      - For CPU models 115 and 125 press START and EOF on the card reader when WAIT appears on the screen.
      - For all other models press START and EOF on the card reader when the WAIT light goes on.

   Note: If the card reader was used to enter the supervisor name and is still ready, press STOP before pressing START.

   When the IPL commands have been read, one of the following will happen, depending on the system you are using:

   - If the time of day (TOD) clock is in the 'not set' or 'error' state, the system enters a hard-wait state and message code 0131A is displayed in bytes 0-4 of real storage. You must provide the SET command with DATE and CLOCK parameters, repeat the IPL procedure, and press the TOD clock switch to the ENABLE SET position at the time specified in the CLOCK parameter.
   - If the TOD clock is not operational, message code 0132A is displayed in bytes 0-4 of real storage and the system enters a hard-wait state. In this case you must perform the IPL procedure from the printer-keyboard or display console.
   - If the TOD clock is in the 'set' state or has the proper values, the system issues message

     0120I IPL COMPLETE FOR DOS/VS REL xx.x ECLEVEL=nn

     [SUPVR USERID IS: id]

     This message may or may not be followed by message

     1T00A WARM START COPY OF SVA FOUND

4. There are three possible responses to message 1T00A, which are to be entered via the console typewriter:

   a. Enter KEEP if you wish to keep the current copy of the SVA (shared virtual area); in this case, your IPL deck must not include the cards mentioned under steps 5 through 7.

   b. Simply press END/ENTER. This has the same effect as A, above.

   c. Enter REJ if you do not wish to keep the current copy of the SVA; in this case, your IPL deck should include some or all of the cards mentioned under steps 5 through 7 depending on your installation requirements.

5. If the size of the existing SVA is to be changed, submit (through the card read to which SYSRDR is assigned) the SET SVA=(nK,nK) job control command.

6. If a system directory list (SDL) is to be created, submit (through the card reader to which SYSRDR is assigned) the command

   SET SDL=CREATE

   followed by a list of phase names in the format

   phasename[,SVA]

   A /* card must follow the last phase name.

7. If you wish to use one of the standard procedures provided by IBM, do one of the following:

   a. If you need no VSAM modules, submit the command

   EXEC PROC=SDL

   This procedure builds a system directory list containing the names of frequently used non-VSAM modules and phases.

   b. If you need VSAM modules, submit the command

   EXEC PROC=VSAMSVA

   This procedure creates a system directory list of the VSAM modules, in addition to those phases otherwise entered by the procedure SDL. It also loads these VSAM modules into the SVA.

   c. If you need RPS modules, enter the command

   EXEC PROC=RPS

   This procedure creates a system directory list of the RPS modules, in addition to those modules otherwise entered through SDL. It also loads the RPS modules into the SVA.

   d. If you need VSAM **and** RPS modules, enter

   EXEC PROC=VSAMRPS

   This procedure creates a system directory list of both the VSAM and the RPS modules (together with those entered through SDL) and also loads these modules into the SVA.

Does your system use RDE?

If so, turn to Procedure 8.

(For a description on how to prepare the diskette, refer to the end of this procedure.)

1. Perform steps 1-6 of procedure 2 or 3, depending on the CPU model of your system.

   Note: the supervisor name must always be transmitted via the system console (SYSLOG).

2. Place the diskette in the hopper of the 3540 and make the device ready by pressing the START key. Now IPL will read IPL commands from diskette.

3. The system will respond with the following information message identifying the SYSRES file and CPU:

   ```
   0I04I IPLDEV=devaddr,VOLSER=volserno,CPUID=cpu-id
   ```

   When the IPL commands have been read, one of the following will happen, depending on the system you are using:

   - If the time of day (TOD) clock is in the 'not set' or 'error' state, the system enters a hard-wait state and message code 0I31A is displayed in bytes 0-4 of real storage. You must provide the SET command with DATE and CLOCK parameters, repeat the IPL procedure, and press the TOD clock switch to the ENABLE SET position at the time specified in the CLOCK parameter.
   - If the TOD clock is not operational, message code 0I32A is displayed in bytes 0-4 of real storage and the system enters a hard-wait state. In this case you must perform the IPL procedure from the printer-keyboard or display console.
   - If the TOD clock is in the 'set' state or has the proper values, the system issues message

     ```
     0I20I IPL COMPLETE FOR DOS/VS REL xx.x ECLEVEL=nn
         [SUPVR USERID IS: id]
     ```

     This message may or may not be followed by message

     ```
     1T00A WARM START COPY OF SVA FOUND
     ```

4. There are three possible responses to message 1T00A, which are to be entered via SYSLOG:

   a. Enter KEEP if you wish to keep the current copy of the SVA (shared virtual area); in this case, your IPL deck must not include the cards mentioned under steps 5 through 7.

   b. Simply press END/ENTER. This has the same effect as A, above.

   c. Enter REJ if you do not wish to keep the current copy of the SVA; in this case, your IPL deck should include some or all of the cards mentioned under steps 5 through 7 depending on your installation requirements.

5. If the size of the existing SVA is to be changed, submit (through the card reader to which SYSRDR is assigned) the SET SVA=(nK,nK) job control command.

6. If a system directory list (SDL) is to be created, submit (through the card reader to which SYSRDR is assigned) the command

6

```
SET SDL=CREATE
```

followed by a list of phase names in the format

```
phasename[,SVA]
```

A /* card must follow the last phase name.

7. If you wish to use one of the standard procedures provided by IBM, do one of the following:

   a. If you do not need VSAM modules, enter the command

      ```
      EXEC PROC=SDL
      ```

      This procedure builds a system directory list containing the names of frequently used non-VSAM modules and phases.

   b. If you need VSAM modules, enter the command

      ```
      EXEC PROC=VSAMSVA
      ```

      This procedure creates a system directory list of the VSAM modules, in addition to those phases otherwise entered by the procedure SDL. It also loads these VSAM modules into the SVA.

   c. If you need RPS modules, enter the command

      ```
      EXEC PROC=RPS
      ```

      This procedure creates a system directory list of the RPS modules, in addition to those modules otherwise entered through SDL. It also loads the RPS modules into the SVA.

   d. If you need VSAM **and** RPS modules, enter

      ```
      EXEC PROC=VSAMRPS
      ```

      This procedure creates a system directory list of both the VSAM and the RPS modules (together with those entered through SDL) and also loads these modules into the SVA.

Prepare diskette using the IBM 3741

(For details, refer to *IBM Data Station Operator's Guide*, GA21-9131.)

a) Create the data set label:

   Data set name (position 6-10) = IJIPL
   Record length (position 23-27) = 00080
   Beginning of extent (position 29-33): e.g. 01001
   End of extent (position 35-39): e.g. 01009
   End of data (position 75-79): same as position 29-33

b) Enter IPL commands (ADD, DEL, etc.), one per record.

IPL with the IBM Model 158 Display Console

7

If you intend to create an SDL, see the IPL procedure with a card reader (Procedure 5). IPL with the card reader is also faster than using the IBM Model 158 console if you have many ADD and DEL commands.

It is assumed that you are familiar with the functional characteristics of the Model 158 display console as described in *System/370 Model 158 Operating Procedures*, GC38-0025. Functionally, there is no difference between that console and a 3277 Display Console.

1. Press POWER ON (this assumes that the operations console file has been inserted in the console's diskette reader).
2. Select MANUAL. This causes the manual frame to be displayed.
3. Select L-LOAD UA.
4. Enter the load unit address using the hex input matrix on the display screen.
5. Select 4-LOAD under O-OPERATOR FUNCTIONS.
6. Select X-EXECUTE. This causes the program frame and the WAIT system indicator to be displayed.
7. Press REQ (request). The system will respond with the following information message identifying the SYSRES file and CPU:

   ```
   0I04I IPLDEV=devaddr,VOLSER=volserno,CPUID=CPU-id
   ```

   Then IPL displays the following message:

   ```
   0I03A SPECIFY SUPERVISOR NAME
   ```

   If you wish to use the default supervisor ($$A$SUP1) press END. Otherwise enter the name of the desired supervisor, then press END.

The remainder of the procedure is the same as steps 7 through 14 of the IPL procedure with a 3210 or 3215 printer-keyboard (Procedure 2).

Does your system use RDE?

If so, turn to Procedure 8.

RDE Data Entry After IPL

8

RDE information enables your operations manager or system programmer to minimize recurrence of errors.

If the supervisor at your installation was generated to support the reliability data extractor (RDE), the message

1189A IPL REASON CODE =

appears on SYSLOG as soon as the first // JOB statement is processed after IPL. You must then enter a two-character code, which indicates the reason why IPL was performed. For IPL reason codes refer to the section *Reference Information.*

If you enter an invalid code, the message

1192I INVALID CODE

appears on SYSLOG and message

1189A IPL REASON CODE =

is reissued until you have entered a valid code. When you have entered a valid reason code, the message

1191A SUB-SYSTEM ID =

appears on SYSLOG. You must then enter a two-character ID code. For IPL ID codes refer to the section *Reference Information.*

The ROD command has no operands. Simply enter ROD and press END or ENTER.

Before shutting down at the end of your working day you must issue the ROD command. This command causes, among other things, the RDE information to be written on the disk pack assigned to SYSREC. You can obtain a listing of the SYSREC file by running the EREP program as described in *DOS/VS SADP*, GC33-5380.

Creating the System Recorder File

The DOS/VS Recovery Management Support Recorder (RMSR) requires a disk extent on which to record statistical information on machine errors. This disk extent is called the system recorder file and is identified by the symbolic name SYSREC. The SYSREC file must be created after the **first** IPL procedure only (**not** after each IPL). If, however, the SYSREC file is damaged, you must re-IPL and recreate the system recorder file.

The Models 115 and 125 without software recording require no system recorder file.

The following job stream creates the system recorder file. The commands and statements in the shaded area are included to show the proper placement of the statements and commands that create the recorder file.

```
0I10A  GIVE IPL COMMANDS

       DEL
       ADD
       SET
       CAT
       DPD

0I20I  IPL COMPLETE FOR DOS/VS REL xx.x ECLEVEL=nn
1I00A  READY FOR COMMUNICATIONS
ASSGN SYSREC,DISK,VOL=222222,SHR
SET RF=CREATE
// OPTION STDLABEL
// DLBL IJSYSRC,'DOS.SYSTEM.RMSR.FILE'
// EXTENT SYSREC,nnnnnn,,,nnnnn,nnnnn
.
. (other label information for this disk pack)
.
// JOB FIRST
```

First statement of the normal job stream. When this statement is read, the system recorder file is created and opened.

The extent information for the // EXTENT statement is supplied by your system programmer.

To obtain a listing of the SYSREC file, run the EREP program as described in *DOS/VS SADP*, GC33-5380. During execution of EREP, recording on SYSREC is suppressed.

Creating the Hard-Copy File

10

The hard-copy file does not replace the system recorder file.

If a display console is operated without a console printer, a disk extent is required which is used for recording all information on the screen, both from the system and from the operator. This disk extent is called the hard-copy file and resides on the device assigned to SYSREC.

The hard-copy file must be created after the **first** IPL procedure only (**not** after each IPL). If, however, the IPL device is changed or the SYSREC file is damaged, you must re-IPL and recreate the hard-copy file.

The following job stream is an example of how to create the hard-copy file. The commands and statements in the shaded area are included to show the proper placement of the statements and commands that create the hard-copy file.

```
0I10A  GIVE IPL COMMANDS

       DEL
       ADD
       SET
       CAT
       DPD

0I20I  IPL COMPLETE FOR DOS/VS REL xx.x ECLEVEL=nn
1I00A  READY FOR COMMUNICATIONS
ASSGN SYSREC,DISK,VOL=222222,SHR
SET HC=CREATE
// OPTION STDLABEL
// DLBL IJSYSCN,'DOS HARD COPY FILE'
// EXTENT SYSREC,nnnnnn,,,nnnnn,nnnnn
.
. (other label information for this disk pack)
.
// JOB
// JOB FIRST
```

Dummy JOB card required to create and open the hard-copy file.

The extent information for the // EXTENT statement is supplied by your system programmer.

Printing the hard-copy file is described in Procedure 24.

When a new hard-copy file is created, the records of the old hard-copy file are lost unless they are printed first.

Loading Print Buffers

11

Applies to all buffer-controlled IBM printers except the 1403 UCS printer.

A job may require printed output in a nonstandard page layout or with the use of a nonstandard print train or both. If the printer is buffer-controlled, you must load the appropriate control buffer image into the forms control buffer (FCB) for output in a nonstandard page layout and into the universal character set buffer (UCB) for output with a UCS print train.

There are two ways of loading these buffers: (1) by having the system execute the SYSBUFLD program and (2) by using attention commands. In addition, the FCB can be loaded dynamically under the control of a problem program that is being executed. In this case, you merely have to service the particular printer in response to appropriate system messages.

Use caution when loading the buffers via attention commands while a printer is in operation, because it is impossible for you to predict the end of the output that is being printed with the current buffer image. For a printer in operation, it is recommended that you use this command if, for example, printed output for a program was started with the wrong FCB image and you are able to correct the output by issuing the LFCB command.

Loading the FCB by Using the SYSBUFLD Program

You need one such job stream for each printer whose FCB is to be loaded.

1. Enter the following control statements via SYSIN:

```
// JOB name
// EXEC SYSBUFLD
   FCB SYSxxx,phasename[,NULMSG]
/*
/&
```

   The operands of the FCB statement are the same as the corresponding operands of the LFCB command. For a discussion of these operands, see *Operator Commands* in *Section 4. Reference Information.*

Load the FCB-Execution of SYSBUFLD

2. The system issues message 1B21A after it has loaded the new FCB image.

   Note: If a change of forms is not required, skip steps 4, 6, and 7.

PRT1 printer: 3211 or 3203-4

3. If the subsequent print job requires a different setting of the carriage clutch, change the setting now (does not apply to a PRT1 printer).
4. Stop the printer.
5. Press END/ENTER.
6. To align the new forms, place the upper margin of the first new form exactly over that fold of the currently used forms which has just passed the print train.
7. Ready the printer.

11

An example of loading both the FCB and the UCB in the same job entry is given under *Loading the UCB under POWER/VS.*

Loading the FCB via POWER/VS JECL

By placing the FCB=phasename parameter in a POWER/VS LST statement, POWER/VS will load the phase from the Core Image Library, check it for validity, convert the carriage information to LTAB specification and then, if the LST card applies to an FCB type printer, cause this FCB to be loaded on the printer when the output is actually printed (providing the actual printer used is of the same type as specified at execution time.

Example:

```
* $$ LST FNO=BLUE,FCB=$$BFCB3
```

Note: If the FCB=phasename parameter is omitted from a LST statement and that LST statement applies to an FCB type printer, the following FCB name will default depending on type of printer:

3203	-	$$BFCB3
3203-4	-	$$BFCB00
3211	-	$$BFCB
5203	-	$$BFCB5

The phasename is the same as is loaded by the DOS/VS IPL routines.

Loading the FCB within a POWER/VS Job

The job step to load an FCB under POWER/VS may be placed anywhere within a POWER/VS job. When POWER/VS prints the output of that job and finds the load FCB request, the current forms are aligned to line 1 and the buffer is loaded. Note that line 1 and the channel-1 line may be different. If the load request was preceded by a * $$ LST statement specifying a form number, then POWER/VS displays message

```
1Q40A ON cuu FORMS ffff NEEDED FOR jobname number
```

After receiving this message, proceed as follows:

- Place the forms indicated in the message on the printer and align them using the PSETUP command. This command allows you to adjust forms alignment during printing.
- Enter the command

  ```
  G cuu
  ```

  where cuu is the address of the printer.

In message 1Q40A, ffff is the number specified in the forms-number parameter of the * $$ PRT statement. If several FCB load operations are executed in one job entry, the message 1Q40A is printed for each load operation.

Loading the FCB as a Separate POWER/VS Job

This procedure allows you to change the contents of the FCB whenever required. Submit the following POWER/VS job in any partition

The JECL statements (* $$ in columns 1 to 4) are discussed in detail in *DOS/VS POWER/VS Installation Guide and Reference*, SH12-5430.

that is waiting for work:

```
* $$ JOB JNM=FCB
* $$ LST DISP=D,CLASS=A,FORMS=A123
// JOB FCB
// EXEC SYSBUFLD
   FCB SYSLST,phasename
/&
* $$ EOJ
```

Then enter the command

```
S LST,cuu
```

The FCB will now be loaded as described under *Loading the FCB Within a POWER/VS Job.*

Using the LFCB command

Loading the FCB by Using the LFCB Command

Execution of the LFCB command temporarily suspends all other I/O operations and should therefore be used with caution, especially in an installation using TP or real-time equipment, such as the IBM 1275 or 1419.

1. Enter the LFCB command (for details about this command, see *Operator Commands* in *Section 4. Reference Information*). The format of the command is:

   ```
   LFCB X'cuu',phasename[,FORMS=xxxx]
         [,LPI=n][,NULMSG]
   ```

   Note: If neither FORMS=xxxx nor LPI=n is specified, the system does not return a message and no further operator action is required.

2. If operator intervention is necessary and you issued the command for a printer other than a PRT1 printer, the system prints message 1B13A. If you issued the command for a PRT1 printer, the system prints message 1B14A.

   PRT1 printer: 3211 or 3203-4

   Respond to message 1B13A in the following way:

   a. Stop the printer (only if forms must be changed).
   b. Change the setting of the carriage clutch (if necessary).
   c. Press END/ENTER (omit steps d and e if forms need not be changed).
   d. Place the new forms on the printer and align line 1 of the first new form with the print line.
   e. Ready the printer.

   Respond to message 1B14A in the following way:

   a. Stop the printer.
   b. Press END/ENTER.
   c. Place the new forms on the printer and align line 1 of the first new form with the print line.
   d. Ready the printer.

11

Load the UCB

You need one such job stream for each printer whose UCB is to be loaded.

Loading the UCB by Using the SYSBUFLD Program

1. Mount the new print train on the printer and ready the device.
2. Enter the following control statements on the card reader:

```
// JOB name
// EXEC SYSBUFLD
   UCB SYSxxx,phasename[,FOLD][,NOCHK][,NULMSG]
/*
/&
```

The operands of the UCB statement have the same purpose as the corresponding operands of the LUCB command. For details refer to *Operator Commands* in *Section 4. Reference Information.*

Execution of SYSBUFLD under POWER/VS

Loading the UCB under POWER/VS

The procedure for loading the UCB under POWER is the same as that for loading the FCB, except that no message is issued to the operator unless an error occurs. An example of loading the UCB as a separate job entry follows:

```
* $$ JOB JNM=UCB
* $$ LST DISP=D,CLASS=A
// JOB UCB
// EXEC SYSBUFLD
   UCB SYSLST,phasename
/&
* $$ EOJ
```

Loading FCB and UCB

If you need to load both the UCB and the FCB together, the following example shows how this can be done within the same job entry:

```
*  $$ JOB  JNM=UCB/FCB
*  $$ LST  DISP=D,CLASS=A,FORMS=A123,UCS=UCB-name,
             FCB=FCB-name
*  $$ EOJ
```

Using the LUCB command (applies also to 1403 with the UCS feature)

Loading the UCB by Using the LUCB Command

1. Enter the LUCB command (for details about this command, see *Operator Commands* in *Section 4. Reference Information*). The format of the command is:

```
LUCB X'cuu',phasename[,FOLD][,NOCHK]
     [,TRAIN=xxxxxx][,NULMSG]
```

   If you specified TRAIN=xxxxxx in your command, the system responds with message 1B18A.

   Note: Steps 2 to 4 are necessary only if a new print train (or chain) must be mounted on the printer.

Execution of the LUCB command temporarily suspends all other I/O operations and should therefore be used with caution, especially in an installation using TP or real-time equipment, such as the IBM 1275 or 1419.

2. Stop the printer.
3. Press END/ENTER.
4. Mount the new print train (or chain) and ready the printer again.

If you did not specify NULMSG, the system prints the buffer load verification message, and causes an additional skip-to-1 operation.

Loading the UCB for an IBM 1403 UCS Printer

12

The UCS command can be used only to load the UCB of the 1403 UCS printer.

Before the UCS (universal character set) feature of a 1403 printer can be used, the UCB in the 2821 control unit must be loaded with the appropriate buffer load.

