GC33-5378-1
File No. S370-40

Systems

Operator's Library DOS/VS Operating Procedures

Release 29

IBM

GC33-5378-1
File No. S370-40

Systems

Operator's Library
DOS/VS
Operating Procedures

Release 29

IBM

Summary of ammendments

This edition documents

- Virtual Storage enhancements and
- Support of the following new devices:

System/370 Model 115
3203 and 5203 Printers
3340 Disk Storage
3540 Diskette I/O Unit
3780 Data Communication Terminal
5425 Multifunction Card Unit

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

Changes in content are indicated by a vertical bar to the left of the change.

Second Edition (November 1973)

This edition applies to Version 5, Release 29, of the IBM Disk Operating System: 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/360 and System/370 Bibliography, GA22-6822*, for the editions that are applicable and current.

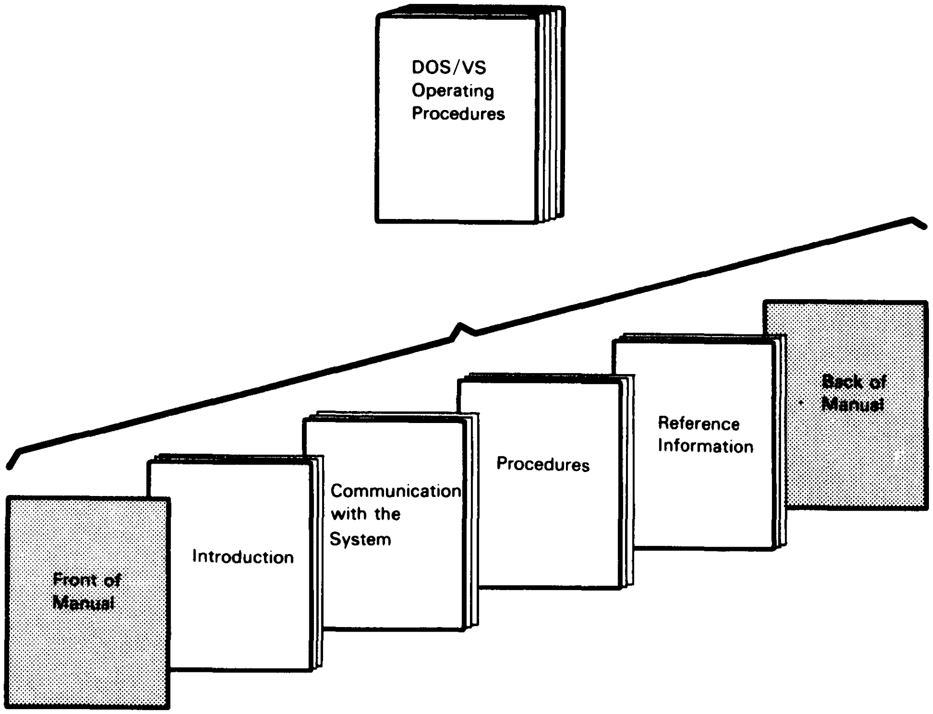
Requests for copies of IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

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 Laboratories, Programming Publications Department, Schoenaicher Strasse, 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.

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

The section *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

The section *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

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

Reference Information

The section *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.

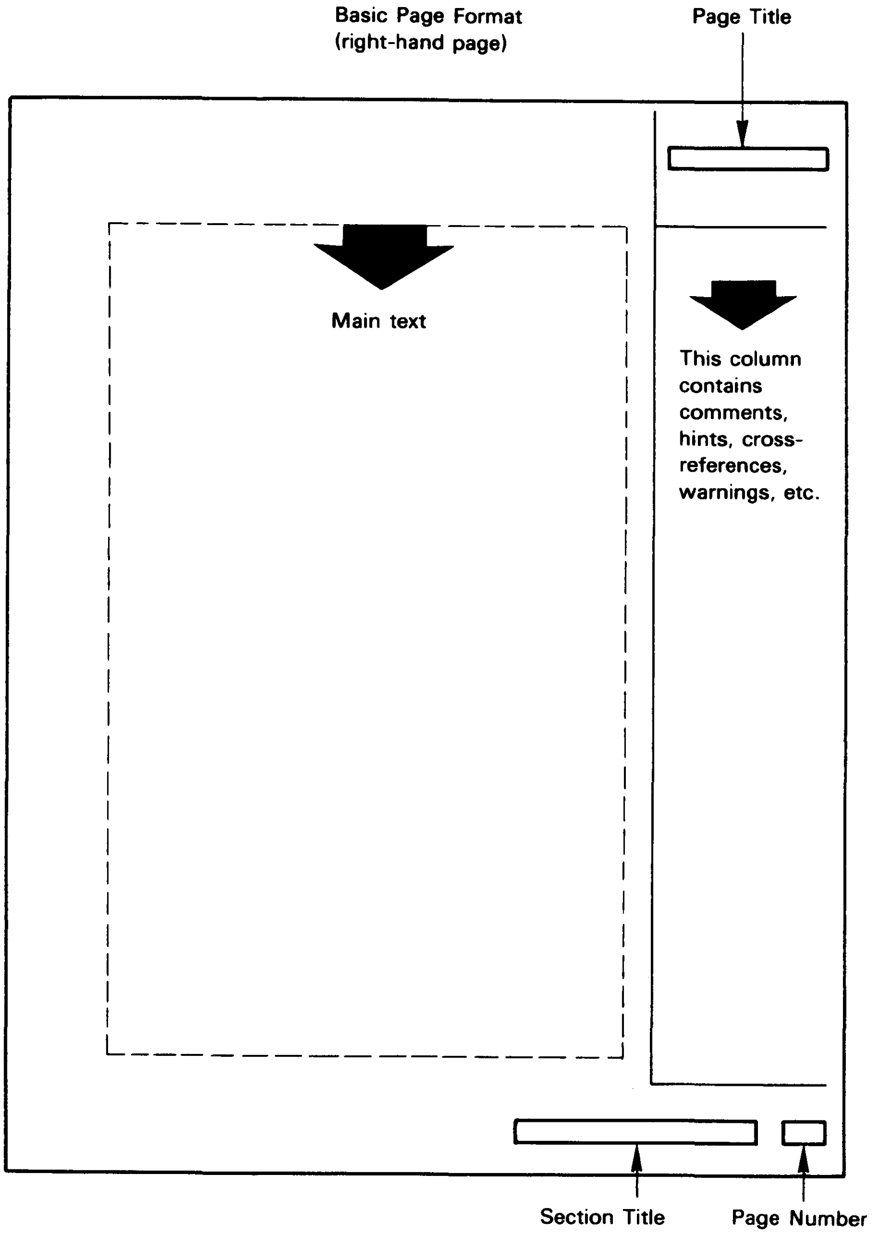
The bibliography contains a list of DOS/VS manuals, which enable you to obtain detailed information concerning individual topics.

Contents

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	priority output writers, execution processors, and input readers
RDE	reliability data extractor
RJE	remote job entry
RMSR	recovery management support recording
SDL	system directory list
SVA	shared virtual area
TOD	time of day
UCB	universal character set buffer
VSAM	virtual storage 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 an operator, are responsible for the operations that are necessary to enable your data-processing installation to perform these functions.

Preparing the system

Before the system can start performing any of these 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.

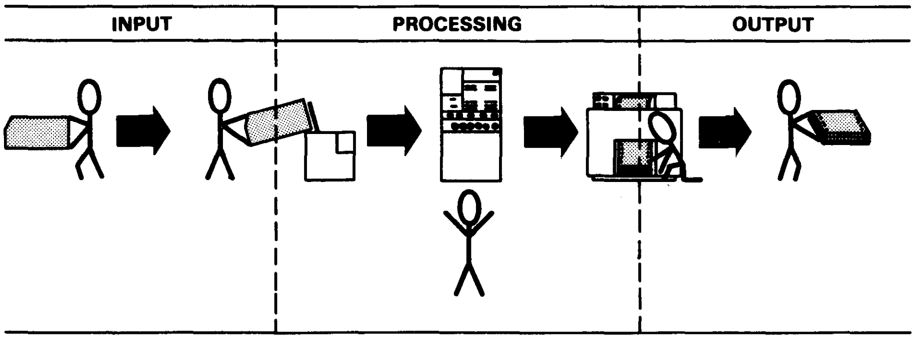
The input usually consists of programs, data, and control information and is normally on punched cards, on tape, on disk, or on diskette. In most cases the input is supplied by the programmer. Sometimes, however, you may have to correct errors in control statements or prepare such statements yourself. 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, you may have to mount tape reels or disk packs. You may also have to respond to messages that are printed on the printer keyboard or, if you operate a Model 115 or 125, appear on the display 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

Information previously in a procedure

When a programmer writes a program that is expected to run for a long time, he can make provisions for writing checkpoints on disk or tape at regular intervals. These checkpoints contain the status of the job and system at the time they were written. If the program ends abnormally there is no need to rerun it from the beginning; it may be restarted from the last checkpoint.