There are two ways of loading the buffer: (1) using the UCS job control command or (2) using the LUCB attention command. The procedure to be applied when using the UCS job control command is described below. For the procedure to be applied when using the LUCB attention command, see Procedure 11.

Loading the Buffer Using the UCS Job Control Command

1. Mount the new print chain (or train) on the 1403.

2. Enter the following job control command on the console keyboard:

```
UCS SYSxxx,phasename[,FOLD][,BLOCK][,NULMSG]
```

The operands of the UCS command have the same purpose as the corresponding operands (BLOCK=NOCHK) of the LUCB command. For details refer to *Operator Commands* in *Section 4. Reference Information.*

Assigning I/O Devices

When you have started the system, the I/O device assignments are standard, which means that the assignments established during system generation are in effect.

You may have to change one or more of the standard device assignments and you can do this either by means of the ASSGN command or the // ASSGN job control statement.

If you use the job control command, the assignment is permanent, which means that it remains in effect until the next IPL procedure is performed, unless the command is superseded by another ASSGN command. If the ASSGN command has the TEMP option, the assignment is temporary.

If you use the job control statement, the assignment is temporary, which means that it is valid until the next // JOB statement, unless it is superseded by an ASSGN command or statement. If the // ASSGN statement has the PERM option, the assignment is permanent.

Example: To assign a printer with the physical device address 00E to the logical unit SYSLST, enter one of the following statements or commands.

Temporary Assignment:

```
        // ASSGN SYSLST,X'00E'
```

or

```
        ASSGN SYSLST,X'00E',TEMP
```

Permanent assignment:

```
        ASSGN SYSLST,X'00E'
```

or

```
        // ASSGN SYSLST,X'00E',PERM
```

The device address (X'00E' in the example) may be replaced by a generic name or device type, for example PRINTER or 1403. In that case, the X and the apostrophes must be omitted. You will find more information on how to use ASSGN in the section *Reference Information.*

If you are not certain about the current I/O device assignments at your installation, issue the LISTIO command, which prints all current assignments on SYSLOG. The // LISTIO job control statement prints all current assignments on SYSLST. Therefore, if you operate a Model 115 or 125, make a habit of using the // LISTIO statement, because the screen of the DOC may be too small to accommodate the full list of assignments.

Consider the following ASSGN statements and commands given for one job:

```
        // ASSGN SYSSLB,DISK,VOL=111111
        // ASSGN SYS000,DISK,VOL=222222
```

13

```
ASSGN SYS001,X'280',X'C0'
ASSGN SYS001,X'281',ALT
// ASSGN SYS002,X'280',X'C0'
// ASSGN SYS003,X'281'
// ASSGN SYS004,X'282',X'C0'
// LISTIO F1
```

The output of this LISTIO statement, produced on SYSLST, is shown in Figure III-1.

If a standard or permanent assignment is temporarily superseded, it is flagged STD in the comment (CMNT) column. The temporary assignment that superseded the assignment is listed one line higher (see SYSSLB in Figure III-1).

In the case of assignments that are temporary only, UA is printed between the channel and unit columns.

ALT in the comment column indicates that an alternate tape unit has been assigned to SYS001.

For the generic assignment of SYSSLB and SYS000, the system selected the devices X'131' and X'290', respectively.

You can use the RESET command to reset temporary I/O device assignments. This command resets temporary device assignments either to the last preceding permanent assignments, if any (made by an ASSGN command without the TEMP option), or to the standard assignment established during system generation. If neither a standard nor a permanent assignment exists, the temporary assignment is reset to UA (unassigned).

Resetting is performed only for the temporary assignments in the partition for which the RESET command is given.

```
            *** FOREGROUND 1 ***

I/O UNIT     CMNT   CHNL   UNIT   MODE

SYSRDR                0     0C
SYSIPT                0     0C
SYSPCH               ** UA **
SYSLST                0     CE
SYSLOG                0     1F
SYSLNK                1     31
SYSRES                1     30
SYSSLB                1     31
SYSSLB        STD     1     30
SYSRLB                1     30
SYSREC                1     30
SYSCLB               ** UA **

            *** FOREGROUND 1 ***

I/O UNIT     CMNT   CHNL   UNIT   MODE

SYS000                2     90
SYS000        STD    ** UA **
SYS001                2     80     C0
SYS001        ALT     2     81     C0
SYS002                2     80     C0
SYS002        STD    ** UA **
SYS003                2     81     C0
SYS003        STD    ** UA **
SYS004                2     82     C0
SYS004        STD    ** UA **
SYS005               ** UA **
SYS006               ** UA **
SYS007               ** UA **
SYS008               ** UA **
SYS009               ** UA **
SYS010               ** UA **
SYS011               ** UA **
SYS012               ** UA **
SYS013               ** UA **
SYS014               ** UA **
SYS015               ** UA **
SYS016               ** UA **
SYS017               ** UA **
SYS018               ** UA **
SYS019               ** UA **
SYS020               ** UA **
SYS021               ** UA **
SYS022               ** UA **
SYS023               ** UA **
SYS024               ** UA **
```

Figure III-1. Sample output of the LISTIO statement for F1

14

The size of the real and virtual partitions in a multiprogramming system is usually defined during system generation. To run a particular job, you may have to change these standard storage allocations.

ALLOC

ALLOCR

There are two commands - ALLOC and ALLOCR - that enable you to effect such changes. The ALLOC command allocates storage to virtual partitions, the ALLOCR command to real partitions.

There are a number of rules and restrictions that must be observed when using the ALLOC and ALLOCR commands. It is the responsibility of your programmer to give you precise instructions as to the procedure to follow when you have to change the size of one or more partitions.

A detailed description of the rules to be followed when using the ALLOC and ALLOCR commands is given in *DOS/VS System Control Statements*, GC33-5376.

MAP

Before the programmer can decide on the new allocations to be made, he may need a list of the current storage allocations in the system. You can obtain such a list by entering the MAP operator command. The MAP command produces a list on SYSLOG of the current sizes and starting addresses of all virtual and real partitions.

It is advisable to execute the MAP command again, after the new storage allocations have been made, to verify the correct function of the ALLOC/ALLOCR commands and to keep a record of the new system status.

Starting the Background Partition

After successful completion of the IPL procedure, the system is ready to accept jobs for processing.

Immediately after the message 0I20I IPL COMPLETE FOR DOS/VS REL xx.x ECLEVEL=nn, the partition indicator on SYSLOG is BG, indicating that the background partition is ready for processing.

Assume that you now want to run a job, or a series of jobs, in that partition. You must then prepare the I/O devices that are needed by the first (or only) job and ready the device that is to be used as SYSRDR.

If SYSRDR has already been assigned, the job will start as soon as you press END or ENTER. If not, the following message will appear on SYSLOG:

```
BG 1C10A PLEASE ASSIGN SYSRDR
```

Use the ASSGN command to assign SYSRDR and press END or ENTER. The partition indicator BG will now appear on SYSLOG, and pressing END or ENTER again will cause the job to be started.

As long as only the background partition is processing, most messages that appear on SYSLOG will refer to jobs in that partition (identifier BG), and it will be easy for you to respond to these messages. If you enter attention commands, you may also get messages from the attention routine (identifier AR).

Starting a Foreground Partition

16

Assume that you operate a multiprogramming system. The background partition is processing a job stream, and you now want to start a foreground partition.

To do this, press REQUEST, which gives you the attention routine identifier AR.

You may now start, for example, F1 by entering either one of the attention commands

BATCH and START have identical functions

```
BATCH F1   or   START F1
```

When you have pressed END or ENTER after the BATCH or START command, the partition identifier F1 appears on SYSLOG. If a standard assignment exists for SYSRDR and you need not enter any commands from SYSLOG, you can now ready the device assigned to SYSRDR, and press END or ENTER again. The system will then start processing the F1 job stream.

If you wish to enter job control statements or commands from SYSLOG before job-stream processing starts, you can do this as soon as the partition identifier F1 has appeared on SYSLOG.

You can obtain a list of assignments by entering the job control command LISTIO. If you enter

LISTIO:
output on SYSLOG
// LISTIO:
output on SYSLST
(recommended for DOC)

```
LISTIO ALL
```

you will get, on SYSLOG, a list of all current assignments for all partitions. To obtain BG and F1 assignments only, you must enter LISTIO BG, wait until the list of I/O assignments has been produced and the partition identifier F1 appears again on SYSLOG. You then enter LISTIO F1. When the list of assignments for F1 has been produced, the partition identifier F1 will again appear on SYSLOG.

Make sure that the assignments are correct; if necessary, make new assignments for F1, using the ASSGN command.

You may now enter job control statements or commands from SYSLOG. If you do not want to do this, ready SYSRDR and press END or ENTER. The program in F1 will then start reading from SYSRDR; processing of the F1 job stream begins.

You may start the other foreground partitions in the same way as described for F1.

When you have several partitions operating, messages from all partitions may appear on SYSLOG. All these messages are preceded by a partition identifier, which enables you to determine to which job stream they belong.

LOG
NOLOG

If you want all job control statements and commands within a job stream to be listed on SYSLOG, enter the LOG command. It remains in effect until you enter the NOLOG command. The LOG and NOLOG commands have no operands.

VTAM

If VTAM is to be used, refer to the starting procedures for VTAM described in *DOS/VS VTAM Network Operating Procedures*, GC27-6997.

Interrupting or Terminating Processing

17

Once a partition has been started, processing continues until the end of the job stream. It may happen, however, that you wish to temporarily suspend processing in a partition or to release a partition altogether.

You must suspend processing in a partition, for instance, if you have to mount tapes or disk packs between jobs or job steps. If you want to insert a job in the job stream at short notice, you may have to suspend processing twice: once at the end of the current job to insert the new one, and once at the end of the inserted job.

You have two possibilities to suspend processing in a partition temporarily:

// PAUSE

1. By means of a // PAUSE job control statement, which is inserted between the job control statements at the point where you want the interruption.

PAUSE

2. By means of a PAUSE attention command from SYSLOG for the desired partition, for example:

   ```
   PAUSE F3
   ```

   This will cause processing to be interrupted at the end of the current job step in F3. If you prefer to interrupt processing at the end of the current job, enter:

   ```
   PAUSE F3,EOJ
   ```

all three cases, you continue processing by pressing END or ENTER.

STOP

Example:

```
ASSGN SYSRDR,UA
ASSGN SYSIPT,UA
ASSGN SYSLST,UA
STOP
```

In certain cases you may wish to suspend processing in a partition and continue at some later time. You can achieve this by issuing the STOP job control command for that partition. The partition remains active, all assignments remain intact (unless you unassigned devices prior to the STOP command), and the job control program retains control of the partition. You can resume processing at any time by issuing a START or BATCH attention command for the partition.

UNBATCH

Example:

```
CLOSE SYSIN
CLOSE SYSLST
UNBATCH
```

If no further processing is to take place in a foreground partition, you should use the UNBATCH command. Since this command unassigns all logical devices for the partition, you must close any system files on disk, tape, or diskette (SYSRDR, SYSIPT, SYSIN, SYSPCH, SYSLST, SYSOUT) using the CLOSE command before you issue the UNBATCH command. UNBATCH releases the partition entirely. It makes the partition 'inactive'.

UNBATCH cannot be used for the background partition.

Using Cataloged Procedures

DOS/VS allows you to catalog procedures, that is sets of job control and linkage editor control statements with or without SYSIPT data in the procedure library. You can access this library whenever you need a procedure to complete or set up a job stream.

This section describes how to
- catalog a procedure into the procedure library,
- retrieve a procedure from the procedure library, and
- modify control statements of a cataloged procedure for the duration of one job.

Cataloging a Procedure

To catalog a procedure into the procedure library, you use the librarian program MAINT with a CATALP control statement:

```
Submit via  ───────────►{ // JOB jobname
SYSRDR or               { // EXEC MAINT
SYSLOG                  ┌    CATALP procedurename
                        │          .
                        │          .
                        │          .
Submit via              │    control statements to
SYSIPT     ────────────►┤    be cataloged
                        │          .
                        │          .
                        │          .
Submit via              │ /+
SYSRDR or               └ /*
SYSLOG     ──────────────►/&
```

The CATALP statement specifies the name of the procedure to be cataloged. The statements to be cataloged follow the CATALP statement. The end of the procedure is indicated by the /+ statement. You may catalog more than one procedure in one job. In that case, the next CATALP statement follows the /+ statement. The /* statement signals the end of input data.

Restrictions

A number of restrictions apply to cataloging procedures. For details, refer to *DOS/VS System Management Guide*, GC33-5371.

Example

The following example shows how to catalog the job control statements needed to execute a program PAYROLL. The name of the cataloged procedure is PRCPAY.

The symbolic name starting in column 73 of each cataloged statement is needed if you wish to modify the procedure at a later time (see *Modifying Cataloged Procedures*, later in this section).

18

A procedure must not contain control statements for more than one job

symbolic name column 73

```
// JOB CATPROC
// EXEC MAINT
   CATALP PRCPAY
// ASSGN SYS017,READER                    S1
// ASSGN SYS018,PUNCH                     S2
// ASSGN SYS019,PRINTER                   S3
// ASSGN SYS020,TAPE                      S4
// ASSGN SYS021,DISK,VOL=111111           S5
// TLBL TAPFLE,'FILE-IN'                  S6
// DLBL DSKFLE,'FILE-OUT'                 S7
// EXTENT SYS021,111111,1,0,200,50        S8
// EXEC PAYROLL                           S9
/+
/*
/&
```

In addition to job control and linkage editor control statements, a procedure may contain SYSIPT data, such as control statements for system utility and service programs, statements for compilers and object modules, and input data for user programs. Such statements are referred to as inline SYSIPT data. For the rules that apply to the cataloging of procedures with inline SYSIPT data, refer to *DOS/VS System Control Statements*, GC33-5376. They are not discussed here because usually you will get all the information you need from your programmer.

Note: A description of independent procedure names is to be found in *DOS/VS System Management Guide*, GC33-5371.

Including a Procedure in the Job Stream

To retrieve a cataloged procedure from the procedure library and include it in the job stream, you enter the following statements on SYSRDR:

```
// JOB jobname
// EXEC PROC=procname
/&
```

procname in the EXEC statement specifies the name of the procedure to be fetched.

To retrieve and execute the procedure PRCPAY from the procedure library via the card reader, enter

```
// JOB USER1
// EXEC PROC=PRCPAY
/&
```

If the cataloged procedure includes a JOB statement, you must not submit a JOB statement when retrieving the procedure.

Invoking a Procedure from the Console

You can also invoke and execute a procedure from the console keyboard. There are two typical situations where this is done:

1. To start an urgent job in a partition which is specifically reserved for such jobs, and which is not working at present.
2. To insert an urgent job as the next job in a running partition.

In case 1 there is normally no reader assigned to the partition, and the EXEC command should be used. This will return control to the console at the end of the procedure. For example, to execute the procedure PRCPAY, you would enter

```
// JOB USER
EXEC PROC=PRCPAY
```

At the end of the procedure the following messages are issued:

```
EOP PRCPAY
READY FOR COMMUNICATION
```

You can then enter an end-of-job (/&) and/or start a new job.

In case 2, you would use the // EXEC statement. Control is now returned automatically to the reader assigned to the partition, and processing continues normally.

Modifying Cataloged Procedures

It is possible to modify, add, and delete statements of a cataloged procedure temporarily, for the duration of one job. You do this by submitting modifier statements (also called overwrite statements) following the EXEC PROC= statement when you retrieve the procedure from the procedure library.

Permanent changes can be made to a procedure only by cataloging a new version of the procedure.

18

Symbolic names must be unique.

You cannot modify an unnamed statement.

To enable changes to be made to a procedure statement, both the modified and the modifier statements must have the **same symbolic name.** This name consists of a maximum of seven characters. These symbolic names are punched in columns 73 through 79 of the control statements. The modifier statements must be in the same sequence as the referenced procedure statements.

To indicate whether a statement is to be added, deleted, or overwritten, specify one of the following characters in column 80 of the modifier statement:

A - indicates that the statement is to be inserted in the cataloged procedure **after** the statement that has the same name.

B - indicates that the statement is to be inserted in the cataloged procedure **before** the statement with the same name.

To determine the symbolic names of statements in a procedure, print the procedure on SYSLST (see Procedure 19).

D - indicates that the statement that has the same name is to be deleted from the cataloged procedure.

ƀ - indicates that the statement is to **replace** the statement that has the same name. Instead of blank, any character other than A, B, or D may be used. Make a habit of using the character M; this enables you to identify a modifier statement when it is printed on SYSLOG or SYSLST. If you use a blank in column 80, modifier statements are more difficult to spot.

Examples

Assume, for example, that the cataloged procedure contains the statement

```
// ASSGN SYS005,DISK     NAME4
```

To delete this statement for your current job, enter the modifier statement

```
                       column 73   column 80
                           ↑           ↑
// ASSGN SYS005,DISK     NAME4         D
```

In addition to naming the statements and indicating the function to be performed, you must inform the job control program that it has to carry out a procedure modification. This is done by specifying the additional parameter OV (for overwrite) in the EXEC PROC= statement that calls the procedure:

```
// EXEC PROC=procname,OV
```

The end of the modifier statements for one procedure must be indicated by the statement:

```
// OVEND
```

Assume that the cataloged procedure PRCPAY contains, among others, the following control statements:

```
                              column 73
                              ↑
// ASSGN SYS020,X'181'        S4
// ASSGN SYS021,X'190'        S5
```

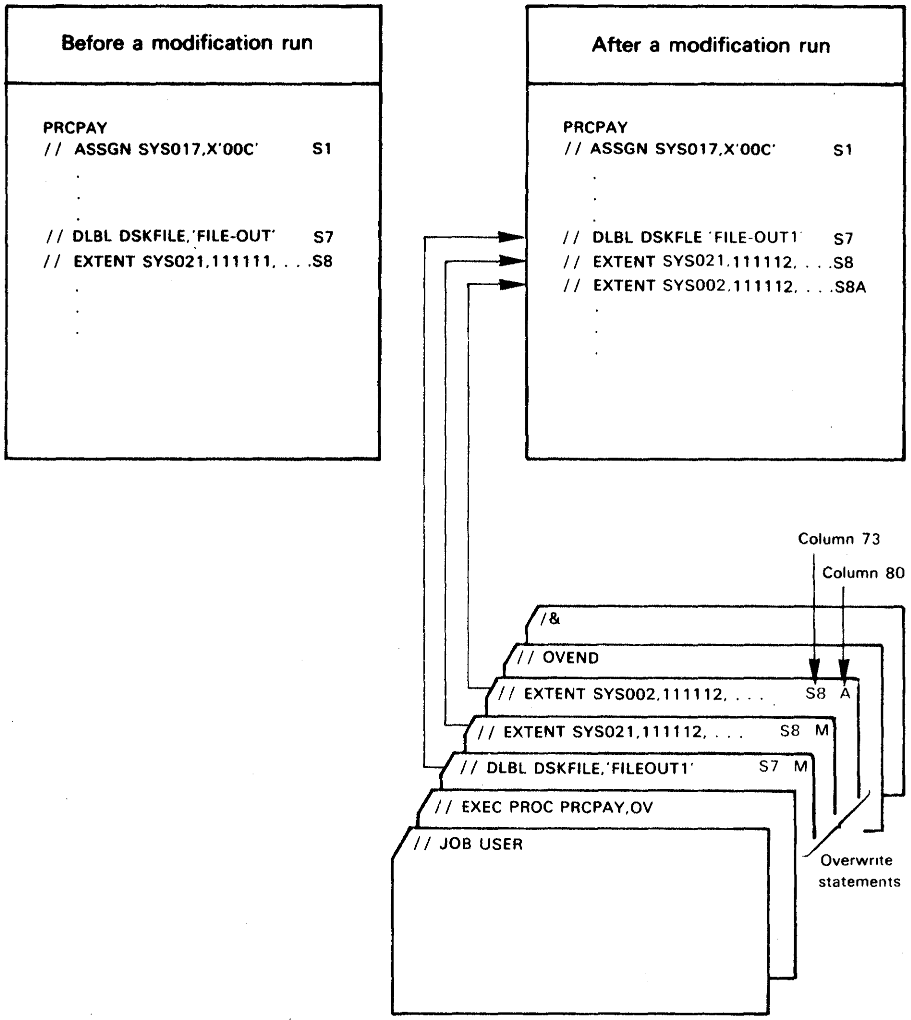
Assume further that tape unit 181 is out of order and you must use unit 182 instead. To invoke and execute procedure PRCPAY and at the same time modify the statement named S4 for this job, enter the following statements:

```
                              column 73     column 80
// JOB MODIFY                 ↑             ↑
// EXEC PROC=PRCPAY,OV        |             |
// ASSGN SYS020,X'182'        S4            M
// OVEND
/&
```

You may modify statements from the printer-keyboard. In that case, use the EXEC command instead of the EXEC statement.