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.*

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.

The Operator's Responsibilities (continued)

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 (CPU, input/output devices, programs, etc.) was fairly easy. However, as installations grew in size and complexity, the need to simplify resource management became apparent. 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 and 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.

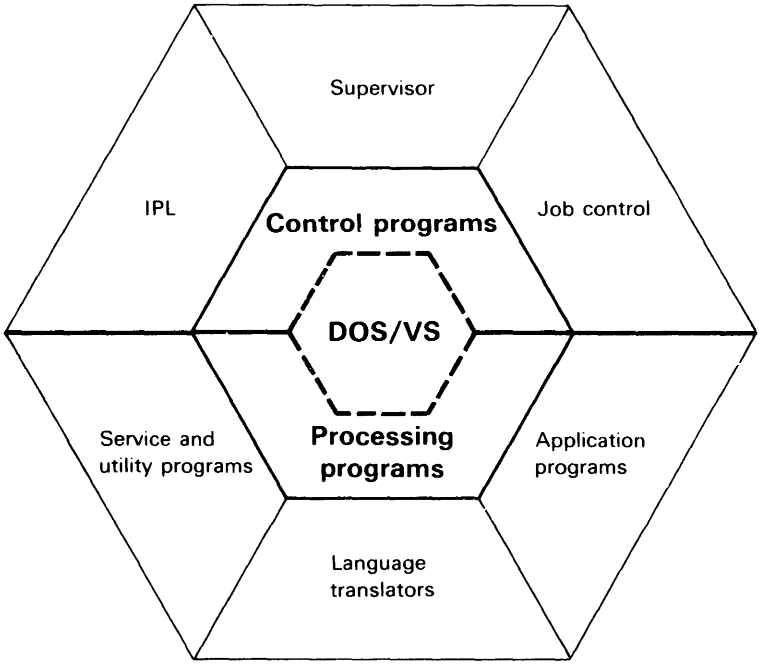


Figure I-1. DOS/VS consists of control programs and processing programs.

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.

Job control statements/commands

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 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

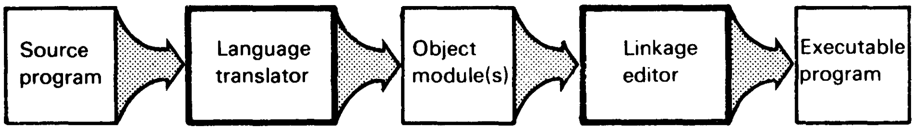
The DOS/VS service programs consist of (1) the linkage editor program and (2) a set of programs called the librarian.

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

Linkage Editor

Executable program
Phase

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

Information previously in a procedure

Utility Programs

A number of IBM system utility programs are available, which are designed to perform everyday tasks. These programs are:

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.

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.

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.

Fast Copy Disk Volume: This program copies the complete contents of an IBM 3336 Disk Pack or 3348 Data Module on to another IBM 3336 Disk Pack or 3348 Data Module, respectively. Copying may be performed either directly from disk to disk, or it may be accomplished in two steps, using magnetic tape as intermediate output. There are two versions of the program: "Integrated", which runs under the DOS/VS supervisor like the other utilities, and "stand-alone",which is contained in cards and executed independently of the DOS/VS system.

Initialize Data Cell: This program prepares from one to five 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 from one to eight IBM standard tape volume labels in numerical sequence, followed by one dummy header label and one tapemark on EBCDIC tapes.

Printlog: This program produces a printed copy of messages contained in the hard-copy file of the IBM System/370 Models 115 and 125.

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 21 and 26 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

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.

Books

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.

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.

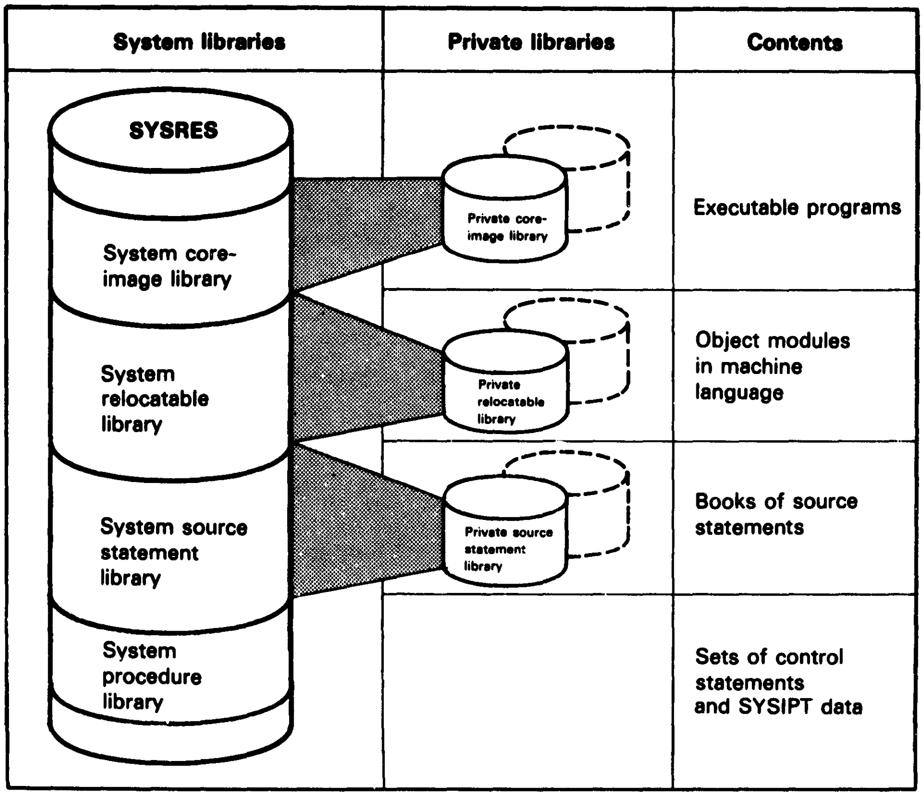
All system libraries together form the DOS/VS system. They all reside on the same disk pack, which is referred to as the system residence pack or SYSRES (see Figure I-2).

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.

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.

A private library can be contained on any disk pack of the same device type as the corresponding system library. Private libraries are, in fact, 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.

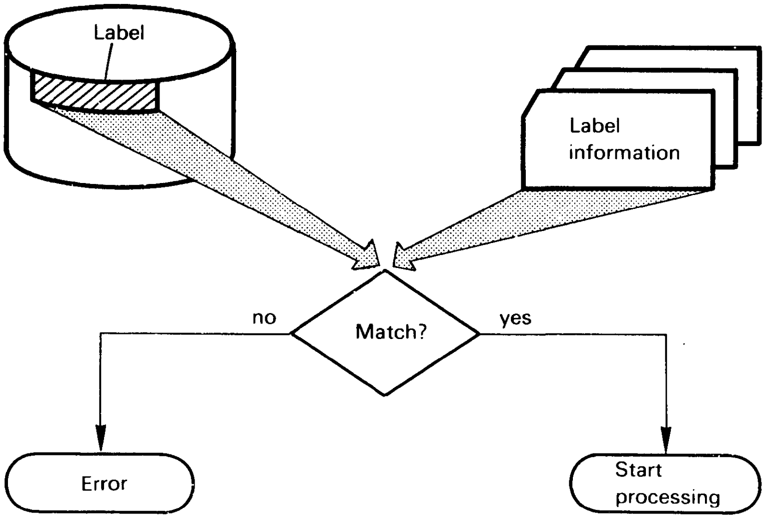


Figure I-3. **Label checking.**

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, etc.) 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).

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, etc.

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.

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.

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, DOC*, or printer	operator messages and logging job control statements. It can also be assigned to a printer if used in a single-partition environment.
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
SYSUSE	disk extent	system-internal purposes
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 125, messages to the operator are also stored on SYSREC for subsequent printing.
SYSVIS	disk extent	virtual-storage page data set
SYSCAT	disk extent	VSAM catalog

* Display Operator Console

Figure I-4. System logical units and their use.

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 job control cards (punched cards containing job control statements or commands), or you, the operator, can enter them directly from your console printer keyboard or DOC.

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.

LUB and PUB tables

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.

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.

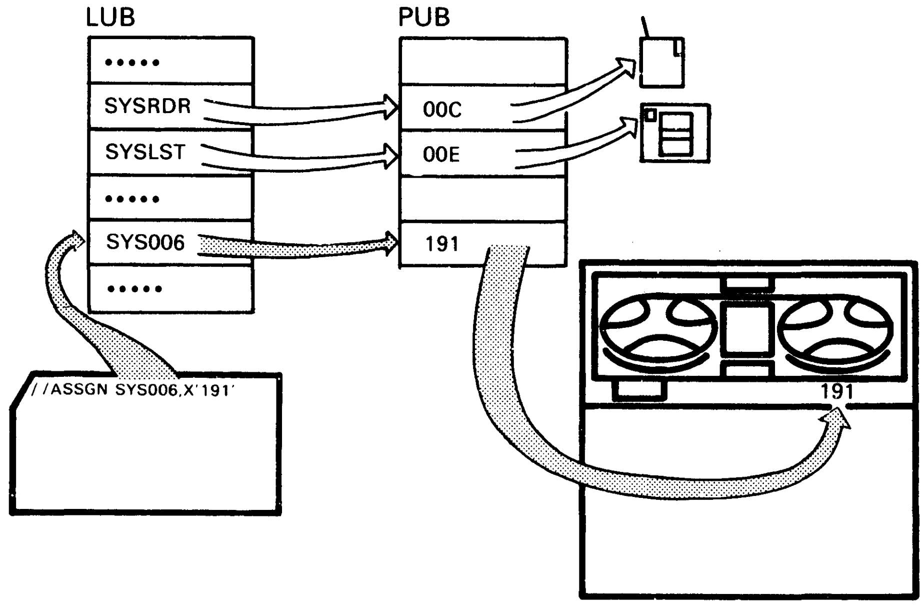


Figure I-5. **The ASSGN statement establishes a link between SYS006 and tape drive 191.**

Jobs and Job Steps

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

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 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.

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.

Job stream

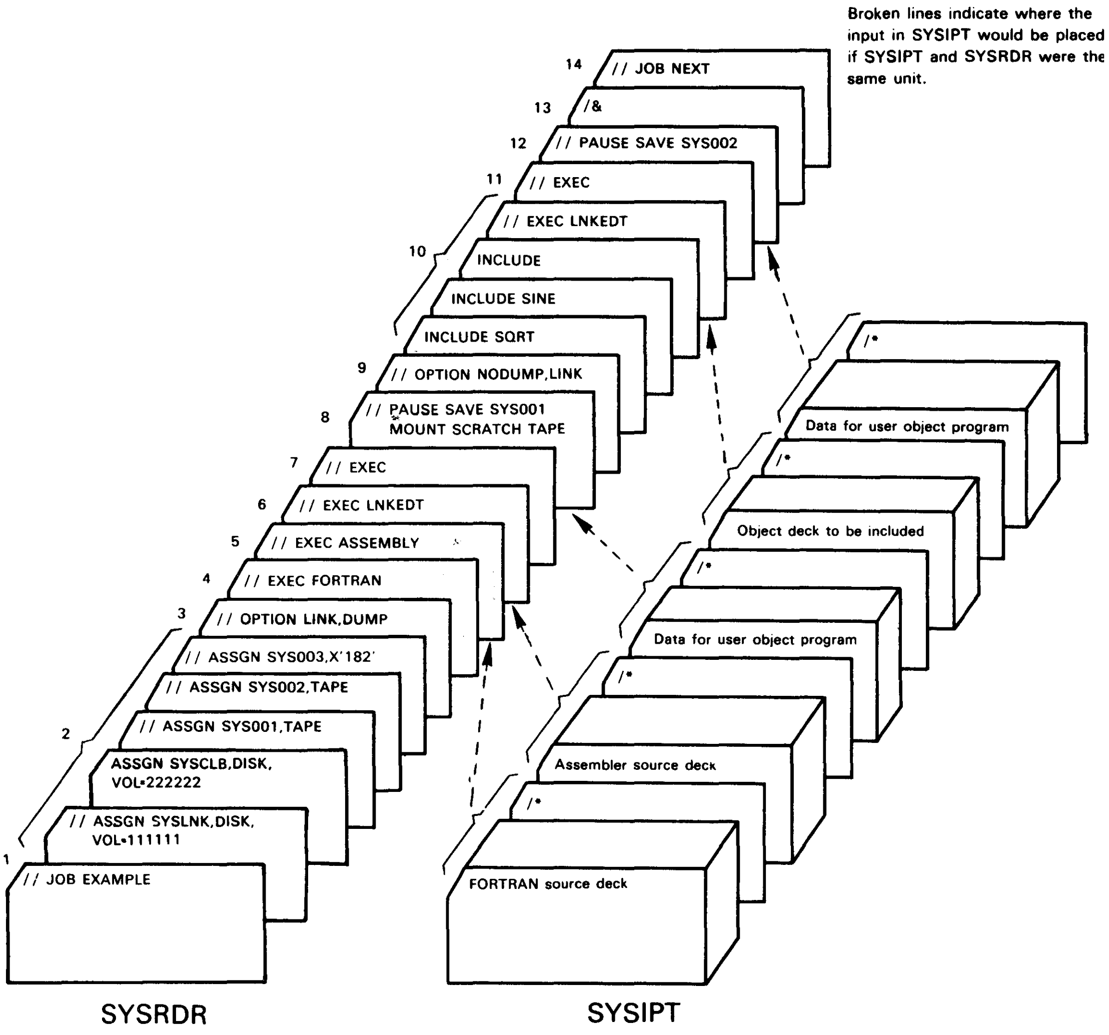
You can interrupt the processing of a job stream to make last-minute changes to one of the jobs, or to squeeze in a rush job. You do this by means of the PAUSE command, which you enter from the console printer-keyboard or the DOC, and which causes processing to halt at the end of the current job or job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

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 must be marked by a statement containing /* in positions 1 and 2.

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

A detailed example of a job is shown in Figure I-6. The numbers on the left-hand side of the statements and commands refer to the explanatory 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.*



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.

Multiprogramming

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.

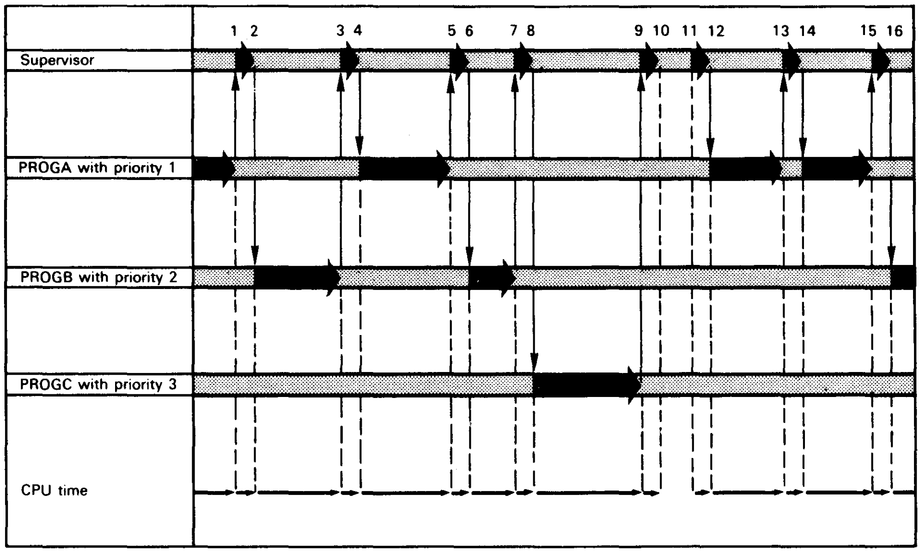


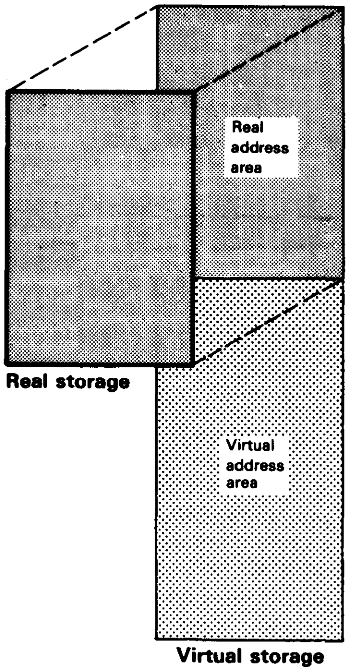
Figure 1-7. CPU-usage in a multiprogramming system.