As another example, assume that the program PAYROLL is to use the file FILE-OUT1 instead of FILE-OUT, and that FILE-OUT1 resides on two extents of a disk pack with volume serial number 111112. The input stream and processing would be as shown in Figure III-2.

Contents of the Procedure Library



Before a modification run

```
PRCPAY
// ASSGN SYS017,X'00C'          S1
     .
     .
     .
// DLBL DSKFILE,'FILE-OUT'      S7
// EXTENT SYS021,111111, . . .S8
     .
     .
     .
```

After a modification run

```
PRCPAY
// ASSGN SYS017,X'00C'          S1
     .
     .
     .
// DLBL DSKFLE 'FILE-OUT1'      S7
// EXTENT SYS021,111112, . . .S8
// EXTENT SYS002,111112, . . .S8A
     .
     .
     .
```

```
/&
// OVEND
// EXTENT SYS002,111112, . . .    S8 A
// EXTENT SYS021,111112, . . .    S8 M
// DLBL DSKFILE,'FILEOUT1'        S7 M
// EXEC PROC PRCPAY,OV
// JOB USER
```

Figure III-2. Example of overwrite statements

The operator action for a number of DOS/VS messages includes the listing of the volume table of contents (VTOC) of a certain disk pack. This listing helps your programmer to determine the error that caused the message.

An example of LVTOC output is given in *DOS/VS SADP, GC33-5380*, together with recommendations when to use this program.

To list the VTOC of a disk pack, you use the LVTOC utility program. You can do this in one of two ways:

1. Using SYSRDR
2. Using the printer-keyboard or display console (SYSLOG)

Using SYSRDR

The job control statements that are required to display the VTOC are:

```
// JOB name
// ASSGN SYS004,X'cuu'
// ASSGN SYS005,X'cuu'
// EXEC LVTOC
/&
```

SYS004 **must** be assigned to the physical address of the disk drive on which the disk pack in question is mounted. SYS005 **must** be assigned to the printer on which the VTOC is to be displayed.

Using the printer-keyboard or display console (SYSLOG)

To display the VTOC from the printer-keyboard or display console, enter the following statements:

```
// ASSGN SYS004,X'cuu'
// ASSGN SYS005,X'cuu'
// EXEC LVTOC
```

If you want to display the VTOC in response to an error message, cancel the program that caused the error before executing LVTOC.

An example of LVTOC output is given in *DOS/VS SADP*, GC33-5380, together with recommendations when to use this program.

Displaying the Label Information Cylinder(s)

20

The 'operator action' for a number of DOS/VS messages includes the listing of the label information cylinder(s) of the system residence pack (SYSRES). This listing helps your programmer to determine and correct the error that caused the message.

To display the label information cylinder(s), you use the LSERV program. You can do this in one of two ways:

1. Using SYSRDR
2. Using the printer-keyboard or display console (SYSLOG)

LSERV prints the contents of the label information cylinder(s) on the device assigned to SYSLST.

Using SYSRDR

The job control statements required to execute the LSERV program from SYSRDR are:

```
// JOB name
// EXEC LSERV
/*
/&
```

Using the printer-keyboard or display console (SYSLOG)

To execute the LSERV program from the printer-keyboard or display console, enter

```
// EXEC LSERV
```

If you want to display the label information cylinder(s) in response to an error message, cancel the program that caused the error before executing LSERV.

An example of LSERV output is given in *DOS/VS SADP*, GC33-5380, together with recommendations when to use this program.

Displaying and Punching the Libraries

The service functions of the librarian program allow you to display and/or punch the entire contents or selected parts of a library. For example, your programmer may request you to have a phase of the core image library punched into cards and then re-linkedited with one or more other phases to form a new program.

Recommendations on when to use these programs are given in *DOS/VS SADP, GC33-5380*.

Four librarian programs are available to print and punch the contents of the libraries:

CSERV for the core image library

PSERV for the procedure library

RSERV for the relocatable library

SSERV for the source statement library

The punch or print output of these programs is produced on the device assigned to SYSPCH or SYSLST.

The following tables contain the control statements required to punch libraries or their individual elements. When preparing the control statements, be sure to replace the words phase1, module1, and so on, by the actual phase or module names.

CSERV (core image library service program)

Function	**Control Statements Required**

Punch one or more phases	`// JOB jobname`
	`// EXEC CSERV`
	` PUNCH phase1[,phase2,...]`
	`/*`
	`/&`

Punch one or more programs	`// JOB jobname`
	`// EXEC CSERV`
	` PUNCH prog1.ALL[,prog2.ALL,...]`
	`/*`
	`/&`

Punch entire library	`// JOB jobname`
	`// EXEC CSERV`
	` PUNCH ALL`
	`/*`
	`/&`

Instead of PUNCH you may specify DSPLY or DSPCH. DSPLY causes the output to be printed; DSPCH causes the output to be printed **and** punched.

21

Examples

To punch the phases CORPHA and CORPHB from the core image library, enter

```
// JOB PUNCHPH
// EXEC CSERV
   PUNCH CORPHA,CORPHB
/*
/&
```

PUNCH $\underbrace{\text{prog.1}}_{\text{First four characters}}$.ALL

To punch an entire program - consisting of four phases named PHASE1 through PHASE4 - from the core image library, enter

```
// JOB PCHPROG
// EXEC CSERV
   PUNCH PHAS.ALL
/*
/&
```

You can use this form of the PUNCH statement only if all phases of the program to be punched have the first four characters in common; only these four characters are specified in the prog1.ALL operand.

PSERV (procedure library service program)

Instead of PUNCH you may specify either DSPLY or DSPCH. DSPLY causes the output to be printed; DSPCH causes the output to be printed **and** punched.

Function	**Control Statements Required**

Punch one or more procedures	`// JOB jobname`
	`// EXEC PSERV`
	`   PUNCH proc1[,proc2,...]`
	`/*`
	`/&`

Punch entire library	`// JOB jobname`
	`// EXEC PSERV`
	`   PUNCH ALL`
	`/*`
	`/&`

Example

To display the procedures PROCA and PROCB from the procedure library on the printer, enter

```
// JOB DISPROC
// EXEC PSERV
   DSPLY PROCA,PROCB
/*
/&
```

RSERV (Relocatable library service program)

Function	Control Statements Required

Punch one or more modules	`// JOB jobname`
	`// EXEC RSERV`
	`   PUNCH module1,module2,...`
	`/*`
	`/&`

Punch one or more programs	`// JOB jobname`
	`// EXEC RSERV`
	`   PUNCH prog1.ALL,prog2.ALL,...`
	`/*`
	`/&`

Punch entire library	`// JOB jobname`
	`// EXEC RSERV`
	`   PUNCH ALL`
	`/*`
	`/&`

Instead of PUNCH you may specify either DSPLY or DSPCH. DSPLY causes the output to be printed; DSPCH causes the output to be printed **and** punched.

Examples

To punch the modules MODA and MODC from the relocatable library, enter

```
// JOB PCHMOD
// EXEC RSERV
   PUNCH MODA,MODC
/*
/&
```

PUNCH $\underbrace{\text{prog.1}}$ ALL

First three characters

To punch an entire program - consisting of six modules named MOD1 through MOD6 - from the relocatable library, enter

```
// JOB PCHPROG
// EXEC RSERV
   PUNCH MOD.ALL
/*
/&
```

You can use this form of the PUNCH statement only if all phases of the program to be punched have the first three characters in common; only these three characters are specified in the prog1.ALL operand.

Displaying and Punching the Libraries (continued)

21

SSERV (Source Statement library service)

Function	Control Statements Required

Punch one or more books	`// JOB jobname`
	`// EXEC SSERV`
	`   PUNCH sublib.book1[,sublib.book2,...]`
	`/*`
	`/&`

Punch one or more sublibraries	`// JOB jobname`
	`// EXEC SSERV`
	`   PUNCH sublib1.ALL[,sublib2.ALL,...]`
	`/*`
	`/&`

Punch entire library	`// JOB jobname`
	`// EXEC SSERV`
	`   PUNCH ALL`
	`/*`
	`/&`

Instead of PUNCH you may specify DSPLY or DSPCH. DSPLY causes the output to be printed; DSPCH causes the output to be printed **and** punched.

Examples

To print and punch the books BKONE and BKTWO from the A sublibrary, enter

```
// JOB PPBOOK
// EXEC SSERV
   DSPCH A.BKONE,A.BKTWO
/*
/&
```

To print and punch the entire sublibrary A from the source statement library, enter

```
// JOB PPSUBL
// EXEC SSERV
   DSPCH A.ALL
/*
/&
```

Displaying the Library Directories

You can obtain reports on the contents of the libraries by displaying the library directories. For instance, if you receive a message that a requested procedure of the procedure library was not found, you can display the procedure directory to determine whether the procedure is missing from the library or whether you have merely misspelled the name of the procedure.

The library directories are displayed by the DSERV librarian program. The DSERV output is written on the device assigned to SYSLST.

An example of DSERV output is given in *DOS/VS SADP, GC33-5380*, together with recommendations when to use this program

Depending on the control statement used, the directories can be displayed in one of two ways:

1. Sorted alphamerically by element name (DSPLYS).
2. Unsorted, in the order in which the entries appear in the directory (DSPLY).

The following table shows the control statements required to obtain a sorted listing of each of the library directories.

Displaying the Library Directories (continued)

22

DSPLYS: sorted output
DSPLY: unsorted output

Directory	Control Statements Required

Core Image	// JOB jobname
	// EXEC DSERV
	   DSPLYS CD
	/*
	/&

Procedure	// JOB jobname
	// EXEC DSERV
	   DSPLYS PD
	/*
	/&

Relocatable	// JOB jobname
	// EXEC DSERV
	   DSPLYS RD
	/*
	/&

Source Statement	// JOB jobname
	// EXEC DSERV
	   DSPLYS SD
	/*
	/&

Transient	// JOB jobname
	// EXEC DSERV
	   DSPLYS TD
	/*
	/&

System Directory List	// JOB jobname
	// EXEC DSERV
	   DSPLYS SDL
	/*
	/&

System	// JOB jobname
	// EXEC DSERV /* /&

Condensing Libraries

When elements are deleted from a library the space occupied by these elements cannot immediately be used for cataloging new elements. To make this space available again you must condense the library by means of the MAINT librarian program and a CONDS control statement. The operands of the CONDS statement indicate the libraries to be condensed:

CL	-	core image library
RL	-	relocatable library
SL	-	source statement library
PL	-	procedure library

You may condense one or more libraries in one run. SYSIN, SYSLST, and SYSLOG must be assigned.

The following example shows the control statements needed to condense the core image and relocatable libraries; standard assignments for SYSIN, SYSLST, and SYSLOG are assumed:

```
// JOB NAME
// EXEC MAINT
   CONDS CL,RL
/*
/&
```

Restrictions

- The **system core image library and any relocatable or source statement libraries** can only be condensed from the background partition, and then only if there are no programs running in a foreground partition.

- A **private core image library** may be condensed in any partition, provided it is exclusively assigned to that partition and there are no other private core image libraries assigned to that disk.

- The **procedure library** can be condensed from any partition unless

  - it is being accessed by the job control program in another partition
  - it is being accessed by DSERV or PSERV in another partition
  - a procedure is being executed
  - a non-ending job, such as POWER, was started by a procedure and is still running.

Condense Limit

You can also specify that each time the available free space in a library drops below a specified minimum, referred to as the condense limit, you are to be notified by the following message:

3M78I library-name LIBRARY HAS REACHED CONDENSE LIMIT

Caution

Do not condense libraries unless you are told to.

Condense runs may take a long time and show all the symptoms of a looping program.

Use backup-restore instead of condense if tape drive is available.

NEVER CANCEL A CONDENSE RUN

If a condense run is interrupted by a hardware error or by operator intervention, the library being condensed is not usable again until it has been reconstructed.

23

This warning message is requested through the CONDL control statement indicating the library or libraries to be serviced and the condense limit(s).

Your programmer will tell you for which libraries the condense message service is to be established and the condense limits to be specified. At an appropriate time, the library or libraries can be condensed.

Example:

```
// JOB NAME
// EXEC MAINT
   CONDL CL=10
/*
/&
```

The CONDL statement in the above example indicates that the condense message for the core image library is to be issued whenever the number of available blocks in the library becomes less than ten.

Printing the Hard-Copy File

Each line that appears on the screen of the display console (DOC or 3277) is written to the hard-copy file, which resides on SYSREC. An IBM utility program (PRINTLOG) prints the hard-copy file from disk onto SYSLST.

SYSLST cannot be assigned to a DASD device. It must be a printer.

To print the hard copy file, proceed as follows:

1. Type `// EXEC PRINTLOG` and press ENTER.
   The following message appears on the screen:

   ```
   ENTER ONE OR MORE OF THE FOLLOWING OPTIONS:
   ALL OR NEW, AR OR BG OR F1 OR F2 OR F3 ... ETC.,
   A,D,I,E,U, JOBNAME=NAME,MM/DD/YY
   ```

Instead of MM/DD/YY, DD/MM/YY may appear, depending on the format chosen for your installation.

2. Select the desired options and enter them in the order in which they appear in the message on the screen. The options must be separated by commas; intervening blanks are not permitted.

   ALL specifies printing of all messages that are on the hard-copy file.

   NEW specifies printing of only those messages that have accumulated since PRINTLOG was last run with only the NEW or ALL option specified.

   If ALL or NEW is combined with one or more of the following options, its function applies only to the messages printed by these options. If neither ALL nor NEW is specified, ALL is assumed.

Specifying ALL causes the **entire** contents of the hard copy file to be printed, including any messages stored in overlay mode (see Screen Operation).

   AR specifies printing of messages issued by the attention routine.

   BG - Fn specifies printing of messages issued by a particular partition; only **one** partition identifier or AR may be specified. n can be from 1 up to the number of foreground partitions supported in your system.

   A specifies printing of action messages.

   D specifies printing of decision messages.

   I specifies printing of information messages.

   E specifies printing of eventual-action messages.

   U specifies printing of undefined messages.

   JOBNAME= specifies printing of messages pertaining to the job identified by NAME.

   MM/DD/YY specifies printing of messages issued on a particular day. Enter the date in the format month, day, year, or day, month, year as defined for your installation. Enter only the last two digits of the year. Leading zeros may be omitted.

24

The option ALL is default and may be omitted.

Examples

Print all action messages:

```
ALL,A
```

Print all action messages for F1:

```
ALL,F1,A
```

Print all new messages issued by the Attention routine:

```
NEW,AR
```

Print all messages issued on September 30,1973:

```
9/30/73
```

If you enter an option that does not exist, or if you do not adhere to the prescribed format, the following message appears on the screen:

THE FOLLOWING OPTIONS ARE INCORRECT: xxxxxxx -

PLEASE REENTER

The incorrect option(s) are displayed (xxxxxxx) and you need correct only the option(s) in error.

Executing Cataloged Programs

Programs that are used more than once can be cataloged permanently in the core image library after they have been assembled and link-edited. This saves assembling and link-editing the program for every run.

To execute a cataloged program you use an EXEC job control statement or command, specifying the name under which the program was cataloged.

The following example causes the execution of a program that was cataloged in the core-image library under the name PROGA; data cards are entered via SYSIPT.

Example:

```
// JOB jobname
    .
    .
 assignment and label
 statements, as required
    .
// EXEC PROGA
    .
    .
 input  data
    .
/*
/&
```

Debugging Procedures

26

Debugging aids
Alter/Display
Trace
Dump
EREP

When normal system operation is interrupted due to software or hardware failure, there are a number of procedures that either resolve the problem situation or provide your system programmer with information upon which he can act. These procedures are described in *DOS/VS SADP*, GC33-5380.

SD (system debugging) and PD (problem determination) aids, displaying and altering virtual storage, trace routines, dumps, and EREP (environmental recording, editing, and printing) are some of the means that are available to analyze error situations.

These aids are normally used in response to system messages on SYSLOG. These messages are contained in *DOS/VS Messages*, GC33-5379, and tell you which debugging aids you should use in a particular situation.

You should then consult *DOS/VS SADP*, GC33-5380, which describes the debugging procedures in detail.

Testing Teleprocessing Terminals

DOS/VS provides a set of online tests (OLTs) to test teleprocessing terminals. These tests and the Teleprocessing Online Test Executive Program (TOLTEP) facilities for VTAM make up an online test system.

TOLTEP is automatically included in the system when VTAM is specified during system generation. The test program may be used for:

- checking VTAM terminals for recurring hardware errors;
- diagnosing I/O errors caused by hardware failure;
- verifying terminal repairs;
- verifying engineering changes.

The IBM customer representative is the primary user of TOLTEP, but the facility is available to any user of the system to aid him in isolating device failures and to enable him to periodically check terminals used by VTAM.

Details of the operating procedures, methods of invocation, options, tests, and the messages and responses required to run TOLTEP are given in *TOLTEP for VTAM*, GC28-0663.

Shutting down the System

28

There are a number of things you should do **before** you switch off the power at the CPU:

- If your installation uses the RMSR function, issue the ROD command. This command causes statistical information on the system, accumulated since the previous IPL procedure, to be written to the system recorder file (SYSREC).
- Unload and remove all tape reels to prevent the tape heads from damaging the tape when the power is switched off.
- Switch off all disk drives and remove the disk packs.

Depending on the complexity of your installation, you may have to do other things in addition to those described above. If you operate a system with a display console, for example, you may want to make a printout of the hard-copy file for easy reference. Your installation manager will give you detailed instructions on how to shut down the system.

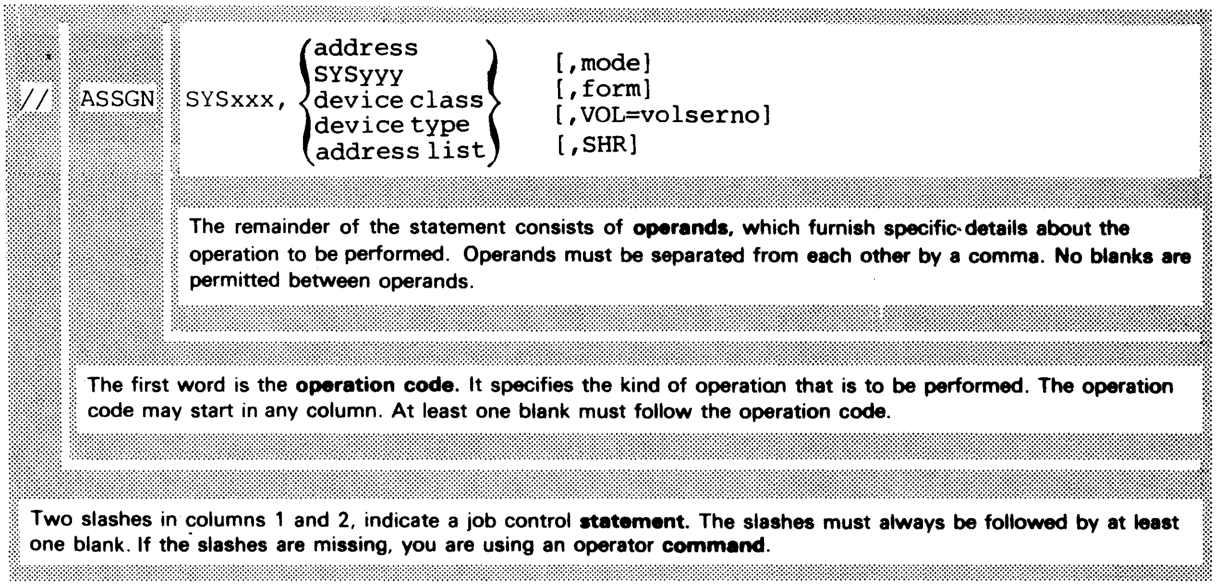
Reference Information

Section 4

In your job you will be concerned with entering statements and commands, all of which are contained in this reference section. If you are not yet an experienced operator, you will have to refer to this section quite frequently.