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).



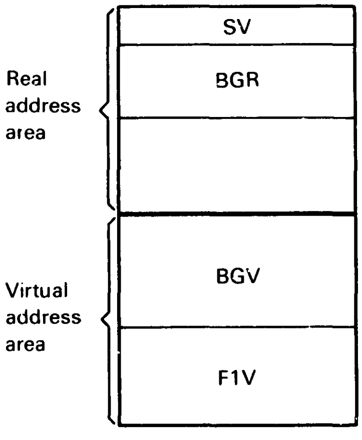
Real address area

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 maximum of five partitions: BG, F4, F3, F2, and F1. 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 OK.

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 looks as follows:



Virtual storage

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

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

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

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 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.

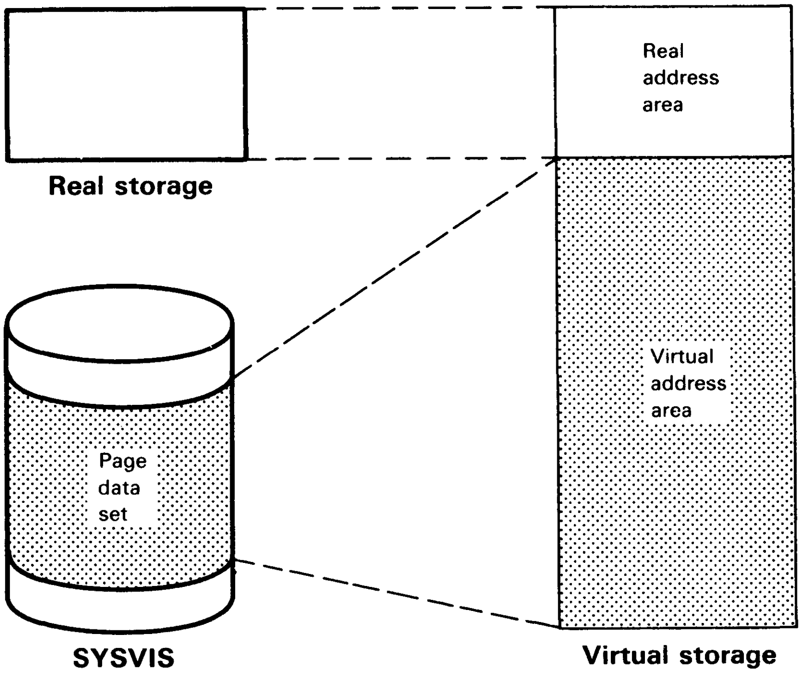


Figure I-8. The page data set is an image of virtual storage.

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.

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.

Pages

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

Page frames

Page pool

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.

Paging in

Paging out

Figure I-9 shows how the pages of Program A are paged in and out between the page data set and virtual 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.

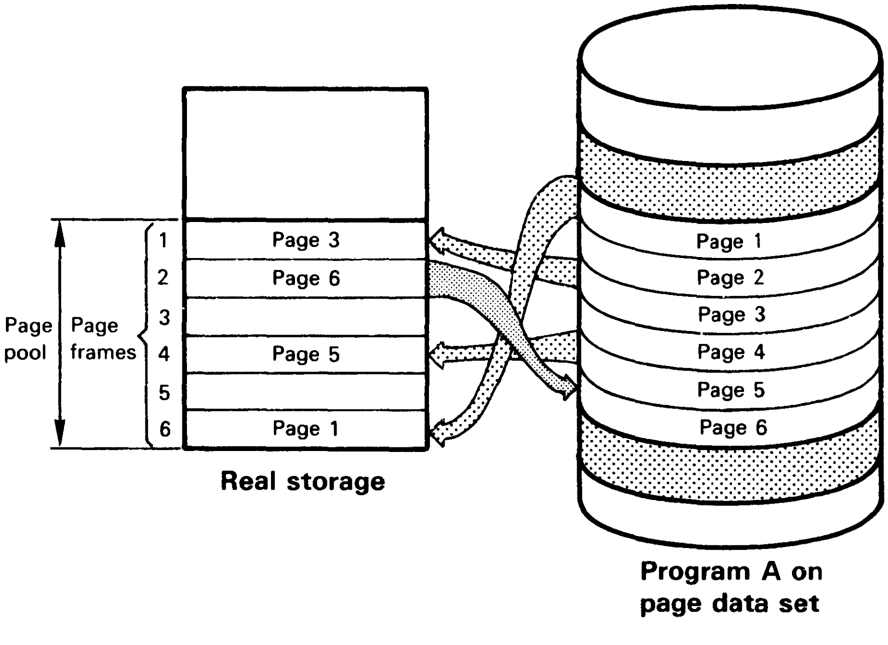


Figure I-9. Paging between the page data set and real storage.

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 belongs 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

POWER is a service program designed to increase the productivity of computer installations operating under DOS/VS. POWER runs in a real partition. It can service up to four other partitions (a maximum multiprogramming system) for virtual-mode as well as real-mode program execution.

POWER's own partition must have a higher priority than any of the partitions it supports.

Job entries

The POWER program reads all SYSIN card input for the partitions operating under control of POWER, and stores it on disk. This card input on disk represents job stream input and is grouped into job entries, which are units of work upon which POWER can act. Job entries are all the cards contained between the // JOB and the /& cards if the job-entry control language (JECL) is not used; if JECL is used, a job entry consists of all the cards contained between the * $$ JOB card and the * $$ EOJ card.

JECL

The POWER program transfers the input from disk to the individual partitions as and when required. Conversely, POWER intercepts all printer and punch output directed to the SYSLST and SYSPCH devices, and again stores it on disk. The actual printing and punching is done later. This procedure is shown in Figure I-10. POWER also provides a writer-only option, which controls output operations only.

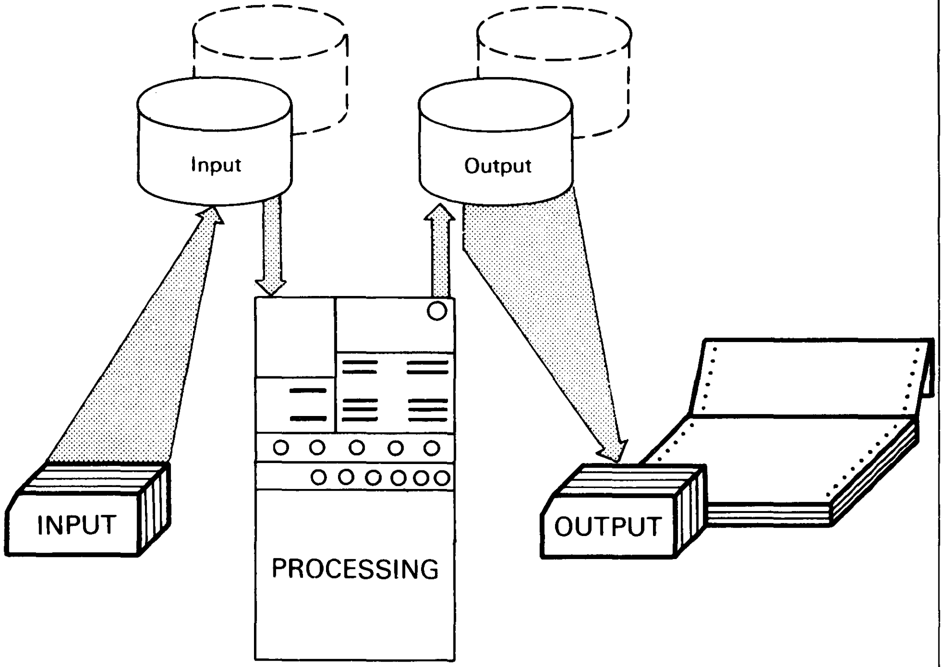


Figure I-10. POWER stores input and output temporarily on disk.