Syntax Rules

Statements and commands are written according to a few simple rules, which are best discussed in an example. The ASSGN statement/command is used to illustrate these rules (see Figure IV-1).



```
                 (address      )
                 |SYSyyy       |   [,mode]
// ASSGN SYSxxx, <device class >   [,form]
                 |device type  |   [,VOL=volserno]
                 (address list )   [,SHR]
```

The remainder of the statement consists of **operands**, which furnish specific details about the operation to be performed. Operands must be separated from each other by a comma. No blanks are permitted between operands.

The first word is the **operation code**. It specifies the kind of operation that is to be performed. The operation code may start in any column. At least one blank must follow the operation code.

Two slashes in columns 1 and 2, indicate a job control **statement**. The slashes must always be followed by at least one blank. If the slashes are missing, you are using an operator **command**.

Figure IV-1. How to read and write statements and commands

Braces { }

If several operands are stacked between braces, for example,

$$\left\{\begin{array}{l}\text{address}\\\text{SYSyyy}\\\text{device class}\\\text{device type}\\\text{address list}\end{array}\right\}$$

You **must** select one (and only one) of these operands.

Square brackets [ ]

If several operands are stacked between square brackets, you **may** use one (and only one), but you do not have to.

A single operand, for example, [,SHR], may either be used or omitted.

Note that braces and square brackets do not appear in the actual statement or command.

Commas, apostrophes, digits, uppercase

Commas, apostrophes, digits, and uppercase letters must be specified exactly as shown in the following examples:

```
// ASSGN SYSLST,X'00E'
// ASSGN SYS001,DISK,VOL=222222,SHR
```

Lowercase

Lowercase letters stand for information that you or the programmer must supply; for instance, 'address' may be specified as X'191'.

Hexadecimal values

Some operand values must be supplied in hexadecimal form (this is always explicitly stated in the description of a statement or command). Hexadecimal values can consist of the numeric characters 0 - 9 and of the alphabetic characters A - F.

Syntax Rules (continued)

Other operand values must be supplied in decimal form, which is indicated by the letter n in the general statement format. Decimal values may consist of the numeric characters 0 - 9. Take, for example, the NEWVOL command:

```
NEWVOL [BG | Fn]
```

Decimal values

For the letter n, you have to specify a decimal value which, in this specific case, identifies one of the foreground partitions in your system, for example F2.

Default values

Operands or operand values that are underlined, for example, BG in the preceding example, are default values; that is, they are assumed if no operand is supplied. In the NEWVOL command, for example, you can either specify the background or one of the foreground partitions. If you do not specify any partition (no operand), the background partition is assumed.

Operator commands and job control statements are described separately in alphabetical order. The operator commands are described first. Figure IV-2 shows a list of all DOS/VS operator commands. A detailed description of each operator command follows this list. Figure IV-5, which shows all DOS/VS job control statements, is followed by a description of each job control statement.

Operator Commands

There are five types of operator commands:

- IPL commands
- job control commands
- attention commands
- POWER/VS commands
- VTAM operator commands

POWER/VS and VTAM operator commands are not discussed here. POWER/VS commands are described in *DOS/VS POWER/VS Installation Guide and Reference*, GC33-6408, and *DOS/VS POWER/VS Workstation User's Guide*, GC33-6049. VTAM commands are described in *VTAM Network Operating Procedures*, GC27-6997. The main difference between the other three types of commands lies in (1) the time at which they can be entered and (2) the way in which they are entered.

IPL commands

IPL commands are accepted only during the IPL procedure. They provide information the system needs before processing can start. IPL commands can be entered from the console printer-keyboard, the display console, a card reader, or from the 3540 Diskette I/O Unit.

When entering IPL commands from the console printer-keyboard, press END after every command.

When entering IPL commands from the display console press ENTER after every command.

When all IPL commands have been entered or read, the system issues the message

```
0I20I IPL COMPLETE FOR DOS/VS REL xx.x ECLEVEL=nn
             [SUPVR USERID IS: id]
```

Then job control tries to read job control statements/commands either from the console (if the IPL commands were entered via console/diskette) or from the device to which SYSRDR was assigned (if the IPL commands were entered via this device and if SYSRDR is the standard system generation assignment for the device).

Job control commands

Job control commands are accepted immediately after IPL and between jobs and job steps. They can be entered either from the console printer-keyboard, from the display console, or from the device to which SYSRDR is assigned.

When entering job control commands from the console printer-keyboard or from the display console, press END or ENTER, respectively, after typing in a command.

Job control commands to be read from SYSRDR are simply placed in the job stream, that is, they are treated in the same way as job control statements. (In certain cases, however, their effect is different.)

Attention commands

Attention commands can be entered at any time after IPL during system operation. They can be entered only from the console printer-keyboard or from the display console. Before entering an attention command from the printer-keyboard, press REQUEST. The system responds by issuing the attention routine identifier AR.

Attention commands from the display console are entered as follows:

1. Type in the command
2. Press ENTER

If the screen displays

ENTER RESPONSE

press REQUEST and wait for the message

ENTER COMMAND

Then type in the command and press ENTER.

Mixed-type commands

Some commands belong to more than one type. The ALLOC command, for example, can be used either as a job control command or as an attention command. It may be that the effect of a command differs depending on the type used.

The size of an **active** real partition, for example, can be reduced only by the ALLOCR job control command. When a real partition is **inactive**, you may use either the ALLOCR job control command or the ALLOCR attention command.

It may also be that certain operands are permitted or not permitted, depending on the type used. Any differences in effect and type are explicitly stated in the following descriptions.

Figure IV-2 lists all operator commands and their functions, except POWER/VS and VTAM operator commands.

Command	Meaning	Type of Command		
		IPL	Job Control	Attention
ADD	Add a device to the PUB table	X		
ALLOC*	Allocate storage to virtual partitions		X	X
ALLOCR*	Allocate storage to real partitions		X	X
ALTER	Alter 1 to 16 bytes in virtual storage			X
ASSGN	Assign a logical unit		X	
BATCH*	Initiate or resume processing in a partition			X
CANCEL	Cancel execution of current job		X	X
CAT	Define the VSAM catalog	X		
CLOSE	Close a logical unit		X	
DEL	Delete a device from the PUB table	X		
DPD	Define the page data set	X		
DSPLY	Display 16 bytes of virtual storage			X
DUMP	Print part or all of virtual storage			X
DVCDN	Make a device 'down'		X	
DVCUP	Make a device 'up'		X	
ENDSD	Terminate execution of SDAID program			X
EXEC	Execute a program or procedure		X	
HOLD	Hold unit assignments after subsequent UNBATCH until end of next job		X	
IGNORE	Ignore command just entered			X
LFCB	Load forms control buffer			X
LISTIO	List current I/O assignments		X	
LOG	Log (print) job control statements and commands		X	X
LUCB	Load universal character-set-buffer			X
MAP*	List virtual storage allocations		X	X
MODE	Set RMS recording mode or make status inquiry			X

* only if your system has more than one partition

Figure IV-2. Alphabetical list of operator commands (Part 1 of 2)

Command	Meaning	Type of Command		
		IPL	Job Control	Attention
MSG*	Communicate with a foreground program			X
MTC	Perform magnetic tape control operations		X	
NEWVOL	Resume processing, required volume is mounted.			X
NOLOG	Suppress logging of job control statements and commands		X	X
OVEND	Indicate end of overwrite statements for cataloged procedure		X	
PAUSE	Interrupt processing at end of job or job step		X	X
PRTY*	Modify or display partition priorities			X
RESET	Reset temporary I/O device assignments		X	
ROD	Record on demand and end-of-day message		X	
SET	Set system values	X**	X	
START*	Initiate or resume processing in a partition			X
STOP*	Stop processing in a partition		X	
TPBAL*	Display or modify TP balancing			X
UCS	Load Universal Character Set buffer		X	X
UNBATCH*	Deactivate a partition		X	

* only if your system has more than one partition
** date, clock and zone only

Figure IV-2. Alphabetical list of operator commands (Part 2 of 2)

ADD
add a device to the PUB table

ADD is an optional IPL command that is used to add a device (not assigned during system generation) to the PUB table. If specified, the ADD command(s) must precede the SET command. It may precede or follow DEL commands.

Operation	Operands	Type
ADD	X'cuu'[k],devicetype [,X'ss' ,X'ssss' ,X'ssssss']	IPL

See also Procedures 2 - 7

X'cuu' indicates channel and unit numbers (in hexadecimal notation).

k may be either of the following:

(S) indicates that the device can be switched (that is, attached to two adjacent channels). The designated channel (X'cuu') is the lower of the two channels.

(n) indicates the priority of a device that cannot be switched. n is a decimal number from 0 to 255. The highest priority is 0. If k is not given, a priority of 255 is assumed. In a multiprogramming environment, all devices on a channel automatically have equal priority.

devicetype see Figure IV-3.

X'ss'
X'ssss'
X'ssssss' specific device information usually supplied to you by the programmer. For details you may, however, refer to *DOS/VS System Control Statements*, GC33-5376.

Device Code	Actual IBM Device	Device Type
2501	2501 Card Reader	Card Readers
2540R	2540 Card Reader	
3505	3505 Card Reader	
3504	3504 Card Reader	
2540P	2540 Card Punch	Card Punches
2520B2	2520B2 Card Punch	
1442N2	1442N2 Card Punch	
2520B3	2520B3 Card Punch	
3525P	3525 Card Punch	
1442N1	1442N1 Card Read Punch	Card Read Punches
2520B1	2520B1 Card Read Punch	
2596	2596 Card Read Punch	
3525RP	3525 Card Punch (with optional read feature)	
2560	2560 Multi-Function Card Machine	
5425	5425 Multi-Function Card Unit	
PRT1	3211 or 3203-4 Printer	Printers
1403	1403 Printer	
1403U	1403 Printer with UCS feature	
1443	1443 Printer	
2245	2245 Kanji Printer	
3211	same as PRT1	
3525P	3525 Card Punch (with optional print feature)	
3203	3203 Printer (Models 1 and 2)	
5203	5203 Printer	
5203U	5203 Printer with UCS feature	
1050A	3210, 3215 Console Printer-Keyboards	Printer-Keyboards
125D	Model 115 or 125 integrated Display Operator Console	Display Consoles **Note:** The 3277 Display Console must not be attached to a selector channel.
125DP	Model 115 or 125 integrated Display Operator Console with attached 5213 console printer	
3277	3277 Display Console	
2400T9	9-track Magnetic Tape Units (2400-series)	Magnetic Tape Devices
2400T7	7-track Magnetic Tape Units (2400-series)	
3410T9	9-track Magnetic Tape Units (3400-series)	
3410T7	7-track Magnetic Tape Units (3400-series)	
3420T9	9-track Magnetic Tape Units (3400-series)	
3420T7	7-track Magnetic Tape Units (3400-series)	
2495TC	2495 Tape Cartridge Reader	Tape Cartridge Reader
2311	2311 Disk Drive	DASD
2314	2314 Direct Access Storage Facility	
2314	2319 Disk Storage Facility	
2321	2321 Data Cell Drive	
3330	3330 and 3333 Disk Storage	
3330	3350 Direct Access Storage in 3330 compatibility mode	
3330B	3330-11 Disk Storage	
3330B	3350 in 3330-11 compatibility mode	
3340	3340 Disk Storage	
3340	3344 Direct Access Storage	
3340R	3340 Disk Storage with Rotational Position Sensing (RPS) feature	
3350	3350 Direct Access Storage	
3540	3540 Diskette I/O Unit	Diskette

Figure IV-3. Device codes used in the ADD command (Part 1 of 2)

Device Code	Actual IBM Device	Device Type
1419 1419 1419 1419P 1419S	1255 Magnetic Character Reader 1259 Magnetic Character Reader 1419 Magnetic Character Reader 1419 Dual Address Adapter Primary Control Unit 1419 Dual Address Adapter Secondary Control Unit	MICR - Magnetic Ink Character Recognition Devices
1017 . 2671	1017 Paper Tape Reader with 2826 Control Unit Model 1 2671 Paper Tape Reader	Paper Tape Readers
1018	1018 Paper Tape Punch with 2826 Control Unit Model 1	Paper Tape Punch
1419 1419P 1419S 1287 1288 3881 . 3886	1270 Optical Reader Sorter 1275 Optical Reader Sorter Primary Control Unit 1275 Optical Reader Sorter Secondary Control Unit 1287 Optical Reader 1288 Optical Page Reader 3881 Optical Mark Reader (cannot be used as a SYSIN device) 3886 Optical Character Reader	Optical Readers
2260 3277 3277B	2260 Display Station or 1053 Printer 3277 Display Station or 3284/86 Printer 3277 Display Station or 3284/86 Printer running in burst mode on the MPX channel	Display Station or Printer
2701 2702 2703 2703 . 2703 3704 . 3705-1 . 3705-2 . 3791L	2701 Data Adapter Unit 2702 Transmission Control Unit 2703 Transmission Control Unit Integrated Communications Adapter (Models 125 and 135) 3705 Communications Controller in Emulation Mode 3704 Communications Controller in Network Control Mode 3705 Communications Controller in Network Control Mode with Type 1 Adapter 3705 Communications Controller in Network Mode with Type 2 Adapter 3791 Local Communications Controller	Teleprocessing Lines
2955	2955 Data Adapter Unit	Data Link for RETAIN
7770	7770 Audio Response Unit	Audio Response Unit
UNSP UNSPB	Unsupported Device Unsupported Device	No burst mode on multiplexor channel Burst mode on multiplexor channel

Figure IV-3. Device codes used in the ADD command (Part 2 of 2)

ALLOC
allocate virtual storage

The ALLOC command allocates storage to the virtual foreground partitions (or modifies the amount of storage allocated to the virtual foreground partitions). All of the virtual address area not allocated to virtual foreground partitions is automatically part of the virtual background partition.

See also Procedure 14

If your system uses a shared virtual area (SVA), only the address space in the virtual address area not allocated to the shared virtual area (SVA), can be allocated to foreground partitions with this command.

Operation	Operands	Type
ALLOC	Fn=mK[,Fn=mK]...	AR, JC

n indicates the partition to which storage is to be allocated, m specifies the amount of storage. The minimum amount of storage allocated to an active (batched) virtual foreground partition is 64K. The value m should be an even number; any odd specification (greater than 64K) is rounded up to the nearest even integer.

A non-active foreground partition may be allocated 0K bytes of virtual storage, for example, if you want to add its size to another partition. The size of the virtual background partition cannot be reduced below 64K.

The ALLOC command must have at least one operand; it may have as many operands as there are foreground partitions in your system. The operands can be specified in any order. The first operand must not be preceded by a comma.

Caution: Modifying the size of active partitions may lead to errors, either in the same or in other partitions. You should therefore obtain precise instructions before issuing the ALLOC command. It is further recommended to UNBATCH all affected partitions before issuing the ALLOC command. Refer to *DOS/VS System Control Statements*, GC33-5376, if you wish to modify active virtual partitions.

ALLOCR
allocate real storage

See also Procedure 14

The ALLOCR command is used to allocate storage to real partitions, or to modify the amount of storage allocated to real partitions.

If your system uses a shared virtual area (SVA), only the address space in the virtual address area not allocated to the shared virtual area (SVA), can be allocated to foreground partitions with this command.

Operation	Operands	Type
ALLOCR	BGR=mK[,FnR=mK] ...	AR, JC

n indicates the partition to which storage is to be allocated, m specifies the amount of storage. m should be an even number; any odd specification is rounded up to the nearest even number. m may also be zero.

The ALLOCR command must have at least one operand; it may have as many operands as there are partitions in your system. The operands can be specified in any order. The first operand must not be preceded by a comma.

Caution: Modifying the size of active partitions may lead to errors, either in the same or in other partitions. You should therefore obtain precise instructions from your system programmer before issuing the ALLOCR command. Refer to *DOS/VS System Control Statements*, GC33-5376, if you wish to modify an active real partition.

The size of an **active** real partition can be reduced only by the ALLOCR job control command issued in that real partition. When a real partition is **inactive** (unbatched), you may use either the ALLOCR job control command or the ALLOCR attention command. The effect then is the same.

ALTER
alter virtual storage

The ALTER command allows you to alter, through the device assigned to SYSLOG, up to 16 bytes of virtual storage, starting at the specified hexadecimal address.

Two hexadecimal characters must be entered to change each byte of virtual storage; these characters represent the hexadecimal equivalent of the information to be stored.

Operation	Operands	Type
ALTER	address	AR

address six-digit hexadecimal address at which storage alteration is to start. Use leading zeros, if necessary.

The use of the ALTER command is described in detail in *DOS/VS SADP*, GC33-5380.

ASSGN
assign a logical unit name

See also Procedure 13

The ASSGN command assigns a logical unit name to a physical I/O device.

Permanent Device Assignments. A permanent assignment is set up between jobs or job steps any time after IPL by the ASSGN job control command (no //) or the ASSGN job control statement with the PERM operand. It is valid until the next IPL procedure unless superseded by another ASSGN job control command. A permanent assignment can be changed for the duration of a job or job step by a // ASSGN statement or by an ASSGN command with the TEMP option.

Temporary Device Assignments. A temporary assignment is established either by a // ASSGN statement or by an ASSGN command with the TEMP option. It is valid for a single job, unless superseded by another temporary or permanent assignment. Temporary assignments are reset to standard or permanent by

- a /& or JOB statement, whichever occurs first, or by
- a RESET job control statement or command.

Operation	Operands			Type
ASSGN	SYSxxx,	X'cuu' (address list) UA IGN SYSyyy device-class device-type	,TEMP ,PERM ,VOL=volserno ,SHR ,X'ss' ,ALT ,H1 ,H2	JC

The entries in the operand field represent the following:

SYSxxx the symbolic unit name, which may be either

- one of the system logical units:

  SYSCLB
  SYSIN
  SYSIPT
  SYSLNK
  SYSLOG
  SYSLST
  SYSOUT
  SYSPCH
  SYSRDR
  SYSREC
  SYSRLB
  SYSSLB

- or a programmer logical unit SYSnnn, where nnn is a decimal value between 000 and a maximum number, which depends on the configuration of your system.

Restrictions: The type of device assignment is restricted under certain conditions:

1. If one of the system logical units SYSRDR, SYSIPT, SYSLST, or SYSPCH is assigned to a disk device or diskette, the assignment must be permanent and follow the DLBL and EXTENT statements. SYSCLB must always be (permanently) assigned to disk.

2. If SYSRDR and SYSIPT are to be assigned to the same disk device or diskette SYSIN must instead be assigned and this assignment must be permanent.

3. SYSOUT, if used, must always be a permanent assignment and is valid for a tape unit only.

4. SYSIN and SYSOUT cannot be specified in the ASSGN macro during supervisor generation, that is, they cannot be standard assignments.

X'cuu' is a combined channel and unit number, where

c	=	0 for multiplexor channel or 1 - 6 for the selector channels
uu	=	00 - FE

If you assign a dummy device as SYSIPT, SYSIN, SYSLST, or SYSPCH to a POWER/VS supported partition, X'cuu' must be specified.

address list a string of hexadecimal device addresses between parentheses, separated by commas (X'cuu', X'cuu',...,X'cuu'). The maximum number of addresses that may be specified is seven. The system will select one of these devices.

UA (UnAssigned) indicates that the logical unit is to be unassigned. Any operation attempted on anunassigned device results in job cancelation.

IGN (IGNore) unassigns the specified logical unit and causes any subsequent references to the unit to be ignored. It is invalid for SYSRDR, SYSIPT, SYSIN, and SYSCLB.

SYSyyy can be any system or programmer logical unit (see *SYSxxx*, above. This can be used for example to assign a programmer logical unit to the same physical device as a system logical unit. For example, ASSGN SYS005, SYSIN assigns SYS005 to the physical device which is used by SYSIN.

device class may be any one of the following: READER (card reader), PRINTER, PUNCH (card punch), TAPE (magnetic tape unit), DISK (DASD device), or DISKETTE. If you use one of these words, the system will search for an available device in that category. A successful scan is followed by message 1T20I, informing you of the assigned device address. Do not specify device class for dummy devices. If your installation includes DASD drives with and without the Fixed Head Feature, such as the 3348 Model 70F Data Module, do not use device-class. Instead, use X'cuu' or (address-list) to specify the drives with the feature to avoid job cancellation. For more information, see *IBM 3340 Fixed Head Feature Users Guide*, GA26-1632.

device type

may be any supported device type (for example 2400T7) in one of the classes mentioned above. See the list of devices in the description of the ADD command in this manual (Figure IV-3). The system will search for an available device of the desired type. When found, message 1T20I is issued, telling you the device address to which the symbolic unit has been assigned. If your installation includes DASD drives with and without the Fixed Head Feature, such as the 3348 Model 70F Data Module, do not use device-type. Instead, use X'cuu' or (address-list) to specify the drives with the feature to avoid job cancellation. For more information, see *IBM 3340 Fixed Head Feature Users Guide*, GA26-1632.

TEMP
PERM

Indicates whether the assignment should be permanent (PERM) or temporary (TEMP). It is thus possible to override the // specification or omission.

volserno

is the volume serial number, a one to six digit identification number of a tape reel, a diskette or a disk pack. You might, for example, mount disk pack 222222 on any disk drive and then specify:

```
ASSGN SYS035,DISK,VOL=222222
```

The system will issue message 1T20I, telling you the device address. If at the time the command is issued you have not already mounted the disk pack with the corresponding volume serial number and readied the device, message 1T50A will prompt you to do so.

Specifying a volume serial number in quotes will cause right hand padding. For example, VOL='NR 3' will result in the identification NRƀ3ƀƀ.

SHR

(share) must be included if you use a disk which is already owned by a partition.

X'ss'

is the device specification used to specify mode settings for 7-track and 9-track tapes (see Figure IV-4). If X'ss' is not specified, the following values are assumed depending on the value specified during system generation, or during IPL in the ADD command:

X'C0' for 9-track tapes (except the IBM 3420)

X'D0' for the 9-track IBM 3420
X'90' for 7-track tapes

Density (bpi)	Parity	Convert Feature	Trans- late	ss
200	odd	on	off	10
200	odd	off	off	30
200	odd	off	on	38
200	even	off	off	20
200	even	off	on	28
556	odd	on	off	50
556	odd	off	off	70
556	odd	off	on	78
556	even	off	off	60
556	even	off	on	68
800	odd	on	off	90
800	odd	off	off	B0
800	odd	off	on	B8
800	even	off	off	A0
800	even	off	on	A8
800	single-density 9-track tapes			C8
1600	single-density 9-track tapes			C0
1600	dual-density 9-track tapes			C0
800	dual-density 9-track tapes			C8
6250	single/dual density, 9-track			D0
1600	3420 Models 4, 6, and 8			C0

Figure IV-4. Device specification for tapes

ALT indicates an alternate magnetic tape unit that is used when the capacity of the original assignment is reached. Using multivolume tape files without ALT mode can cause performance degradation because the first tape has to be rewound before the next tape can be mounted. The characteristics of the alternate unit must be the same as those of the original unit. Multiple alternates may be assigned to a symbolic unit. The original and the alternate assignment must be either both permanent or both temporary. If the original unit is reassigned, the alternate must also be reassigned. The ALT operand is invalid for SYSRDR, SYSIPT, SYSIN, SYSLNK, SYSCLB, and SYSLOG.

H1 indicates that hopper 1 will be used for input on the 2560 or 5425. If neither H1 nor H2 is specified, H1 is assumed.

H2 indicates that hopper 2 will be used for input on the 2560 or 5425. H2 must not be specified for programmer logical units (SYSnnn).

If both hoppers (H1 and H2) are used, they must be assigned to the same partition.

Hopper specifications are significant only for device-independent files, associated with one of the following system logical units: SYSIPT, SYSRDR, SYSIN, or SYSPCH. In all other cases they are ignored.

If the 2560 or 5425 is assigned to a partition supported by POWER/VS, hopper 1 must be used for input and hopper 2 for output.
When no separator cards are used also hopper 1 can be used for output.

BATCH
initiate or continue processing

See also Procedure 16

The BATCH command serves to start or continue processing in one of the foreground partitions or to continue processing in the background partition. The function of the BATCH command is exactly the same as that of the START command.

Operation	Operands	Type
BATCH	[partition]	AR

If the specified partition has a minimum size of 64K bytes, the job control program reads the next command or statement from SYSLOG.

If the specified partition was temporarily halted by a STOP command, the job control program also reads the next statement or command from SYSLOG. This may be any job control statement or command.

If the specified partition is already active and has not been temporarily halted when the BATCH command is issued, the system responds with the message

AREA NOT AVAILABLE

If no operand is specified for the BATCH command, the background partition is assumed.

CANCEL
cancel a job

The CANCEL command serves to cancel a job in a partition. The job will be canceled as soon as all outstanding interruptions have been handled.

Operation	Operands	Type
CANCEL	[partition]	AR
CANCEL	blank	JC

The CANCEL command requires no operand when used as a job control command; it cancels the job in the corresponding partition.

As an attention routine command, CANCEL can only be used without an operand in a one-partition system. In a system with more than one partition, the partition to be cancelled must be specified, for example BG or F2.

The CANCEL command cannot be used to cancel the POWER/VS partition in which POWER/VS is running nor for a partition waiting for POWER/VS input.

CAT
assign logical unit for the VSAM master catalog

See also Procedures 2 and 3

Catalogs are used to describe all Virtual Storage Access Method (VSAM) files. A master catalog points to one or more user catalogs. The master catalog must always be available to the system when VSAM is used.

The CAT command serves to assign the system logical unit SYSCAT, on which the VSAM master catalog resides, to a physical I/O device and overrides any system generation SYSCAT assignment until the next IPL. If specified, the CAT command must follow the SET command and precede the DPD command.

Operation	Operands	Type
CAT	UNIT=X'cuu'	IPL

X'cuu' indicates the channel and unit number (in hexadecimal) of the device that contains the VSAM master catalog volume. SYSCAT may only be assigned to a 2314, 2319, 3330, 3333, 3340, or 3350.

CLOSE
close a logical unit

See also Procedure 17

The CLOSE command closes either a system or programmer output logical unit assigned to a magnetic tape, or a system logical unit assigned to disk or diskette.

Operation	Operands	Type
CLOSE	SYSxxx ,X'cuu'[,X'ss] ,UA ,IGN ,ALT ,SYSyyy	JC

SYSxxx may be one of the following:

Disk Drives Diskette Units	Magnetic Tapes
SYSRDR	SYSPCH
SYSIPT	SYSLST
SYSIN	SYSOUT
SYSPCH	SYSnnn
SYSLST	

If a system logical unit is specified, one of the optional operands **must** also be specified.

If a programmer logical unit is specified, no additional operand is required. When no additional operand is specified, the assignments of the closed unit remain unchanged.

X'cuu' reassigns the closed file associated with the logical unit to another device. X'cuu' is the address (channel and unit number) in hexadecimal, where:

c	=	0	for multiplexor channel, or
c	=	1-6	for selector channels
uu	=	00-FE	00-254 in hexadecimal

Assume, for example, that SYSPCH is assigned to X'180'. Specifying

```
CLOSE SYSPCH,X'181'
```

reassigns SYSPCH to the device with the number 181.

UA specifies that the logical unit is to be (permanently) unassigned after the file has been closed.

IGN specifies that the logical unit is to be (permanently) unassigned after the associated file has been closed. Any subsequent references to the unit will be ignored until a new ASSGN is given for the unit or IPL is performed. This operand is invalid for SYSRDR, SYSIPT, and SYSIN.

ALT

specifies that a file associated with the logical unit is to be closed and that a file associated with an alternate unit is to be opened and used. This operand is valid only for SYSPCH, SYSLST, and SYSOUT, currently assigned to a magnetic tape unit.

SYSyyy

specifies that after SYSxxx is closed, it will be assigned to the physical device to which SYSyyy is currently assigned (and to which it remains assigned). If SYSyyy is a system logical unit, it will be opened if it is either a disk, 3540 diskette, or a magnetic tape at load point.

X'ss'

gives device specifications to indicate the mode settings for 7-track and 9-track tapes (see Figure IV-4). If X'ss' is not specified, the mode settings remain unchanged. The LISTIO command may be used to determine the current mode settings for all magnetic tape units.

DEL
delete a device from the PUB table

See also Procedures 2 - 7

DEL is an optional IPL command that deletes a device from the PUB table. A device thus deleted can no longer be used by the system.

If used, the DEL command(s) must precede the SET command. It may precede or follow ADD commands.

Operation	Operands	Type
DEL	X'cuu'	IPL

X'cuu' represents the channel and unit number (in hexadecimal) of the physical device to be deleted from the PUB table.

DPD
define the page data set

See also Procedures 2 and 3

The DPD command defines the page data set. The operation code of this command must always be specified during the IPL procedure.

Operation	Operands	Type
DPD	[TYPE=$\left\{\begin{matrix}\underline{N}\\F\end{matrix}\right\}$][,UNIT=X'cuu',CYL=xxx] [,VOLID=xxxxxx]	IPL

The operands of the DPD command may be entered in any order.

All operands are optional; they need be specified only if the required information was not supplied during system generation or if changes in the definition are desired. The DPD command must be the last command entered during the IPL procedure.

TYPE=N indicates that the page data set need not be formatted (arranging data) because it has already been formatted and the extent limits have not been changed.

If TYPE=N is specified but the page data set does not exist or the extent limits have been changed, TYPE=N is ignored and the page data set is formatted during IPL. In this case, the UNIT and CYL operands must either have been supplied during system generation, or they must be specified in the DPD command.

TYPE=F indicates that the page data set is to be formatted during IPL. Formatting during IPL is required if the page data set is to be extended or if it is to be reallocated.

UNIT=X'cuu' specifies the channel and unit number (in hexadecimal) of the device that is to contain the page data set. If UNIT is specified, CYL must also be specified.

CYL=xxx specifies the number of the cylinder where the page data set is to begin. The size of the page data set extent is calculated by the system. If CYL is specified, UNIT must also be specified. If the space between the specified cylinder and the end of the disk pack is too small to contain the page data set, the system issues a message to this effect.

VOLID=xxxxxx identifies the volume serial number of the disk pack that contains the page data set. If this operand is omitted both during system generation and in the DPD command, the volume serial number is not checked. In that case, however, files may be destroyed if the wrong volume is mounted.

DSPLY
display virtual storage

The DSPLY command allows you to display, on the device assigned to SYSLOG, 16 bytes of virtual storage, starting at the specified address.

Two hexadecimal characters appear on SYSLOG for each byte of information; these characters represent the hexadecimal equivalent of the current information in virtual storage.

Operation	Operands	Type
DSPLY	address	AR

address six-digit hexadecimal address, with leading zeros if necessary, at which the storage display is to start

The use of the DSPLY command is discussed in detail in *DOS/VS SADP*, GC33-5380.

DUMP
print a dump

The DUMP command allows you to print all or part of the contents of the real and/or virtual address area on the printer to which SYSLST is assigned. SYSLST can be assigned to any partition when the dump is printed, but the printer must not be in use, otherwise partition and dump output will be mixed.

Detailed information on dumps is contained in *DOS/VS SADP, GC33-5380*

If SYSLST is assigned to a 3211 printer and the printer's indexing feature is being used, a certain number of characters may get lost on every dump line. To avoid losing characters, you should load a new FCB (forms control buffer) image to turn off indexing before you issue the DUMP command. Loading the new FCB image can be done either by issuing an LFCB command or by having the SYSBUFLD program executed.

Note that the transient area will be tied up while the dump takes place since output is not spooled.

Operation	Operands		Type
DUMP	S BG Fn BGS FnS PDAREA addr,addr	[partition]	AR

Note: There must be no blank between the first and the second operand.

blank

If no dump area operand is given, the following is printed:

- the contents of the general registers, and
- the contents of all real and virtual partitions in which programs are running.

The contents of both real and virtual partitions are in consecutive order in the listings produced; invalid address spaces are indicated.

S

this operand results in the same output as above; it also causes a dump of the supervisor area.

in first operand field

BG or Fn

the contents of the specified partition and its associated registers are printed. If a program is running in real mode in the specified partition, only the real, and not the associated virtual partition is dumped. If a program is running in virtual mode in the specified partition, the virtual partition plus any fixed pages are dumped.

BGS or FnS

same as if BG/Fn was specified; however, the contents of the supervisor area are also dumped.

PDAREA

the PD-area table, the PD area, and the PD alternate address area, if present, are printed.

addr,addr — specifies the start and end addresses of the storage area to be dumped. The contents of the general registers that are associated with the specified storage area are also printed.

If the specified addresses are within an invalid address area, the command is ignored and a message to this effect is issued.

If the storage area crosses the boundary between a valid and an invalid address space, only the contents of the specified valid address space are dumped, together with the general registers that are associated with the valid address area. A message to this effect is issued.

If the storage area to be dumped crosses partition boundaries, the specified storage area is dumped, together with the general registers that belong to the partition in which the starting address is located.

partition — indicates the partition to which SYSLST is assigned for output of the dump. If it is omitted, SYSLST assigned to the background partition is used.

DVCDN
device down

The DVCDN (Device Down) command serves to inform the system that a device is no longer available for system operation. It is used when a device is to be serviced or when a device becomes inoperative.

If a standard or temporary assignment exists for the device, any logical units assigned to it are unassigned. The DVCDN command does not close files associated with logical units, and after the DVCDN command has been issued, files on a DASD or diskette unit cannot be closed or reassigned to another DASD or diskette unit. Therefore, if the unit is a DASD or diskette unit, first attempt to close any files associated with logical units currently assigned to the device, using the CLOSE command.

If an alternate assignment exists for the device, it is removed when the DVCDN command is issued.

When the device has become operative, you must first issue a DVCUP command before the device can again be put to use.

The DVCDN command makes use of supervisor services that prevent other operator communication during execution of this command.

Operation	Operands	Type
DVCDN	X'cuu'	JC

X'cuu' the channel and unit number (in hexadecimal) of the device to be made unavailable.

DVCUP
device up

The DVCUP (Device Up) command serves to inform the system that a device which was inoperative is now again available for system operation. Because the assignments for the device in question were removed by the preceding DVCDN command, the device must be reassigned by an ASSGN statement or command.

The DVCUP command makes use of supervisor services that prevent other operator communication during execution of this command.

Operation	Operands	Type
DVCUP	X'cuu'	JC

X'cuu' the channel and unit number (in hexadecimal) of the device to be made available.

ENDSD
end SD-aids

This command is used to end execution of the SDAID program.

This program and the ENDSD command are used in debugging procedures, and described in detail in *DOS/VS SADP*, GC33-5380.

Operation	Operands	Type
ENDSD	blank	AR

The ENDSD command has no operand.

EXEC
execute program or procedure

See also Procedures 18 and 25

The EXEC command indicates either

- that the end of control information for a job step and the beginning of execution of a program has been reached; in this case it must be the last command processed before a job step is executed, or
- that a cataloged procedure is to be retrieved from the procedure library by the job control program; in this case other commands or statements may follow EXEC.

Operation	Operands	Type
EXEC	[[PGM=]progname][,REAL] [,SIZE= {nK AUTO (AUTO,nK)}]	JC
EXEC	PROC=procname[,OV]	JC

PGM=progname — progname represents the name of the program in the core-image library. It can be from one to eight alphameric characters. If the program to be executed has just been processed by the linkage editor, progname is omitted.

REAL — indicates that the job step started by EXEC will be executed in real mode. If REAL is not specified, the job step is executed in virtual mode.

SIZE= nK
AUTO
(AUTO,nK) — If SIZE is omitted, the entire virtual partition is used for the job to be executed.

The following restrictions apply to n:

- n must not be larger than the size of the partition to which it refers.
- n must be greater than zero.
- n should be a multiple of two. If it is not, the system uses n+1.
- If AUTO is specified, the size of the program is inserted as value in SIZE. If (AUTO, nK) is specified, nK are added to the size of the program and the result is inserted as value in SIZE. The total is rounded upward to a multiple of 2K.

Specified together with REAL, SIZE specifies the size of that part of the real partition that is needed by the job step to be executed. If SIZE is omitted and REAL is specified, the entire real partition is used by the job step.

Specified without REAL, SIZE specifies that the virtual partition to be used by the job step is divided into two parts: the lower part, with a size of nK, will contain the program to be executed; the upper part serves as an additional storage pool for other modules or data required by the program in that partition.

SIZE (without REAL) must always be specified for VSAM programs and for ISAM programs that use the ISAM Interface Program (IIP), and for programs using RPS with other DASD access methods.

PROC=procname procname represents the name of the procedure to be retrieved from the procedure library. It can be from one to eight alphameric characters, the first one of which must be alphabetic.

OV indicates that overwrite statements follow the EXEC command.

Note that the EXEC PROC command and any following overwrite statements **must** be entered from SYSLOG.

HOLD
hold assignments

The HOLD command, if issued prior to the UNBATCH command, causes the assignments for the specified foreground partition(s) to remain in effect until the end of the **next** job.

Operation	Operands	Type
HOLD	Fn[,Fn]...	JC

The HOLD command must have at least one operand; it may have as many operands as there are foreground partitions in your system. The operands can be specified in any order. The first operand must not be preceded by a comma.

IGNORE
ignore a condition

When an abnormal condition exists in the system, you will be notified by an appropriate message on SYSLOG. Depending on the situation, you may want to ignore this condition.

The IGNORE command is a possible response to certain messages that appear on SYSLOG. Therefore, make a habit of looking up the appropriate operator action for **each** message on SYSLOG in *DOS/VS Messages*, GC33-5379.

Operation	Operands	Type
IGNORE	blank	AR,JC

The IGNORE command has no operand.

LFCB
Load forms control buffer

See also Procedure 11

The LFCB command can be used from the console printer-keyboard (or from the display console) to load the forms control buffer (FCB) for a 3203, PRT1 (3211 or 3203-4), or 5203 printer.

This command is very useful in an installation using POWER/VS. When new forms are required while POWER/VS is printing spooled output on a printer with an FCB, the operator must normally load a new FCB image into the printer's FCB. If, at that time, none of the POWER/VS controlled partitions is waiting for work, the SYSBUFLD program cannot be used. Therefore, the only possible way of loading the required FCB image is to type in an LFCB command on the console printer keyboard.

Use this command with caution when the printer is in operation, because there is no way of predicting the end of the output that is being printed with the current FCB image. Use this command if, for example, printed output for a program was started with the wrong FCB image. You will thus be able to correct the output by issuing this command.

Do not use the LFCB command if DOS/VS is operating in a real-time environment, that is, if teleprocessing is being used, or if time dependent equipment such as the IBM 1419 or 1275 is running.

Operation	Operands	Type
LFCB	X'cuu',phasename[,FORMS=xxxx] [,LPI=n][,NULMSG]	AR

X'cuu' specifies the channel and unit number of the printer whose FCB is to be loaded.

phasename specifies the name of the core image library phase that contains the applicable control buffer image. Normally, this name is provided by your programmer.

FORMS=xxxx specifies the form number xxxx of the paper that is to be used together with the new FCB image. for xxxx a string of one to four alphameric characters must be specified. If the new FCB image requires a change of forms, this operand must be given to ensure proper system operation.

LPI=n indicates (for a printer other than the 3211) the required setting of the carriage clutch. For n, you can specify either 6 (six lines per inch) or 8 (eight lines per inch).

NULMSG specifies that the printing of a buffer load verification message is to be suppressed. If this operand is specified, the system continues processing immediately after the FCB load operation has been completed, allowing you no chance to verify that the proper forms have been placed on the printer.

LISTIO
list current I/O
assignments

See also Procedure 13

The LISTIO command produces a listing on SYSLOG of the current I/O assignments. The operands of the command make it possible to produce a listing either of all or of a selected set of the current I/O assignments. Format and meaning of the output of the LISTIO command are described in Procedure 13.

The LISTIO command makes use of supervisor services that prevent other operator communication during execution of this command.