POWER increases system productivity in two ways:

Unit-record devices

1. During processing, jobs in the POWER-supported partitions are not dependent on the slow unit-record I/O devices (card readers, punches, printers). Instead, they obtain their input at disk 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. Optimal use is made of the system's unit record devices. In a multiprogramming system without POWER, each partition requires its own card reader, 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, one card reader assigned to the POWER partition can handle the input for all sup-

ported partitions, and one printer and one punch - also assigned to the POWER partition - are sufficient to produce all output.

You, the operator, may start and stop input reading and output printing and punching independently of execution. Whenever a card reader, a punch, or a printer 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.

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, if you operate an IBM System/370 Model 115 or 125, the Display Operator Console (DOC), 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.*

System-to-Operator Communication

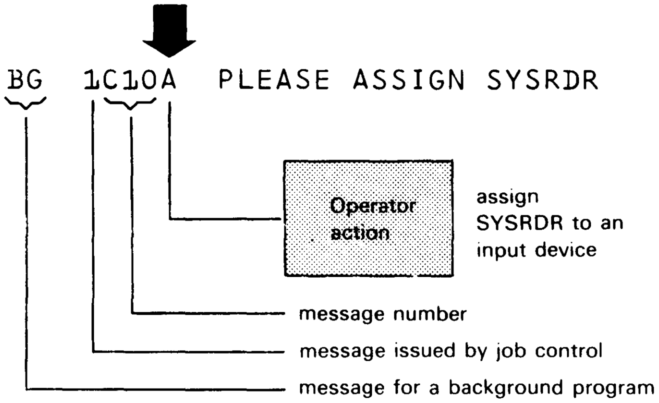
The system communicates with you by issuing messages on SYSLOG, the logical unit to which the console printer-keyboard or the DOC is assigned. Each message is preceded by a partition identifier and a message code. There are four types of message:

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.



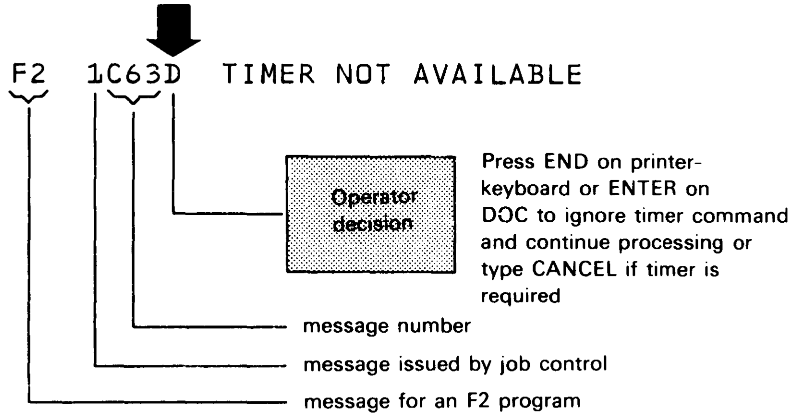
```
BG  1C10A  PLEASE ASSIGN SYSRDR
```

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

Decision Messages

D

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



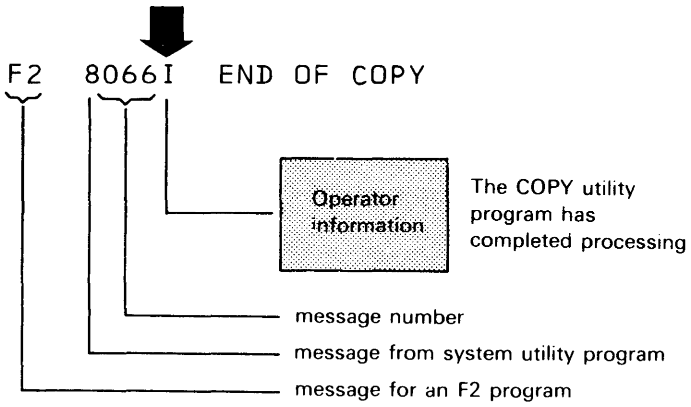
```
F2  1C63D  TIMER NOT AVAILABLE
```

Operator decision
Press END on printer-keyboard or ENTER on DOC to ignore timer command and continue processing or type CANCEL if timer is required
message number
message issued by job control
message for an F2 program

I

Information Messages

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



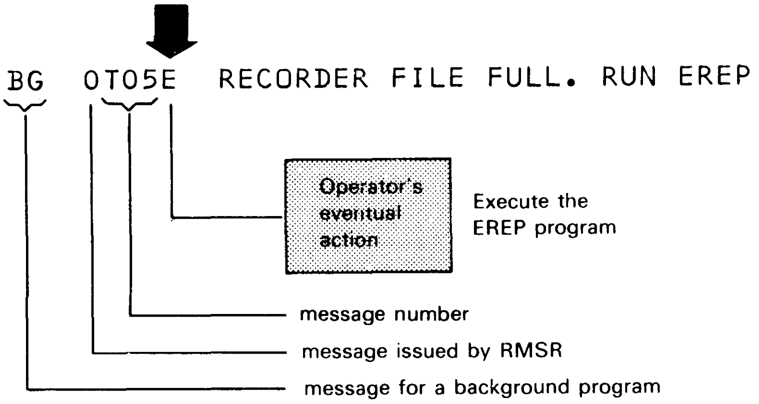
```
F2  8066I  END OF COPY
```

Operator information
The COPY utility program has completed processing
message number
message from system utility program
message for an F2 program