Operation	Operands	Type
LISTIO	ALL partition PROG SYS SYSxxx DOWN UNITS X'cuu' UA	JC

ALL — lists all logical units and the physical units assigned to each.

partition — lists all logical units of the partition and the physical units assigned to each.

PROG — lists all background programmer logical units and the physical units assigned to each.

SYS — lists all background system logical units and the physical units assigned to each.

SYSxxx — lists the physical units assigned to the specified logical unit. SYSOUT and SYSIN are invalid in this command.

DOWN — lists all physical units specified as inoperative.

UNITS — lists all physical units and the logical units assigned to each.

X'cuu' — lists the specified physical unit and the logical units assigned to it.

UA — lists all physical units not currently assigned to a logical unit.

LOG
log control statements and commands

See also procedure 16

The LOG command causes all job control statements and commands to be listed (logged) on SYSLOG until a NOLOG command is entered.

Operation	Operands	Type
LOG	blank	JC, AR

The LOG command has no operand.

LUCB
load universal-character-set buffer

See also Procedures 11 and 12

The LUCB command can be used to load the universal-character-set buffer (UCB) of a UCS printer from the console printer-keyboard (or from the display console).

This command is very useful in an installation using POWER/VS. When new forms are required while POWER/VS is printing spooled output on a printer with a UCB, the operator may have to load a new UCB image into the printer's UCB. If, at that time, none of the POWER/VS controlled partitions is waiting for work, the SYSBUFLD program cannot be used, nor can the UCS command be given, if you are using an IBM 1403 printer. Therefore, the only possible way of loading the required UCB image is to type in an LUCB command on the console printer keyboard.

Use this command with caution while the printer is in operation, because there is no way of predicting the end of the output that is being printed with the current UCB image. It is recommended that you use this command if, for example, printed output for a program was started with the wrong UCB image. You will thus be able to correct the output by issuing this command.

Do not use the LUCB command if DOS/VS is operating in a real-time environment, that is, if teleprocessing is being used or if time dependent equipment such as the IBM 1419 or 1275 is running.

Operation	Operands	Type
LUCB	X'cuu',phasename[,FOLD] [,NOCHK][,TRAIN=xxxxxx] [,NULMSG]	AR

X'cuu' specifies the channel and unit number of the printer whose UCB is to be loaded.

phasename specifies the name of the core image library phase that contains the applicable control buffer image. Normally, this name is provided by your programmer.

FOLD causes lowercase character codes to be printed as uppercase characters (use this operand only if requested by your programmer).

NOCHK causes data checks resulting from mismatches of printline characters with the UCB image to be ignored (use this operand only if it is requested by your programmer).

TRAIN=xxxxxx indicates that the print train (or chain for a 1403) identified by xxxxxx is to be mounted on the printer. The train (or chain) identification xxxxxx may be up to six characters long. The system inserts this operand in a console message. If a new train (or chain) must be installed, this operand is required to ensure proper system operation.

NULMSG specifies that the printing of a buffer load verification message is to be suppressed. If this operand is specified, the system continues normal processing immediately after the UCB load operation has been completed, allowing you no chance to verify that the contents of the UCB match the train (or chain) mounted on the printer.

MAP
produce map of real and virtual storage

See also Procedure 14

The MAP command produces, on SYSLOG, a map of real and virtual storage in a multiprogramming system.

Operation	Operands	Type
MAP	blank	JC, AR

The MAP command has no operand.

The format of the output of the MAP command is shown in *DOS/VS System Control Statements*, GC33-5376.

MODE
error recording

The MODE command initiates or suppresses HIR (Hardware Instruction Retry) and ECC (Error Correction Code) error recording. It also sets the EFL (Error Frequency Limit) for HIR and ECC, or inquires about the status of HIR and channel check handling error recording.

Details and format of the MODE command are given in *DOS/VS SADP*, GC33-5380, under *Hardware Error Recording and Recovery.*

MSG
transfer control to operator

The MSG (message) command enables the operator to communicate with a selected partition. The program running in the partition with which communication is desired must contain an operator communication exit routine.

The operator will always get detailed instructions from his programmer as to when to issue a MSG command and what to enter after having issued the command.

Operation	Operands	Type
MSG	[partition]	AR

'partition' specifies the partition with which communication is desired.

MTC
magnetic tape control

The MTC (magnetic tape control) command enables the operator to perform tape operations such as rewinding, unloading, skipping files. The MTC command is a useful tool to perform tape control operations from the console printer-keyboard or from the display console.

Operation	Operands	Type
MTC	opcode,{SYSxxx X'cuu'} [,nn]	JC

opcode specifies the operation to be performed as explained in the table below.

SYSxxx specifies the logical unit to which the tape is assigned.

X'cuu' specifies the channel and unit number (in hexadecimal).

nn is a decimal number from 01 through 99 that indicates the number of times the specified operation is to be performed. If omitted, 01 is assumed.

Opcode	Meaning	Possible Use
BSF	Backspace File	Backspace one file so tape is positioned for reading the tapemark preceding the file backspaced.
BSR	Backspace Record	Backspace record.
DSE	Data Security Erase	Erase data security.
ERG	Erase Gap	Erase gap.
FSF	Forward Space File	Used when restarting a program. The tape is positioned beyond tapemark following the file spaced over.
FSR	Forward Space Record	Locate a specific record within a file.
RUN	Rewind and Unload	Rewind and unload a tape on a specific unit.
REW	Rewind	Rewind a tape on a specific unit.
WTM	Write Tape Mark	Write a tapemark on an output file.

NEWVOL
new volume mounted

The NEWVOL command is used to inform the system that a volume previously requested by the system has been mounted, and that processing can continue.

If an assignment specifying VOL= has been given for a disk or tape unit and the system cannot find the requested volume on that unit, then the system prints a message on SYSLOG requesting the operator to mount the desired volume. After mounting the proper volume, the operator issues the NEWVOL command to indicate that processing with the new volume may continue.

Operation	Operands	Type
NEWVOL	[partition]	AR

Partition (for example BG or F2) specifies the partition for which the new volume was mounted.

If no operand is specified, BG is assumed. If the specified partition is not waiting for a volume to be mounted, an error message is printed on SYSLOG.

NOLOG
suppress logging

See also Procedure 16

The NOLOG command suppresses listing (logging) of job control statements and commands on SYSLOG until a LOG command is entered. NOLOG does not, however, suppress the listing of the job control statements and commands given below nor of control statements in error. The statements and commands are:

ALLOC
ALLOCR
DVCDN
DVCUP
HOLD
JOB
MAP
PAUSE
STOP
EOJ (/&)
EOP (/+)
*

Operation	Operands	Type
NOLOG	blank	JC, AR

The NOLOG command has no operand.

OVEND
end of overwrite
statements

See also Procedure 18

The OVEND command indicates the end of overwrite statements for a cataloged procedure.

Operation	Operands	Type
OVEND	[comments]	JC

The OVEND command must follow the overwrite statements (on SYSRDR or SYSLOG). It has no operands. Any text following the operation code is treated as comments.

PAUSE
interrupt job-control processing

See also Procedure 17

The PAUSE command interrupts job-control processing **at the end** of the current job or job step. It enables you to perform certain functions, such as mounting tapes or disk packs, changing the printer carriage control tape.

When processing is interrupted by a PAUSE command, the keyboard is unlocked so that the operator can, for instance, indicate on the log what function he has performed. Any information typed in during this time is **not** processed by the system. Processing continues when the operator presses END on the console printer-keyboard or ENTER on the display console. The PAUSE command is therefore ignored if no console printer-keyboard or display console is available.

Operation	Operands	Type
PAUSE	[comments]	JC
PAUSE	partition[,EOJ]	AR

The PAUSE job control command is used within the job stream read from SYSRDR or SYSIN only. That is to say, it is used if the operator knows beforehand that processing is to be interrupted. If a PAUSE job control command is issued, job control will be interrupted the **next** time job control is called, that is, at the end of the current job step. The PAUSE job control command has no operand.

The PAUSE attention command can be given at any time for any partition in which processing is to be interrupted. If no partition is specified, BG will be assumed.

If the optional operand EOJ is specified, the interruption will occur at the end of the current **job**. In this case the partition operand must be present and must precede the EOJ operand. If the EOJ operand is omitted the interruption will occur at the end of the current **job step.**

Note: A PAUSE job control **statement** (with // in columns 1 and 2) will cause an **immediate** interruption in the partition in which it is issued.

PRTY
display or modify partition priority

The PRTY command serves to display or to modify the priority of partitions.

Operation	Operands	Type
PRTY	blank	AR
PRTY	partition [,partition]...	AR

The PRTY command without an operand displays, on SYSLOG, the current priorities of all partitions. The first partition has the lowest priority, the last one the highest.

The PRTY command with operands modifies the priorities of the partitions. The sequence of the partition identifiers indicates the desired new sequence. The first partition specified is given the lowest, and the last one specified is given the highest priority. If you use more than one operand, separate them by a comma. The number of operands specified must coincide with the number of partitions your system has. The command

```
PRTY BG,F3,F1,F2
```

for instance, sets the background partition with the lowest, and F2 with the highest priority in a four-partition system.

If teleprocessing balancing is active, the partition(s) that can be subject to performance degradation will be displayed on SYSLOG, regardless of whether operands are specified with the PRTY command or not. See also the TPBAL command.

RESET
reset I/O assignments

See also Procedure 13

The RESET command resets temporary I/O assignments either to the last preceding permanent assignment, if any, made via an ASSGN command without the TEMP option or to the standard assignment made during system generation. If neither a permanent nor a standard assignment exists, the temporary assignment is reset to UA (unassigned).

Resetting is performed only for the temporary assignments within the partition for which the RESET is given.

Operation	Operands	Type
RESET	$\left\{\begin{array}{l}\text{SYS}\\\text{PROG}\\\text{ALL}\\\text{SYSxxx}\end{array}\right\}$	JC

SYS — resets the temporary assignments of all system logical units of the partition to their permanent or standard assignment.

PROG — resets the temporary assignments of all programmer logical units of the partition to their permanent or standard assignment.

ALL — resets the temporary assignments of all logical units of the partition to their permanent or standard assignment.

SYSxxx — resets the temporary assignment of the specified logical unit to its permanent or standard assignment. SYSIN and SYSOUT are invalid for this operand.

ROD
record on demand

See also Procedure 8

The ROD command writes statistical information to the recorder file (SYSREC) that has been accumulated since the last IPL or since the last time the ROD command was given. The ROD command should be issued each time before the system is shut down or before you re-IPL the system.

Operation	Operands	Type
ROD	blank	JC

The ROD command has no operand.

If your system uses the Reliability Data Extractor (RDE) and the ROD command is issued, the system issues the message on SYSLOG:

```
1I90A END OF DAY =
```

You may respond with Y (for YES) or N (for NO). Any other response causes an error message and the same message as before to be issued:

```
1I92I INVALID CODE
1I90A END OF DAY =
```

When the response is Y, the accumulated statistical information is added to the recorder file and an end-of-day (EOD) record is written.

When the response is N, the accumulated statistical information is added, but no EOD record is written.

SET
set values

The SET command is used to (1) initialize the date, clock, and UPSI byte, (2) to specify the number of lines to be printed on SYSLST and the remaining disk capacity if either SYSLST or SYSPCH is assigned to disk, and (3) to define the status of the system-recorder and the hard-copy file.

Operation	Operands	Type
SET	[DATE=v1,CLOCK=v2][,ZONE=v3]	IPL
SET	[UPSI=n1][,LINECT=n2][,RCLST=n3] [,RCPCH=n4][,RF=n5][,DATE=n6][,HC=n7]	JC
SET	SVA=n8	JC
SET	SDL=n9	JC

The IPL command SET is mandatory. It must follow the ADD and DEL commands, if any, and precede the CAT (if present) and DPD commands.

v1 specifies the date in either of the following two formats:

mm/dd/yy
dd/mm/yy

where mm specifies the month (01-12), dd the day (01-31), and yy the last two digits of the year.

The format to be used is selected during system generation.

When and where DATE and CLOCK should be specified, depends on your system. For details refer to *DOS/VS System Control Statements*, *GC33-5376.*

v2 specifies the local time of day in the format

hh/mm/ss

hh specifies hours (00-23), mm specifies minutes (00-59), and ss specifies seconds (00-59).

v3 specifies the installation's geographical position as either east or west of Greenwich in either of the two following formats:

EAST/hh/mm
WEST/hh/mm

where hh/mm indicates the difference between local time and Greenwich Mean Time in hours (00-12) and minutes (00-59).

n1 sets the bit configuration of the UPSI byte in the communication region. The programmer will tell you when this is necessary. n1 consists of one to eight digits, either 0, 1, or X. Positions containing 0 are set to 0, positions containing 1 are set to 1, and positions containing X remain unchanged. Unspecified rightmost positions are assumed to be X.

n2 sets the standard number of lines to be printed on each page of SYSLST. n2 is an integer number between 30 and 99.

SET (continued)

n3 specifies the minimum number of records remaining to be written on SYSLST (when assigned to disk) before a warning message is issued to the effect that the capacity of the extent has almost been exhausted. n3 can be any decimal number from 100 to 65535.

n4 specifies the minimum number of records remaining to be written on SYSPCH (when assigned to disk) before a warning message is issued to the effect that the capacity of the extent has almost been exhausted. n4 can be any decimal number from 100 to 65535.

Note: When system files are assigned to diskette, the RCLST= and RCPCH= parameters are accepted, but there will be no warning. Because of the multi-volume capability there is no need for a warning message.

See also Procedure 9

n5 defines the status of the system recorder file on SYSREC. It can be:

YES indicates that an active recorder file exists and that it can be opened as an output file. The file is opened when the first // JOB card has been read.

CREATE instructs the system to create a system recorder file when the first // JOB card is encountered. If you want to do this, issue the SET command with RF=CREATE **before** you run your first job.

A SET RF command is ignored when it is issued for a Model 115 or 125 without software recording.

n6 sets the system date permanently to the specified value. The system date in the communication region of each partition is reset to the new value. This subsequently resets the JOB date when a new job is run. n6 has one of the following formats:

mm/dd/yy
dd/mm/yy

where mm specifies the month, dd specifies the day, and yy specifies the year. The format to be used is selected during system generation.

The DATE parameter may be specified only if the TOD clock is not supported in the system or if the clock is not operational.

n7 defines the status of the hard-copy file on SYSREC. n7 can have the following values:

See also Procedure 10

YES indicates that a hard-copy file exists in the system and that it is to be opened as soon as the first // JOB card has been read. YES is the default value.

NO indicates that no recording is to be performed on the hard-copy file. HC=NO can be specified only if the IBM 5213 Console Printer is attached.

CREATE instructs the system to create a hard-copy file; the file is created and opened as soon as the first

// JOB card has been read.

n8 allows the user to change the size of the optional SVA from that specified in the supervisor. The SVA option must be specified as the first statement after IPL (or as the first statement of the first procedure). n8 has the following format: SVA=(nK,mK). The first parameter specifies the size of the SVA (Shared Virtual Area) including the SDL (System Directory List) and the system GETVIS area. This size should be an even value and at least 64K. The second parameter specifies the size of the system GETVIS area. This parameter can be 0K; if a value is given, it should be a multiple of 2K and smaller than the size of the SVA. The recommended values for a system with VSAM are:

```
SET SVA=(302K,0K)
```

SET SVA may only be specified as the first statement after IPL (or as the first statement of the first procedure).

n9 causes job control to build a system directory list in the SVA. It can also make it possible to load phases in the SVA. The format of this option is SDL=CREATE.

SET SDL may only be specified after SET SVA or as the first statement after IPL.

The phasenames to be included in the SDL are entered following the SET SDL command in the form

```
phasename[,SVA]
```

where SVA indicates that the phase is to be placed in the shared virtual area if the phase is SVA-eligible. The end of the list of phasenames is indicated by /*. The SET SDL command, phasenames, and /* can be placed in a cataloged procedure.

The VSAM SVA-eligible phases are listed in *DOS/VS System Generation*, GC33-5377.

START
start or continue processing

See also Procedure 16

The START command serves to start or continue processing in one of the foreground partitions or to continue processing in the background partition. The function of the START command is exactly the same as that of the BATCH command.

Operation	Operands	Type
START	[partition]	AR

If the specified partition has a minimum size of 64K bytes, the job control program reads the next command or statement from SYSLOG. If the specified partition was temporarily halted by a STOP command, job control also reads the next command or statement from SYSLOG. This may be any job control statement or command.

If the specified partition is already active and has not been temporarily halted when the START command is issued, the system responds with the message:

```
AREA NOT AVAILABLE
```

If no operand is specified for the START command, the background partition is assumed.

STOP
stop processing in a partition

See also Procedure 17

The STOP command stops processing in one partition of a multipartition system (it cannot be used in a single-partition system).

It must be issued at the time job control is running in the partition in which processing is to be suspended. Note that the stopped partition remains active. The UNBATCH command must therefore be used if the partition is to be made inactive (for example, to issue the ALLOC or ALLOCR command).

After the STOP command has been given, processing can be continued by issuing the BATCH or the START command.

Operation	Operands	Type
STOP	blank	JC

The STOP command has no operand.

TPBAL
display or modify TP balancing

The TPBAL command is used to display or modify TP (teleprocessing) balancing. The TP balancing facility serves to improve teleprocessing response time, when the average response time exceeds tolerable limits. This improvement in teleprocessing response time is achieved by diverting a certain amount of processing time from other partitions of lower priority to the partition containing the teleprocessing program.

TP balancing should only be considered if all of the following criteria are met:

- Your system includes both batch processing and teleprocessing.
- Paging occurs in your system.
- You are prepared to accept delayed processing in the partition(s) selected for performance degradation.
- The eligible partition(s) have the lowest priority and contain at least one task running in virtual mode.

Note: No real mode processing can be delayed. TP balancing ignores any request to degrade the performance of a partition in which no virtual processing is taking place.

Operation	Operands	Type
TPBAL	blank	AR
TPBAL	n	AR

Without an operand, the TPBAL command displays on SYSLOG the partition(s) in which TP balancing can cause performance degradation. The first partition selected for degradation is always the partition of lowest priority, followed by the partition with the next lowest priority, and so on. If TP balancing is not currently active, the following response to the TPBAL command appears on SYSLOG:

```
TPBAL NONE
```

With an operand, the TPBAL command modifies TP balancing and displays the partitions that will be affected by the new setting. The only valid operand is a decimal number from zero up to the number of partitions in the system minus one. As an example, the following specification of the TPBAL command subjects the two partitions of lowest priority in the system to performance degradation:

```
TPBAL 2
```

A specification of zero deactivates the TP balancing facility.

UCS
load universal character set buffer

The UCS command causes the 240-character Universal Character Set, contained in the core-image library and specified by 'phasename', to be loaded as buffer storage in the IBM 2821 Control Unit. The 240 EBCDIC characters correspond to the 240 print characters on 1403 chains and trains.

See also Procedures 11 and 12

A character sent to the printer for printing is compared with the characters in the UCS buffer. When a match occurs, the corresponding chain or train character is printed in the print-line position that was occupied by the output character. The UCS buffer and the variety of chains or trains that are available thus permit the IBM 1403 printer to be adapted to a number of different printing applications.

The logical unit must be assigned to a 1403 printer with the UCS feature. The UCS program must be available, and the new chain or train must be mounted before the UCS command is executed. The UCS command is not logged on SYSLST.

Operation	Operands	Type
UCS	SYSxxx,phasename[,FOLD][,BLOCK][,NULMSG]	JC, AR

SYSxxx — specifies the name of the logical unit assigned to a 1403 UCS printer to be loaded.

phasename — specifies the symbolic name of the core image library phase containing the 240 EBCDIC characters to be loaded, followed by an 80-character verification message. Each phase may have any valid phase name.

The UCS command cannot be used for the 3211, 3203, or 5203 printers.

FOLD — specifies that the buffer is to be loaded with the folding operation code in the CCW to permit printing of either uppercase or lowercase bit configurations.

BLOCK — specifies that the 2821 latch is to be set to inhibit data checks generated by the 1403 UCS printer because of print-line character mismatches with the UCS buffer.

NULMSG — specifies that the 80-character verification message is not to be printed on the 1403 after the buffer is loaded. If this parameter is not specified after the UCS buffer has been loaded, the program skips to channel 1, issues a print of the last 80 characters in the phase specified by the first parameter, and again skips to channel 1. This is to identify the phase, if the phase name is incorporated in the verification message. If a train or chain can be identified by a unique character, this message can also be used to verify that the mounted train or chain is compatible with the contents of the UCS buffer, by including the character in the verification message.

UNBATCH
terminate processing in a foreground partition

See also Procedure 17

The UNBATCH command terminates processing in the foreground partition for which it is given.

When the UNBATCH command is issued, processing is terminated, all logical I/O units except SYSLOG, SYSRES, SYSREC, SYSVIS, and SYSCAT are unassigned, and the partition is no longer active. Because logical units are unassigned, all tape, disk, and diskette files must previously have been closed (via a CLOSE command).

The partition can subsequently be made active again (that is, restarted) by issuing the BATCH or START command.

Operation	**Operands**	**Type**
UNBATCH	blank	JC

The UNBATCH command has no operand.

Job Control Statements

Job control statements are used by the programmer to request and control system functions required during the execution of his programs. You will rarely have to prepare job control statements; the functions you need are available through the operator commands.

It may occasionally happen, however, that you are asked to run a certain job and to prepare the accompanying job control statements. It will further happen that the system detects errors in the job control statements provided by the programmers. Being familiar with the format of these statements will enable you to correct obvious errors, thus saving the job from having to be canceled and resubmitted. This is why at least the format of each job control statement is shown in this section, even for those statements that you will most probably not prepare yourself.

Figure IV-5 is a list of all DOS/VS job control statements. This list is followed by a short description of the functions and format of each statement. For statements and operands that are always specified by the programmer, reference is made to the appropriate DOS/VS publication where more detailed information can be found if desired.

Job Control Statement	Function
// ASSGN	Assign a logical unit
// CLOSE	Close a logical unit
// DATE	Set the date in the communication region
// DLAB	Define DASD label
// DLBL	Define DASD or diskette label
// EXEC	Execute a program or procedure
// EXTENT	Define a DASD extent or diskette volume
// JOB	Beginning of a new job
// LBLTYP	Reserve storage for label processing
// LISTIO	List current I/O assignments
// MTC	Perform magnetic tape control operations
// OPTION	Specify options for program processing and output
// OVEND	End of overwrite statements for cataloged procedures
// PAUSE	Interrupt job control processing
// RESET	Reset I/O device assignments
// RSTRT	Restart a program after a checkpoint
// TLBL	Define a tape label
// TPLAB	Define a tape label
// UPSI	Set UPSI byte in the communication region
// VOL	Supply volume information for tape and DASD
// XTENT	Define a DASD extent
// ZONE	Initialize value of job zone field in the communication region
/*	End of file
/&	End of job
/+	End of procedure
*	Comments

Figure IV-5. List of all DOS/VS job control statements

// ASSGN
assign a logical unit name

Format and effect of the // ASSGN job control statement when PERM is not specified, are exactly the same as of the ASSGN job control command with the TEMP option. When PERM is specified, format and effect of the // ASSGN job control statement is the same as the ASSGN job control command without the TEMP option.

Operation	Operands			Type
ASSGN	SYSxxx,	X'cuu' (address-list) UA IGN SYSyyy device-class device-type	[,TEMP ,PERM ,VOLvolserno ,SHR ,X'ss' ,ALT ,H1 ,H2]	JC

// CLOSE
close a logical unit

Format and effect of the // CLOSE job control statement are exactly the same as those of the CLOSE job control command.

Operation	Operands	
// CLOSE	SYSxxx	[,X'cuu'[,X'ss'] ,UA ,IGN ,ALT]

// DATE
set date

The // DATE job control statement places the specified date temporarily in the communication region. This date overrides the date given in the SET command, either during or after IPL. The date can be used by the program for identifying printed output, for job accounting.

The date specified in the // DATE job control statement is reset at the end of the job to the date specified in the last SET command.

Operation	Operands
// DATE	dd/mm/yy mm/dd/yy

dd specifies the day of the month, mm the month, and yy the year.

Dates of jobs or job steps that execute past midnight will not be incremented.

// DLAB

The // DLAB job control statement is included for reasons of compatibility. The // DLBL job control statement should be used in its place.

The // DLAB job control statement is entered in the form of two cards. The format is:

Operation	Operand	Col. 72
// DLAB	'label-fields 1-3',	X
blank	xxxx,yyddd,yyddd,'system-code'[,type]	

label-fields consists of 51 characters. The first operand in the second card must start in column 16. VSAM does not support the DLAB statement.

// DLBL
define DASD or diskette label

// DLBL job control statements are normally prepared by the programmer because he is familiar with the format and purpose of the DASD or diskette labels to be used for his files. Detailed information on this statement can be found in *DOS/VS System Control Statements*, GC33-5376.

Operation	Operands
// DLBL	filename,['file-id'],[date],[codes][,DSF] [,BUFSP=][,CAT=filename][,BLKSIZE=n]

// EXEC
execute a program or a procedure

See also Procedure 18

Format and effect of the // EXEC job control statement are exactly the same as those of the EXEC job control command.

The // EXEC PROC statement may be entered either from SYSRDR or SYSLOG. If entered from SYSLOG, any overwrite statements must be entered from SYSRDR.

Operation	Operands
// EXEC	[[PGM=]progname][,REAL] $\left[\text{,SIZE=}\begin{Bmatrix}\text{nK}\\ \text{AUTO}\\ \text{(AUTO,nK)}\end{Bmatrix}\right]$
// EXEC	PROC=procname[,OV]

// EXTENT
define a DASD extent or diskette volume

// EXTENT job control statements are normally prepared by the programmer because he is familiar with DASD or diskette organization and knows exactly where he wishes his programs and data to reside on a disk pack or on a diskette. Detailed information on this statement can be found in *DOS/VS System Control Statements*, GC33-5376.

Operation	Operands
// EXTENT	[symbolic-unit], [serial-number], [type], [sequence-number], [relative-track], [number-of-tracks], [split-cylinder-track], [B=bins]

// JOB
beginning of a new job

The // JOB job control statement indicates the beginning of a job, that is to say, the beginning of control information for the job. If your system uses the TOD feature, the date and time of day appear in positions 73-100 when the // JOB statement is printed on SYSLST. The date and time of day is also printed in positions 1-28 on the next line on SYSLOG.

Operation	Operands
// JOB	jobname [accounting information]

jobname, which must always be specified, is the name of the next job (not of the program). It consists of a maximum of eight alphameric characters.

Accounting information may contain 16 characters of user-specified job accounting information. Entries beyond this field are considered as comments. For details, refer to *DOS/VS System Control Statements*, GC33-5376.

// LBLTYP
reserve storage for label processing

// LBLTYP job control statements are normally prepared by the programmer because he knows the exact amount of storage that has to be reserved for label processing either during link-editing or during execution of the program. Detailed information on this statement is contained in *DOS/VS System Control Statements*, GC33-5376.

Operation	Operands
// LBLTYP	$\left\{\begin{array}{l}\text{TAPE[(nn)]}\\\text{NSD(nn)}\end{array}\right\}$

// LISTIO
list current I/O assignments

Format of the // LISTIO job control statement is the same as that of the LISTIO job control command. The effect of the // LISTIO job control statement, however, differs from that of the LISTIO command in that it produces the listing on SYSLST and not on SYSLOG.

Operation	Operands
// LISTIO	SYS PROG BG Fn ALL SYSxxx UNITS DOWN UA X'cuu'

// MTC
magnetic tape control

The effect of the // MTC job control statement is the same as that of the MTC job control command. Note that X'cuu' is not valid for the // MTC job control statement.

Operation	Operands
// MTC	opcode,SYSxxx[,nn]

Note: X'cuu' instead of SYSxxx is not valid for the job control statement.

// OPTION
job control options

The // OPTION job control statement specifies one or more options to be used during execution of a program or for its output. The operands may be specified in any order. The options specified remain in effect for the duration of the current job.

Operation	Operands
// OPTION	option1[,option2] . . .

The three periods (. . .) mean that as many operands as desired can be specified.

ACANCEL ACANCEL, when used with NOLOG, causes the job to be cancelled after an unsuccessful attempt to assign a device. The cancellation is preceded by one of the messages 1A1nl, 1A5nl, 1A6nl, or 1A7nl, depending on the type of error.

NOACANCEL prevents job cancellation in case of an assignment error. The system then waits for operator action.

If neither ACANCEL or NOACANCEL is included, the default option selected at system generation is in effect.

ALIGN used by programmers only.
NOALIGN suppresses the ALIGN option.

CATAL causes one or more phases to be cataloged into the core image library upon completion of a linkage editor run. It also causes the LINK option to be set.

DECK used by programmers only.
NODECK suppresses the DECK option.

DUMP causes a dump of the registers, of the supervisor, and of the virtual or real partition to be printed on SYSLST in case of abnormal program termination.

PARTDUMP causes a dump of the registers, of supervisor control blocks, and of the pertinent virtual or real partition to be printed on SYSLST if an abnormal program termination occurs.

NODUMP suppresses the DUMP option.

How and when to use this option is described in *DOS/VS SADP*, GC33-5380.

EDECK used by programmers only.
NOEDECK suppresses the EDECK option.

ERRS used by programmers only.
NOERRS suppresses the ERRS option.

LINK indicates that the output of a language translator is to be written on SYSLNK for subsequent link-editing. The LINK option must always precede a // EXEC LNKEDT statement in the input stream. (CATAL also causes the LINK option to be set.)
NOLINK suppresses the LINK option.

LIST used by programmers only.
NOLIST suppresses the LIST option.

LISTX used by programmers only.
NOLISTX suppresses the LISTX option.

LOG causes listing of columns 1-80 of all control statements and commands on SYSLST.
NOLOG suppresses the LOG option.

PARSTD causes all DASD, diskette, or tape labels submitted after this point to be written at the beginning of the partition standard label track.

// OPTION (continued)

PARSTD further causes all file definition statements submitted after this point to be included in the standard file definition set so that they are available to all programs in this partition. The PARSTD option remains in effect (1) until end of job or job step or (2) until // OPTION USRLABEL or // OPTION STDLABEL is specified.

// OPTION PARSTD followed by an end-of-job (/ &) statement clears the partition standard label track.

RLD NORLD	used by programmers only. suppresses the RLD option.
STDLABEL	causes all DASD, diskette, or tape labels submitted after this point to be written at the beginning of the standard label track. STDLABEL further causes all file definition statements submitted after this point to be included in the standard file definition set so that they are available to all programs in all partitions. The STDLABEL option remains in effect (1) until end of job or job step, or (2) until // OPTION USRLABEL or // OPTION PARSTD is specified. // OPTION STDLABEL followed by an end-of-job (/ &) statement clears the standard label track.
SYM NOSYM	used by programmers only. suppresses the SYM option.
SYSPARM	used by programmers only.
USRLABEL	causes all DASD, diskette, or tape labels submitted after this point to be written at the beginning of the user label track.
XREF NOXREF	used by programmers only. suppresses the XREF option.
48C/60C	used by programmers only.

// OVEND
end of overwrite statements for cataloged procedures

Format and effect of the // OVEND job control statement are exactly the same as those of the OVEND job control command.

Operation	Operands
// OVEND	[comments]

// PAUSE
interrupt job-control processing

The // PAUSE job control statement causes an interruption (pause) of the job control program **immediately after** the statement has been processed. Processing continues when the operator presses END on the console printer-keyboard or ENTER on the display console. The // PAUSE statement is therefore ignored if no console printer-keyboard or display console is available.

Any text that appears after the blank following the operation code PAUSE is treated as comments.

The // PAUSE statement is always listed on SYSLOG.

Operation	Operands
// PAUSE	[comments]

The PAUSE **command** causes an interruption **at the end of the current job or job step** in the specified partition.

// RESET
reset I/O assignments

Format and effect of the // RESET job control statement are exactly the same as those of the RESET job control command.

Operation	Operands
// RESET	{SYS PROG ALL SYSxxx}

// RSTRT
restart a program from a checkpoint

The // RSTRT job control statement is issued to restart a program that did not complete execution, that is, to restart it from a checkpoint. This statement is usually prepared by the programmer because restarting a job requires a considerable amount of knowledge about the exact status of the program at the checkpoint (for example, position of magnetic tapes, names of files, partition used).

Operation	Operands
// RSTRT	SYSxxx,nnnn[,filename]

SYSxxx — specifies the name of the symbolic unit that contains the checkpoint records. This unit must have been previously assigned.

nnnn — the four-digit number of the checkpoint from which the file containing the checkpoint records has to be restarted.

filename specifies the name of the file on SYSxxx that contains the checkpoint records. This operand will always be supplied, if necessary, by your programmer.

// TLBL
define a tape label

// TLBL job control statements are normally prepared by the programmer because he knows the format and purpose of the tape labels to be used for his files. Detailed information on this statement can be found in *DOS/VS System Control Statements*, GC33-5376.

Operation	Operands
// TLBL	file-serial-number filename,['file-id'],[date], , set-identifier volume-sequence-number ,[file-sequence-number], file-section-number [generation-number],[version-number]

// TPLAB

The // TPLAB job control statement should no longer be used. See the // TLBL job control statement.

Operation	Operands
// TPLAB	{'label-fields 3-10' 'label-fields 3-13'}

// UPSI
set UPSI byte in communication region

The // UPSI (set User Program Switch Indicators) job control statement allows the programmer to set switches in the communication region that his program can test and react to. The // UPSI statement is therefore used by programmers only. Detailed information on this statement can be found in *DOS/VS System Control Statements*, GC33-5376.

Operation	Operands
// UPSI	nnnnnnnn

// VOL

The // VOL job control statement should no longer be used. See the // DLBL job control statement for disk and diskette files and // TLBL for magnetic tape files.

Operation	Operands
// VOL	SYSxxx,filename

// XTENT

The // XTENT job control statement should no longer be used. See the // EXTENT job control statement.

Operation	Operands
// XTENT	type,sequence,lower,upper,'serial-no',SYSxxx[,B]

// ZONE
set value of zone field in communication region

The // ZONE job control statement initializes the value of the job zone field in bytes 143 and 144 of the communication region. The // ZONE statement is accepted only if time-of-day-clock support is included in the system. Otherwise, the message

1S0nD INVALID STATEMENT

appears on SYSLOG and SYSLST.

If this statement is not supplied, the job control program provides the zone as contained in the system zone field (bytes 68 and 69) of the communication region extension. If no // DATE statement is supplied, the job date is updated by means of the values contained in the system date field and in the // ZONE statement.

Installations that use Greenwich Mean Time (GMT) need not specify the // ZONE statement.

Operation	Operands
// ZONE	$\left\{\begin{matrix}\text{EAST}\\\text{WEST}\end{matrix}\right\}$ /hh/mm

EAST A geographical position east of Greenwich.
WEST A geographical position west of Greenwich.

hh/mm A decimal value indicating the difference in hours and minutes between local time and GMT.

If time-of-day-clock support is provided, the end-of-job statement is printed on SYSLST in the following format:

// ZONE (continued)

Column	Contents
1-3	EOJ
5-12	job name
14-72	blanks or user comments
73-118	date, time of day, and job duration, in the format

DATE mm/dd/yy,CLOCK hh/mm/ss,
DURATION hh/mm/ss

or

DATE dd/mm/yy,CLOCK hh/mm/ss,
DURATION hh/mm/ss

The format depends on the options selected for your installation during system generation.

The date, time of day, and job duration appear on SYSLOG in the same format, occupying 46 positions on the line following the EOJ statement.

If time-of-day-clock support is part of your system, the zone and date values are reset every time this statement is encountered.

/*
end of file

The /* (end-of-file) job control statement indicates the end of the input data for a program (job or job step). It must be placed at the end of the data (on SYSRDR, SYSIPT, or SYSIN) for each program.

Note: If the input stream is entered on an IBM 5425 MFCU, the /* statement must be followed by a blank card.

The /* job control statement has no operand, but may contain comments.

/ &
end of job

The / & (end-of-job) job control statement indicates the end of a job. Therefore, it must be the last statement of each job. Comments, if any, can begin in column 4. These comments are printed on SYSLST, together with the / & statement, unless NOLOG was specified.

/+
end of procedure

The /+ (end-of-procedure) job control statement must be used to indicate the end of a procedure to be cataloged. Column 3 must contain a blank. Any text that appears after column 3 is treated as comments.

*
comment

The * (comment) job control statement can be used to insert comments in the job stream between job control statements. Any text that appears after the blank in column 2 is treated as comments.

Linkage Editor Control Statements

This page shows the general formats of the linkage editor control statements. It is unlikely that you will be concerned with these statements; full details are contained in *DOS/VS System Control Statements*, GC33-5376.

PHASE

The PHASE statement indicates the beginning of a phase.

Operation	**Operands**
PHASE	name,origin[,NOAUTO]

INCLUDE

The INCLUDE statement signals that an object module is to be included.

Operation	**Operands**
INCLUDE	[modulename][,(namelist)]

ENTRY

The ENTRY statement provides an optional transfer address for the first phase.

Operation	**Operands**
ENTRY	[entrypoint]

ACTION

The ACTION statement specifies options to be taken.

Operation	**Operands**
ACTION	{CLEAR,MAP,NOMAP,NOAUTO,CANCEL,BG ,Fn,REL,NOREL}

IPL Reason Codes and ID-Codes

For details on how to enter these codes, see Procedure 8 in the section *Procedures*.

Reason Code	Comments
CE	IBM representative has control of the system for maintenance or testing purposes.
DF	Default.
EN	Environmental problems (power, overheating, for example) caused failure.
IE	IBM hardware or IBM-supplied program error that did not require an IBM representative.
IM	IBM hardware or IBM-supplied program error that required an IBM representative.
ME	Media - hardware error caused by faulty disk pack, reel of tape, cards, and so on.
NM	Normal IPL.
OP	Operational problem - operator error or procedural problem.
UN	Unknown - undetermined error.
UP	A user (non-IBM-supplied) program caused the failure.

Always use ID code 00 with reason codes DF, EN, NM, OP, UN, and UP.

ID Code	Comments
00	Unknown - Must be used with Reason Codes DF, EN, NM, OP, UN, and UP. 00 is the default.
10	Processor - CPU, channel (integrated), storage unit, and so on, failure.
20	DASD - A failure occurred in a DASD unit or its associated control unit (such as, 2311, 2314, 2841, 3330, 3340, 3350).
30	Other - A device without an ID code (such as a paper tape unit) caused the failure.
40	Magnetic Tape - A failure occurred in a magnetic tape unit or its associated control unit (such as, 2401, 2803, 3400, or 3540 Diskette).
50	A failure occurred in a card reader/punch, a printer, or in its associated control unit (such as, 2540, 1403, 2821).
60	MICR/OCR - A magnetic ink character reader (such as, 1417, 1419) or an optical character reader (such as, 1285, 1287) failure.
70	A teleprocessing failure occurred in a teleprocessing control unit (such as, 2701, 2702).
80	Graphic - A video display unit (such as, 2260) or its associated control unit failure.
90	An IBM-supplied program (such as the DOS/VS system or one of its components) failure.
91	An IBM Programming Product failure.

EXEC Statements

The EXEC statement calls a program from the core-image library into storage for execution. The following table lists the most frequently used EXEC statements (1) for language translators, (2) the linkage editor, (3) standard label cylinder display, and (4) the librarian.

1	// EXEC ASSEMBLY	Calls the assembler program.
2	// EXEC LNKEDT	Calls the linkage editor program that edits all programs to be run in the system.
3	// EXEC LSERV	Calls the standard label cylinder display program.
4	// EXEC CSERV	Calls the service program that punches or writes user programs from a core image library.
	// EXEC MAINT	Calls the maintenance program that maintains the libraries (add, delete, condense, etc.).
	// EXEC RSERV	Calls the service program that displays and/or punches the contents of a relocatable library.
	// EXEC SSERV	Calls the service program that displays and/or punches the contents of a source statement library.
	// EXEC ESERV	Calls the service program that de-edits and displays and/or punches macros from the macro sublibrary.
	// EXEC CORGZ	Calls the organization program that copies all or part of the resident system.
	// EXEC DSERV	Calls the service program that displays the contents of the directories.
	// EXEC PSERV	Calls the service program that displays and/or punches the contents of the procedure library.