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
```

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-System 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 are normally punched in cards and are entered via a card reader (SYSRDR); they are accepted by the system only between jobs or job steps.

Operator commands

Operator commands are usually entered from the console printer-keyboard or, if you operate an IBM System/370 Model 115 or 125, from the DOC, which are both assigned to SYSLOG. If entered from the console printer keyboard, commands must be followed by pressing END; if entered from the DOC, they must be followed by pressing ENTER.

There are three types of operator command:

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 and the message READY FOR COMMUNICATIONS has appeared. Attention commands from the DOC may be entered without pressing REQUEST, unless the message ENTER RESPONSE is displayed. In this case, press REQUEST first.

Job control commands

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

All DOS/VS job control statements and operator commands are described in detail in the section *Reference Information.*

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 (mount the next tape reel or disk pack, etc.).

- **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, etc.
- **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.

  START and BATCH are identical

  The START command enables you to resume processing after a STOP command. By means of the TIMER command you can give interval-time support to a specified partition.

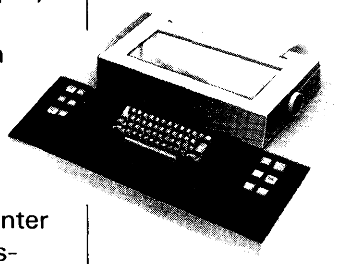
Although the normal communication device for operator commands is SYSLOG (assigned to a console printer-keyboard or to the DOC), commands may also be entered from the card reader (assigned to SYSRDR), with the exception of attention commands.

Devices Used for Operator/System Communication

This chapter deals with two types of device 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).

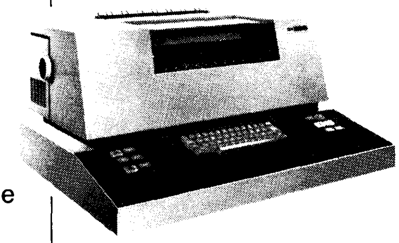


IBM 3210 Console Printer-Keyboard

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.

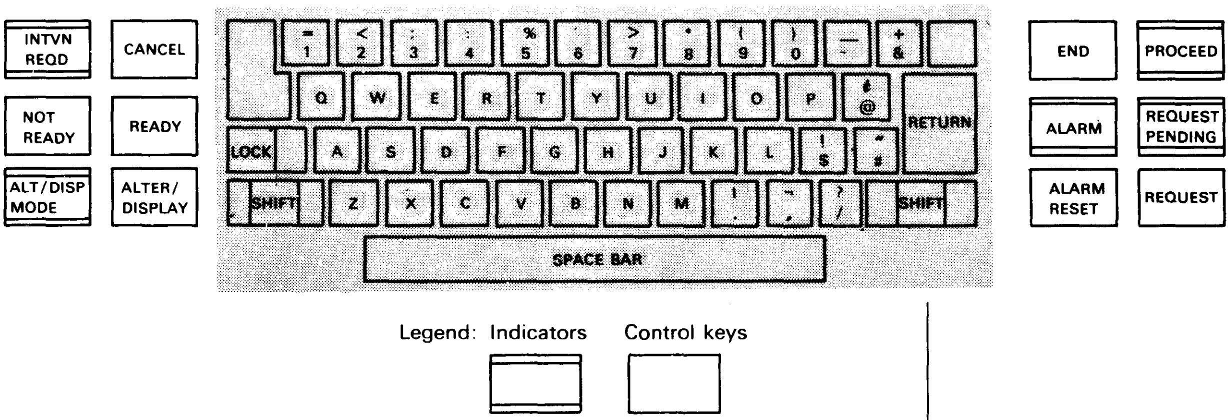


IBM 3215 Console Printer-Keyboard

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.

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

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
PROCEEC	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.

The Display Operator Console (DOC)

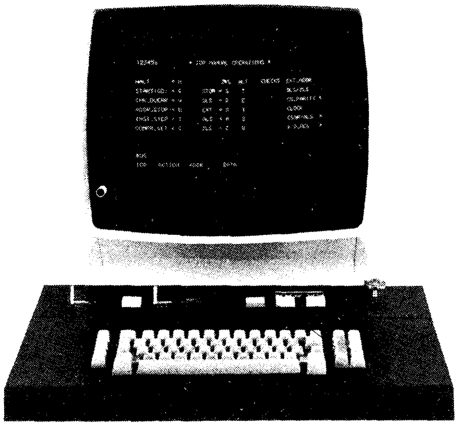
The DOC is an integral part of the IBM System/370 Models 115 and 125 and serves two purposes:

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:

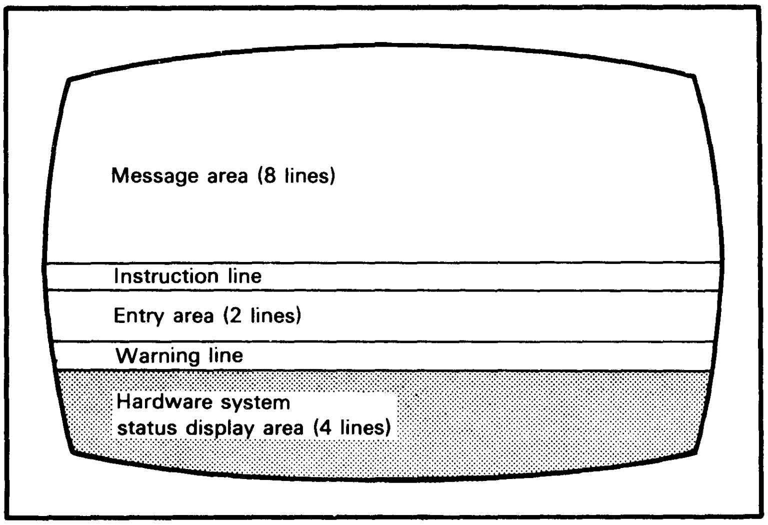
- input can be entered immediately; there is no need to wait for the message READY FOR COMMUNICATION,
- messages are delivered faster, and
- errors in statements and commands can be corrected more easily.

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.

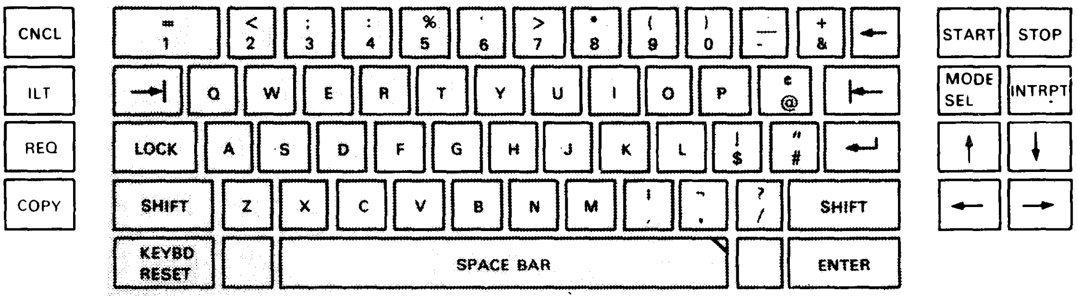
Blank	Line number	Blank	Partition indicator	Message text	
1	2	3	4	6	56

Figure II-3. Format of the message lines.

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

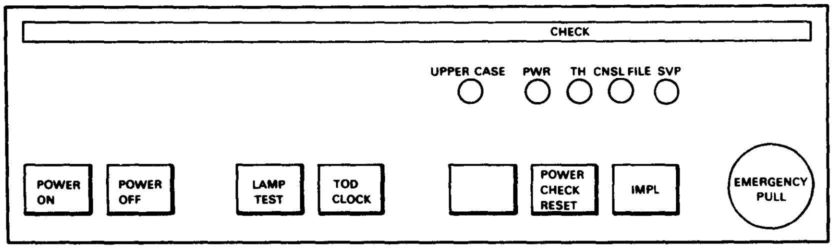


The functions of the keys are described in detail in *Operator's Library System/370:*

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.

DOC Screen Operation

DOC screen operation consists of deleting and redisplaying messages on the screen. Since there are only eight lines available for messages, the message area will soon be full and therefore 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 can print them on SYSLST. 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 eight lines of the message area on the screen are 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:

```
K     ┌ S              ┐
      │ E  [,options]  │
      └ D              ┘
```

For syntax notation refer to Reference Information

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:

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

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

K S,DEL=Y

Automatic Deletion Mode
In automatic deletion mode, the deletable messages that start in the first four lines 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-, D-, and E-type messages from POWER must be deleted manually.

K S,DEL=N

Manual Deletion Mode
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*).

K S,CON=N

Non-Conversational Deletion Mode
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*).

Audible Alarm

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

K S,ALM=Y

K S,ALM=N

Setting the Deletion Range

The SEG option allows you to specify the number of message lines (from 1 to 8) to be deleted at a time by the K command. (See also *Deleting Messages Manually*).

K S,SEG=6

K S,SEG=n

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

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 and 6) and enter the new characters (N and 4), or
- enter the command K S,DEL=N,SEG=4

Note that non-conversational mode (CON=N) is still in effect and the audible alarm remains switched off.

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  ┘
```

K E,SEG or K

Deleting a Predefined Number of Messages

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.

K E,n

Deleting a Single Line

To delete a single line, specify its line number n, where n may be a digit from 1 to 8.

K E,n,n

Deleting Several Lines

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.

K E,N

Deleting Line Numbers of Messages

The N option of the E operand specifies that the message line numbers in character position 2 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 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

K D,N

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.

canceled by K E,...

K D,N,HOLD

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.

Redisplaying Messages

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

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.

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  │ │
   └ └ ,Fx  ┘ ┘
```

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.

Fx - specifies redisplay of messages from a foreground partition. x is a digit from 1 to 4.

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:

```
                                      ┌Fx=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 messages 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  │,Fx    [,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 Fx, 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.

The other operands are described on the preceding page.

All operands are optional. If you just specify D, the default is D L,8 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 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.

Procedures

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)

0 Starting the System

1 Preparing the IBM 3211/3203/5203 for Buffer Load

2 IPL with the 3210 or 3215 Printer-Keyboard

3 IPL Procedure with the DOC

4 IPL Procedure with the Card Reader

5 RDE Data Entry after IPL

6 Creating the System Recorder File

7 Creating the Hard-Copy File

8 Loading the Buffers of the IBM 3211/3203/5203

9 Loading the UCB for an IBM 1403 UCS Printer

10 Assigning I/O Devices

11 Allocating Storage

12 Starting a Background Partition

13 Starting a Foreground Partition

14 Interrupting or Terminating Processing

15 Using POWER

16 RJE Terminal Operation

17 2770 Terminal Operation

18 2780 Terminal Operation

19 3780 Terminal Operation

20 Using Cataloged Procedures

21 Displaying the VTOC

22 Displaying the Label Information Cylinder

23 Displaying and Punching the Libraries

24 Displaying the Library Directories

25 Condensing Libraries

26 Printing the Hard-Copy File

27 Executing Cataloged Programs

28 Debugging Procedures

29 Shutting down the System

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).

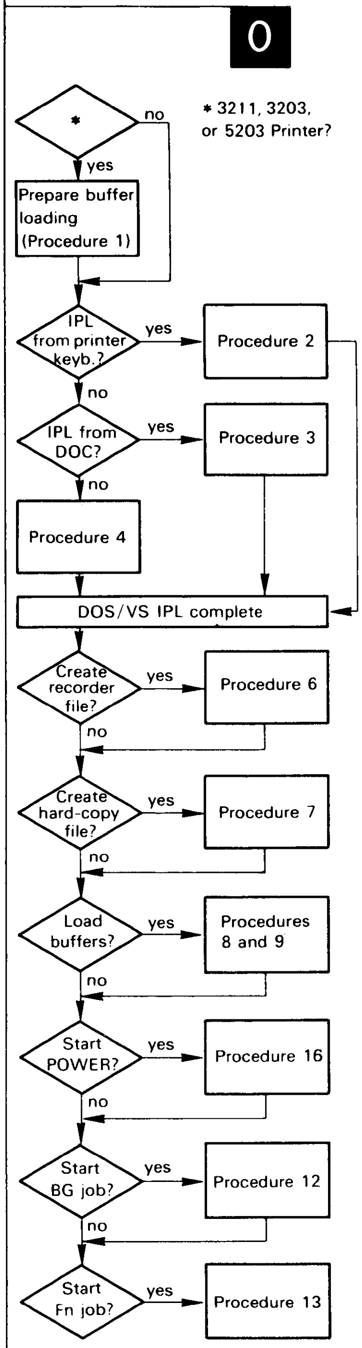
If your installation uses an IBM 3211, 3203, or 5203 Printer, buffer loading for this device must be prepared prior to the IPL procedure. Loading the buffers of these printers is described in Procedure 1.

Procedures 2, 3, and 4 describe the different methods of performing the IPL procedure:

1. From the printer-keyboard. This method applies to all System/370 CPU models, except Models 115 and 125.
2. From the DOC. This method applies to System/370 Models 115 and 125 only.
3. From a card reader. This method applies to all System/370 CPU models.

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 5 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 Model 115 or 125 without the optional console printer, you may have to create the hard-copy file. Procedures 6 and 7 describe how these files are created. Buffer loading of the IBM 3211, 1403, 3203, and 5203 Printers is described in Procedures 8 and 9. Once you have completed these preparations, you are ready to initiate POWER or run jobs in the background or foreground partitions (Procedures 15, 12, and 13, respectively).



Preparing the IBM 3211/3203/5203 for Buffer Load

1

If your computing system includes an IBM 3203, 3211, 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 load, while a UCB is loaded with the buffer load for an A11 (3211) or AN (3203 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. Adjust the paper in the carriage to the channel 1 position.
3. Ready the printer.

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

IPL with the 3210 or 3215 Printer-Keyboard

2

> If you have many ADD and DEL commands, it is faster to use the card reader (refer to Procedure 4).

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

2. Mount the SYSRES disk pack on a disk drive and ready this device.

3. Mount the pack containing the page data set on the disk drive assigned to SYSVIS. 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.

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. This prints out the following message:

   ```
   0103A 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. This prints one of the following sets of messages:

   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 printed on SYSLOG (see step 6), 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 ENABLE SET.

      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 ENABLE SET.

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

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

C. Enter the SET command with DATE and CLOCK parameters and press ENABLE SET.

D. Enter the SET command with the DATE parameter.

E. Enter the SET command with the DATE parameter.

F. Enter the SET command with DATE and CLOCK parameters.

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

```
DOS/VS IPL COMPLETE
```

in which case you can go to steps 13, 14, and 15; or it issues the message

```
DOS/VS IPL COMPLETE
1I00A 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, 14, and 15 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, 14, and 15.

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 create a system directory list (SDL), specify the SET SDL=CREATE job control command, followed by a list of the phase names to be included in the SDL.

If you wish to use the IBM supplied SDL, enter the command EXEC PROC=SDL.

15. If you wish to create an SVA and SDL for the VSAM modules and include your VSAM modules in this SVA specify:

```
        EXEC PROC=IKQVPSVA
```

> Does your system use RDE?
>
> If so, turn to Procedure 5.

IPL Procedure with the DOC

3

If you have many ADD and DEL commands, it is faster to use the card reader (refer to Procedure 4).

1. Press POWER ON and wait until PROGRAM LOAD appears on the screen.

2. Mount the SYSRES disk pack on a disk drive and ready this device.

3. Mount the pack containing the page data set on the disk drive assigned to SYSVIS. (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.)

4. Type in the physical device address of the disk drive that holds the SYSRES disk pack. Type in the character C and press ENTER.

5. When WAIT appears on the screen, press REQUEST. This 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.

6. When WAIT appears on the screen again, press REQUEST. This displays one of the following sets of messages:

   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
   0I10A GIVE IPL COMMANDS
   ```

7. Depending on the messages that were printed on SYSLOG (see step 6), 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 ENABLE SET.

      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 ENABLE SET.

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

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

C. Enter the SET command with DATE and CLOCK parameters and press ENABLE SET.

D. Enter the SET command with the DATE parameter.

E. Enter the SET command with the DATE parameter.

F. Enter the SET command with DATE and CLOCK parameters.

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

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

10. Press ENTER. The system then issues the message

```
DOS/VS IPL COMPLETE
```

in which case you can go to steps 11, 12, and 13, or it issues the messages

```
DOS/VS IPL COMPLETE
1I00A 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 11, 12, and 13 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 11, 12, and 13.

11. 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.

12. If you wish to create a system directory list (SDL), specify the SET SDL=CREATE job control command, followed by a list of the phase names to be included in the SDL.

If you wish to use the IBM supplied SDL, enter the command EXEC PROC=SDL.

13. If you wish to create an SVA and SDL for the VSAM modules and include your VSAM modules in this SVA specify:

```
EXEC PROC=IKQVPSVA
```

Does your system use RDE?

If so, turn to Procedure 5.

IPL Procedure with the Card Reader

4

If you have many ADD and DEL commands, this procedure is faster than the one using the printer-keyboard or the DOC.

1. Perform steps 1 to 4 of procedure 2 or 3, depending on the CPU model of your system.
2. Place DEL, ADD, and CAT commands (optional), as well as SET and DPD commands (mandatory) in the form of punched cards in the card reader.
3. a) If the card reader is assigned to SYSRDR,
   - press START on the reader;
   - press LOAD on the system control panel or, for the 115 and 125, enter the load address followed by C and press ENTER on the DOC;
   - when the WAIT light comes on or, for the 115 and 125, WAIT appears on the screen, press INTERRUPT.

   b) If the card reader is not assigned to SYSRDR,
   - press LOAD on the system control panel or, for the 115 and 125, enter the load address followed by C and press ENTER on the DOC;
   - when the WAIT light comes on or, for the 115 and 125, WAIT appears on the screen, press START on the card reader.

   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 has the proper values, the message DOS/VS IPL COMPLETE is issued and control is given to the control program.
   - If the 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 in the 'set' state, the message DOS/VS IPL COMPLETE is issued and control is given to the control program.
   - 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 DOC.
4. 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.
5. If you wish to create a system directory list (SDL), specify the SET SDL=CREATE job control command, followed by a list of the phase names to be included in the SDL
   If you wish to use the IBM-supplied SDL, enter the command EXEC PROC=SDL.

4

6. If you wish to create an SVA and an SDL which include the VSAM modules, specify PROC=IKQVPSVA.

7. WARNING: Warm start copies of the SVA are not used since you cannot respond to the accept or reject message (1100A). Thus you can only rebuild the SVA and SDL.

Does your system use RDE?

If so, turn to Procedure 5.

RDE Data Entry After IPL

5

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

1I89A 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

1I92I INVALID CODE

appears on SYSLOG and message

1I89A IPL REASON CODE =

is re-issued until you have entered a valid code. When you have entered a valid reason code, the message

1I91A 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.*

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.*

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

Creating the System Recorder File

6

The Models 115 and 125 without software recording require no 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 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  DOS/VS IPL COMPLETE
1I00A  READY FOR COMMUNICATIONS
ASSGN SYSREC,X'190'
SET RF=CREATE
// OPTION STDLABEL
// DLBL IJSYSRC,'DOS.SYSTEM.RMSR.FILE'
// EXTENT SYSREC,,,,1700,43
// EXTENT SYSREC,,,,1700,43
.
. (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.