Examples of the K Command

The K command is used in conjunction with a display console only.

First Operand	Second Operand	Meaning	Example	Explanation of Example
S	,REF	Display current values of the S-operands	K S,REF*	Assuming that the initialization values are still in effect, K S,DEL=Y,CON=Y, SEG=6 is displayed in the entry area.
S	,DEL=Y	Delete messages automatically	K S,DEL=Y	When the screen is full, all deletable messages are deleted.
S	,DEL=N	Do not delete messages automatically	K S,DEL=N	When the screen is full, use the K command or the cursor to delete messages.
S	,CON=Y	Delete messages after verification	K S,CON=Y	When a deletion command has been entered, you can check the messages before they are deleted.
S	,CON=N	Delete messages immediately	K S,CON=N	When a deletion command has been entered, messages are deleted immediately.
S	,ALM=y	Activate audible alarm	K S,ALM=y	The audible alarm will sound if you enter an incorrect control (K) command, or when the message 'MESSAGE WAITING' is displayed.
S	,ALM=N	De-activate audible alarm	K S,ALM=N	The audible alarm will not warn you if you enter an incorrect control (K) command, or when the message 'MESSAGE WAITING' is displayed.
S	,SEG=n	Delete n lines at a time	K S,SEG=4	When you enter K S,SEG (or just K), lines 1 through 4 are deleted.

* You may also enter K S since REF is the default value of the S operand.

First Operand	Second Operand	Meaning	Example	Explanation of Example
E	,SEG	Delete message lines as specified in S,SEG=n	K E,SEG**	Assuming S,SEG=5 was specified, lines 1 through 5 are deleted.
E	,n	Delete line n	K E,4	Message line 4 is deleted.
E	,n,n	Delete the range of lines from n to n	K E,2,6	Lines 2 through 6 are deleted.
E	,N	Delete the line numbers	K E,N	The message line numbers are deleted from the screen.
D	,N	Display line numbers in all message lines	K D,N	All message lines, including continuation lines, are numbered until a K E command is issued.
D	,N,HOLD	Prevents line numbers from being deleted	K D,N,HOLD	All message lines are numbered. Line numbers are erased only by K E,N command.

** You may also enter K since E and SEG are default values.

Examples of the D Command

Entering redisplay mode

The D command is used in conjunction with a display console only.

Controlling redisplay operation

Terminating redisplay mode

Command	Meaning
D	Enter redisplay mode for all messages
D L	
D L,ALL	
D L,AR	Enter redisplay mode for AR messages only
D L,BG	Enter redisplay mode for BG messages only
D L,Fx	Enter redisplay mode for messages from a specified foreground partition only
D L,ALL	Redisplay all messages
D L,F2	Redisplay messages from F2 only
D L,F4,R	Reset the screen to the most recent F4 messages
D L,B	Change from forward to backward redisplay
D L,F	Change from backward to forward redisplay
D L,F,240	Space forward 240 lines
D L,B,70	Space backward 70 lines
D L,R	Reset the screen to status when redisplay started
D L,170	Space 170 lines forward or backward, depending on the redisplay direction currently in effect
D E	Terminate redisplay mode

Section 5

This section contains a glossary, a bibliography, and the index. The glossary lists and defines some of the most frequently used terms with which you, as an operator, will be confronted. The bibliography lists the titles of IBM manuals to which you may want to refer for more detailed information on specific subjects.

Glossary

IBM is grateful to the American National Standards Institute (ANSI) for permission to reprint its definitions from the American National Standard Vocabulary for Information Processing (Copyright © 1970 by American National Standards Institute, Incorporated), which was prepared by Subcommittee X3.5 on Terminology and Glossary of American National Standards Committee X3.

access method: A technique for moving data between virtual storage and input/output devices.

address: (1) An identification, as represented by a name, label, or number, for a register, location in storage, or any other data source or destination such as the location of a station in a communication network. (2) Loosely, any part of an instruction that specifies the location of an operand for the instruction.

address translation: The process of changing the address of an item of data or an instruction from its virtual address to its real storage address. See also dynamic address translation.

alphameric characters: The alphabetic characters A through Z, the digits 0 through 9, and the characters #, $, and @.

alternate track: One of a number of tracks set aside on a disk pack for use as alternatives to any defective tracks found elsewhere on the disk pack.

application program: A program written by a user that applies to his own work.

assembler language: A source language that includes symbolic machine language statements in which there is a one-to-one correspondence with the instruction formats and data formats of the computer.

auxiliary storage: Data storage other than real storage; for example, storage on magnetic tape or disk. Synonymous with external storage, secondary storage.

batch partition: Partition in which batch processing takes place.

batch processing: Sequential processing of programs submitted to the computer as a batch of jobs, each job being separated from the next by job control statements.

book: A group of source statements written in any of the languages supported by DOS/VS and stored in a source statement library.

byte: A sequence of eight adjacent binary digits that are operated upon as a unit and that constitute the smallest addressable unit of the system.

card punch: A device to record information in cards by punching holes in the cards to represent letters, digits, and special characters.

card reader: A device which senses and translates into machine code the holes in punched cards.

cardless system: A System/370 configured without a card reader or card punch, but with an IBM 3540 Diskette Input/Output Unit.

catalog: To enter a phase, module, book, or procedure into one of the system or private libraries.

central processing unit: A unit of a computer that includes the circuits controlling the interpretation and execution of instructions. Abbreviated CPU.

channel: (1) A path along which signals can be sent, for example, data channel, output channel. (2) A hardware device that connects the CPU and real storage with the I/O control units.

compile: To prepare a machine language program from a computer program written in a high-level language by making use of the overall logic structure of the program, or generating more than one machine instruction for each symbolic statement, or both, as well as performing the function of an assembler.

compiler: A program that translates high-level language statements into machine language instructions.

configuration: The group of machines, devices, etc., which make up a data processing system.

control program: A program that is designed to schedule and supervise the performance of data processing work by a computing system.

control unit: A device that controls the reading, writing, or display of data at one or more input/output devices.

core image library: A library of phases that have been produced as output from link-editing. The phases in the core image library are in a format that is executable either directly or after processing by the relocating loader in the supervisor.

data file: A collection of related data records organized in a specific manner. For example, a payroll file (one record for each employee, showing his rate of pay, deductions, etc., or an inventory item, showing the cost, selling price, number in stock, etc.). See also file.

default value: The choice among exclusive alternatives made by the system when no explicit choice is specified by the user.

diagnostic routine: A program that facilitates computer maintenance by detection and isolation of malfunctions or mistakes.

direct access: (1) Retrieval or storage of data by a reference to its location on a volume, other than relative to the previously retrieved or stored data. (2) Pertaining to the process of obtaining data from, or placing data into, storage where the time required for such access is independent of the location of the data most recently obtained or placed in storage. (3) Pertaining to a storage device in which the access time is effectively independent of the location of the data. Synonymous with random access.

directory: An index that is used by the system control and service programs to locate one or more sequential blocks of program information that are stored on direct access storage.

disk pack: A direct access storage volume containing magnetic disks on which data is stored. Disk packs are mounted on a disk storage drive, such as the IBM 3330 Disk Storage Drive.

diskette: A flexible magnetic oxide coated disk suitable for data storage and retrieval.

dump: (1) To copy the contents of all or part of virtual storage. (2) The data resulting from the process as in (1).

dynamic address translation (DAT): (1) The change of a virtual storage address to an address in real storage during execution of an instruction. (2) A hardware function that performs the translation.

error message: The communication that an error has been detected.

error recovery procedures: Procedures designed to help isolate, and, when possible, to recover from errors in equipment. The procedures are often used in conjunction with programs that record the statistics of machine malfunctions.

extent: A continuous space on a direct-access storage device, occupied by or reserved for a particular file.

file: A collection of related records treated as a unit. For example, one line of an invoice may form an item, a complete invoice may form a record, the complete set of such records may form a file, the collection of inventory control files may form a library, and the libraries used by an organization are known as its data bank.

hard copy: A printed copy of machine output in a visually readable form, for example, printed reports, listings, documents, and summaries.

hard wait state: In general, a wait state is the condition of a CPU when all operations are suspended. System recovery from a hard wait state requires that the user performs a new IPL (initial program load) procedure.

hardware: Physical equipment, as opposed to the computer program or method of use, for example, mechanical, magnetic, electrical, or electronic devices. Contrast with software.

idle time: That part of available time during which the hardware is not being used.

indexed-sequential organization: The records of an indexed sequential file are arranged in logical sequence by key. Indexes to these keys permit direct access to individual records. All or part of the file can be processed sequentially.

Initial Program Load (IPL): The initialization procedure that causes a DOS/VS to commence operation.

I/O: Abbreviation for input/output.

job: (1) A specified group of tasks prescribed as a unit of work for a computer. By extension, a job usually includes all necessary computer programs, linkages, files, and instructions to the operating system. (2) A collection of related problem programs, identified in the input stream by a JOB statement followed by one or more EXEC statements.

job accounting interface: A function that accumulates, for each job step, accounting information that can be used for charging usage of the system, planning new applications, and supervising system operation more efficiently.

job control: A program that is called into a virtual partition to prepare each job or job step to be run. Some of its functions are to assign I/O devices to certain symbolic names, set switches for program use, log (or print) job control statements, and fetch the first program phase of each job step.

job step: The execution of a single processing program.

label: identification record for a tape or disk file.

label information cylinder: Under DOS/VS, one cylinder (two cylinders for the 3340) of the system residence pack that stores information read from job control statements or commands. Synonymous with label cylinder.

language translator: A general term for any assembler, compiler, or other routine that accepts statements in one language and produces equivalent statements in another language.

librarian: The set of programs that maintains, services, and organizes the system and private libraries.

library: A collection of files or programs, each element of which has a unique name, that are related by some common characteristics. For example, all phases in the core image library have been processed by the linkage editor.

linkage editor: A processing program that prepares the output of language translators for execution. It combines separately produced object modules; resolves symbolic cross references among them, and generates overlay structures on request; and produces executable code (a phase) that is ready to be fetched or loaded into virtual storage.

load: (1) In programming, to enter instructions or data into storage or working registers. (2) In DOS/VS, to bring a program phase from a core image library into virtual storage for execution.

multiprogramming system: A system that controls more than one program simultaneously by interleaving their execution.

object code: Output from a compiler or assembler which is suitable for processing by the linkage editor to produce executable machine code.

object module: A module that is the output of an assembler or compiler and is input to a linkage editor.

object program: A fully compiled or assembled program. Contrast with source program.

operand: (1) That which is operated upon. An operand is usually identified by an address part of an instruction. (2) Information entered with a command name to define the data on which a command processor operates and to control the execution of the command processor.

operator command: A statement to the control program, issued via a console device, which causes the control program to provide requested information, alter normal operations, initiate new operations, or terminate existing operations.

operator message: A message from the operating system or a problem program directing the operator to perform a specific function, such as mounting a tape reel, or informing him of specific conditions within the system, such as an error condition.

page: (1) In DOS/VS, a 2K block of instructions, data or both. (2) To transfer instructions, data, or both between real storage and the page data set.

page data set: An extent in auxiliary storage, in which pages are stored.

page fault: A program check interruption that occurs when a page that is marked *not in real storage* is referred to by an active page. Synonymous with page translation exception.

page frame: A 2K block of real storage that can contain a page.

page in: The process of transferring a page from the page data set to real storage.

page out: The process of transferring a page from real storage to the page data set.

page pool: The set of all page frames that may contain pages of programs in virtual mode.

paging: The process of transferring pages between real storage and the page data set.

parameter: A variable that is given a constant value for a specific purpose or process.

peripheral equipment: A term used to refer to card devices, magnetic tape and disk devices, printers, and other equipment bearing a similar relation to the CPU.

phase: The smallest complete unit that can be referred to in the core image library.

printer: A device that expresses coded characters as hard copy.

private library: A user-owned library that is separate and distinct from the system library.

problem determination aid (PDAID): A program that traces a specified event when it occurs during the operation of a program.

problem program: Any program that is executed when the central processing unit is in the problem state; that is, any program that does not contain privileged instructions. This includes IBM-distributed programs, such as language translators and service programs, as well as programs written by a user.

processing program: (1) A general term for any program that is not a control program. (2) Synonymous with problem program.

processor storage: The general purpose storage of a computer. Processor storage can be accessed directly by the operating registers. Synonymous with real storage.

queue: (1) A waiting line or list formed by items in a system waiting for service; for example, tasks to be performed or messages to be transmitted in message switching systems. (2) To arrange in, or form, a queue.

random processing: The treatment of data without respect to its location in auxiliary storage, and in an arbitrary sequence governed by the input against which it is to be processed.

real address: The address of a location in real storage.

real address area: In DOS/VS, the area of virtual storage where virtual addresses are equal to real addresses.

real mode: In DOS/VS, the mode of a program that cannot be paged.

real partition: In DOS/VS, a division of the real address area of virtual storage that may be allocated for programs that are not to be paged, or for those pages of virtual programs which are to remain in storage during program execution.

real storage: The storage of a System/370 computing system from which the central processing unit can directly obtain instructions and data, and to which it can directly return results. Synonymous with processor storage.

relocatable library: A library of relocatable object modules required by various compilers. It allows the user to keep frequently used modules available for combination with other modules without recompilation.

routine: An ordered set of instructions that may have some general or frequent use.

sequential file: A file whose records are organized on the basis of their succesive physical positions, contrast with direct access.

service program: A program that assists in the use of a computing system, without contributing directly to the control of the system or the production of results.

shared virtual area: An area located in the highest addresses of virtual storage. It can contain a system directory list of highly used phases and resident programs that can be shared between partitions.

software: A set of programs, concerned with the operation of the hardware in a data processing system.

source program: A computer program written in a source language. Contrast with object program.

source statement library: A collection of books (such as macro definitions) cataloged in the system by the librarian program.

spooling: The reading and writing of input and output streams on auxiliary storage devices, concurrently with job execution, in a format convenient for later processing or output operations. Synonymous with concurrent peripheral operation.

stand-alone dump: A program that displays the contents of the registers and part of the real address area and that runs independently and is not controlled by DOS/VS.

standard label: A fixed-format identification record for a tape or disk file. Standard labels can be written and processed by DOS/VS.

storage protection: An arrangement for preventing unauthorized access to storage.

supervisor: A component of the control program. It consists of routines to control the functions of program loading, machine interruptions, external interruptions, operator communications and physical IOCS requests and interruptions. The supervisor alone operates in the privileged (supervisor) state. It coexists in real storage with problem programs.

switched line: A communication line in which the connection between the computer and a remote station is established by dialing. Synonymous with dial line.

system directory list: A list containing directory entries of highly used phases and of all phases resident in the shared virtual area. This list is placed in the shared virtual area.

system residence device: The direct access device on which the system residence pack is located.

system residence pack: The disk pack on which the basic system and all related supervisor code is located.

teleprocessing: The processing of data that is received from or sent to remote locations by way of telecommunication lines.

teleprocessing balancing: A supervisor function that allows teleprocessing users who have concurrent batch processing in a paging environment to obtain better teleprocessing response times at the expense of slower job execution in the batch partition(s).

terminal: (1) A point in a system or communication network at which data can either enter or leave. (2) Any device capable of sending and receiving information over a communication channel.

throughput: The total volume of work performed by a computing system over a given period of time.

track: The portion of a moving storage medium, such as a drum, tape, or disk, that is accessible to a given reading head position.

transient area: An area of real storage used for temporary storage of transient routines.

user label: An identification record for a tape or disk file; the format and contents are defined by the user, who must also write the necessary processing routines.

utility program: A problem program designed to perform a routine task, such as transcribing data from one storage device to another.

virtual address: An address that refers to virtual storage and must, therefore, be translated into a real storage address when it is used.

virtual address area: In DOS/VS, the area of virtual storage whose addresses are greater than the highest address of the real address area.

virtual mode: In DOS/VS, the mode of execution of a program which may be paged.

virtual partition: In DOS/VS, a division of the virtual address area of virtual storage that is allocated for programs that may be paged.

virtual storage: Addressable space that appears to the user as real storage, from which instructions and data are mapped into real storage locations. The size of virtual storage is limited by the addressing scheme of the computing system and by the capacity of the page data set, rather than by the actual number of real storage locations.

Virtual Storage Access Method (VSAM): VSAM is an access method for direct or sequential processing of fixed and variable-length records on direct access devices. The records in a VSAM file can be organized either in logical sequence by a key field (key sequence) or in the physical sequence in which they are written on the file (entry sequence). A key-sequenced file has an index, an entry-sequenced file does not.

Virtual Telecommunications Access Method (VTAM): VTAM is an access method that supports communication between application programs and terminals in a telecommunications network.

volume: That portion of a single unit of storage media which is accessible to a single read/write mechanism, for example, a drum, a disk pack, or part of a disk storage module. (2) A recording medium that is mounted and dismounted as a unit, for example, a reel of magnetic tape, a disk pack, a data cell.

volume table of contents: A table on a direct access volume that describes each file on the volume. Abbreviated VTOC.

VSAM access method services: A multifunction utility program that defines VSAM files and allocates space for them, converts indexed sequential files to key-sequenced files with indexes, facilitates data portability between operating systems, creates backup copies of files and indexes, helps to make inaccessible files accessible, and lists file and catalog entries.

VSAM catalog: A key-sequenced file, with an index, containing extensive file and volume information that VSAM requires to locate files, to allocate and deallocate storage space, to verify the authorization of a program or operator to gain access to a file, and to accumulate usage statistics for files.

Bibliography

Introduction to DOS/VS, GC33–5370

DOS/VS System Control Statements, GC33–5376

DOS/VS Messages, GC33–5379

DOS/VS POWER/VS Installation Guide and Reference, SH12–5430

DOS/VS POWER/VS Workstation User's Guide, SH12–5524

VTAM Network Operating Procedures, GC27–0025

DOS/VS System Management Guide, GC33–5371

DOS/VS Serviceability Aids and Debugging Procedures, GC33–5380, abbreviated in this manual as DOS/VS SADP

TOLTEP for VTAM, GC28–0663

System/370 Model 125 Procedures, GA33–1509

System/370 Model 135 Procedures, GC38–0005

System/370 Model 145 Procedures, GC38–0015

IBM 3340 Fixed Head Feature User's Guide, GA26–1632

Index

Index

Index

Operator's Library
DOS/VS Operating Procedures

GC33-5378-4

READER'S COMMENT FORM

This sheet is for comments and suggestions about this manual. We would appreciate *your* views, favorable or unfavorable, in order to aid us in improving *this* publication. This form will be sent directly to the author's department. Please include your name and address if you wish a reply. Contact your IBM branch office for answers to technical questions about the system or when requesting additional publications. Thank you.

Name

Address

What is your occupation?

How did you use this manual?

As a reference source

As a classroom text

As a self-study text

Your comments* and suggestions:

*** We would especially appreciate your comments on any of the following topics:**

Clarity of the text
Organization of the text
Accuracy
Cross-references
Index
Tables
Illustrations
Examples
Appearance
Printing
Paper
Binding

GC33-5378-4

YOUR COMMENTS, PLEASE . . .

This manual is part of a library that serves as a reference source for systems analysts, programmers and operators of IBM systems. Your answers to the questions on the back of this form, together with your comments, will help us produce better publications for your use. Each reply will be carefully reviewed by the persons responsible for writing and publishing this material. All comments and suggestions become the property of IBM.

Please note: Requests for copies of publications and for assistance in utilizing your IBM system should be directed to your IBM representative or to the IBM sales office serving your locality.

CUT ALONG THIS LINE

Fold

Fold

FIRST CLASS
PERMIT NO. 1359
WHITE PLAINS, N. Y.

BUSINESS REPLY MAIL

NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

POSTAGE WILL BE PAID BY . . .

IBM Corporation
1133 Westchester Avenue
White Plains, N.Y. 10604

Attention: Department 813 BP

Fold

Fold

IBM®

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
(U.S.A. only)

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
(International)

GC33-5378-4

Operator's Library DOS/VS Operating Procedures Printed in U.S.A. GC33-5378-4

IBM

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
(U.S.A. only)

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
(International)