The IBM System/370 Models 115 and 125, if operated without the optional console printer, requires a disk extent on which to record the lines written to the DOC, both by the system and the operator. This disk extent is called the hard-copy file and resides on the device assigned to SYSREC.

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

The hard-copy 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 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  DOS/VS IPL COMPLETE
1I00A  READY FOR COMMUNICATIONS
ASSGN SYSREC,X'190'
SET HC=CREATE
// OPTION STDLABEL
// DLBL IJSYSCN,'DOS HARD COPY FILE'
// EXTENT SYSREC,,,,nnnnn,nnnnn
// EXTENT SYSREC,,,,nnnnn,nnnnn
.
. (other label information for this disk pack)
.
// JOB FIRST
```

First statement of the normal job stream. When this statement is read, the hard-copy file is created and opened.

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

Printing the hard-copy file is described in Procedure 26.

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

Loading the Buffers of the IBM 3211/ 3203/5203

8

A job may require printed output on a 3211, 3203, or 5203 printer in a nonstandard page layout or with the use of a nonstandard print train or both. In these cases you must load the Forms Control Buffer (FCB) and/or Universal Character Set Buffer (UCB) of these printers with the appropriate buffer load.

Load the FCB

Loading the FCB

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

1. Enter the following control statements on the card reader:

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

SYSxxx - symbolic device name of the printer

phasename - name (to be provided by your programmer) by which the FCB load phase is cataloged in the system core image library.

NULMSG - causes the printing of the buffer load verification message to be suppressed. Since this message enables you to verify that the contents of the buffer agrees with the new printer forms to be used, specifying NULMSG is not recommended.

If phasename is omitted, the FCB load phase (provided by your programmer in cards) must be placed on SYSIPT.

2. Restore the printer carriage so that the buffer is aligned to channel 1.

3. Adjust the paper in the carriage to the channel 1 position and ready the printer.

FCB Load within a Job Entry

The job step to load an FCB under POWER (see *Loading the FCB as a Separate Job Entry*) may be placed anywhere within a job entry. When POWER prints the output of a job entry and encounters an FCB card, 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. After the message

1Q80E JOB ENTRY name NEEDS FORMS ffff for xxycuu

proceed as follows:

- Place the fold of the new form on the fold of the old form that shows just above the print chain.
- Enter the command `G PRT,cuu`

where cuu is the address of the printer.

'ffff' in message 1Q80E is the number specified in the forms-number parameter of the * $$ PRT statement (see the section *Reference Information*). If several FCB loads are contained in one job entry, the message 1Q80E is printed for each FCB load, but 'ffff' will always be the number specified in the * $$ PRT statement, even if the FCB loads require different forms. In such a case, ask your programmer what forms are needed.

Loading the FCB as a Separate Job Entry

Load the FCB under POWER

This procedure allows you to change the contents of the FCBwhenever required. Submit the following job entry in any partition that is waiting for work:

These JECL statements are discussed in detail in the section *Reference Information.*

```
* $$ JOB FCB
* $$ PRT Da
// JOB FCB
// EXEC SYSBUFLD
   FCB SYSLST,phasename
/&
* $$ EOJ
```

Then enter the command `S PRT,cuu,,a`
The FCB will now be loaded as described under *FCB Load Within a Job Entry.*

Loading the UCB

Load the UCB

1. Mount the new print train on the 3211, 3203, or 5203 and ready the device.

2. Enter the following control statements on the card reader:

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

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

SYSxxx - symbolic device name of the printer.

phasename - name (to be provided by your programmer) by which the UCB load phase is cataloged in the system core image library.

FOLD - causes lower-case character codes to print as upper case characters (use this parameter only if requested by your programmer).

8

NOCHK - causes data checks resulting from mismatches of printline characters with the UCB to be ignored (use this parameter only if requested by your programmer).

NULMSG - causes the printing of the buffer load verification message to be suppressed. Since this message enables you to verify that the contents of the buffer agrees with the new print train to be used, specifying NULMSG is not recommended.

The optional operands FOLD, NOCHK, and NULMSG can be entered in any order.

Load UCB under POWER

Loading the UCB as a Separate Job Entry

The procedure for loading the UCB under POWER is the same as that for loading the FCB. The control statements required are:

```
* $$ JOB UCB
* $$ PRT Da
// JOB UCB
// EXEC SYSBUFLD
   UCB SYSLST,phasename
/&
* $$ EOJ
```

Loading the UCB for an IBM 1403 UCS Printer

9

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. The procedure is as follows:

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

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

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

SYSxxx - symbolic device name of the 1403.

phasename - name (to be provided by your programmer) by which the UCB load phase is cataloged in the system core image library.

FOLD - causes lower-case character codes to be printed as upper case characters (use this parameter only if requested by your programmer).

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 - causes the printing of the buffer load verification message to be suppressed. Since this message enables you to verify that the contents of the buffer agrees with the new print train to be used, specifying NULMSG is not recommended.

The optional operands FOLD, BLOCK, NULMSG can be entered in any order.

The UCS command cannot be used to load the UCB of a 3211, 3203, or 5203 printer. See Procedure 8.

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
   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).

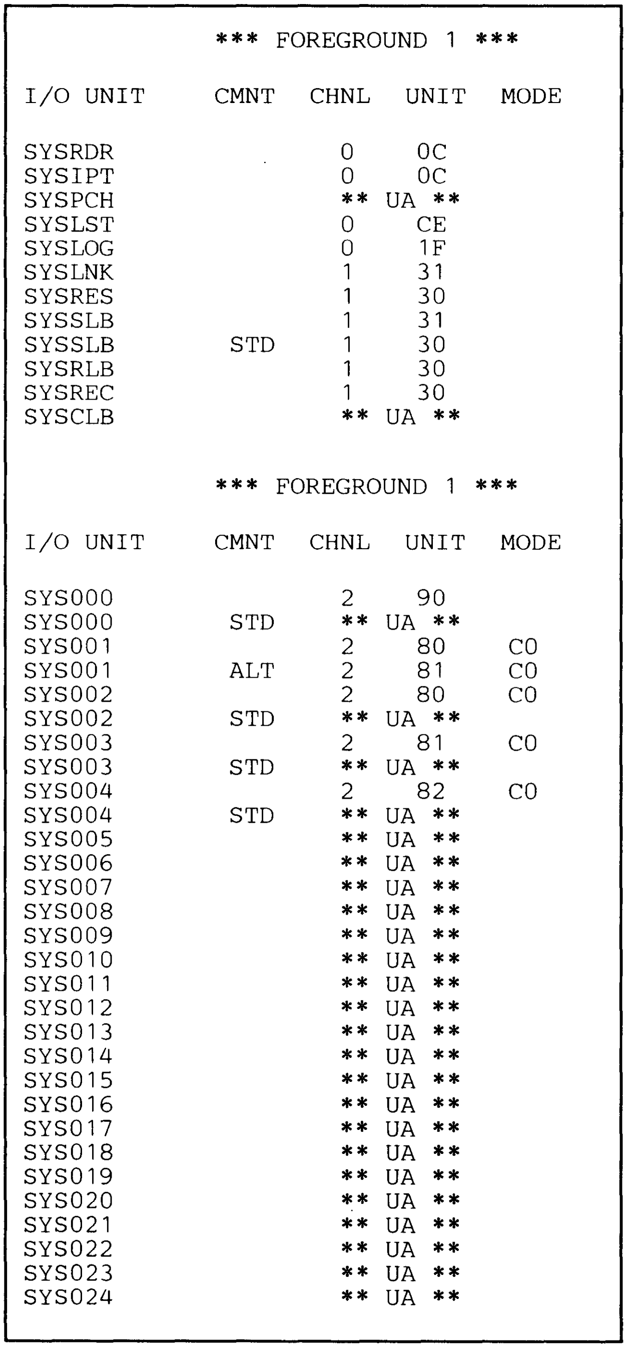
In the case of assignments that are temporary only, UA is printed between the channel and unit columns.

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.

Allocating Storage

11

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 a Background Partition

12

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

Immediately after the message DOS/VS IPL COMPLETE, 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, place the card deck containing the job control information in the card reader that is to be used as SYSRDR, and ready that reader.

If SYSRDR has already been assigned to the card reader, 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 to the card reader 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

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 message

```
AR 1I60A READY FOR COMMUNICATIONS
```

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

BATCH F1 or START F1

BATCH and START have identical functions

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 place the card deck in the reader, ready this device, 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 ALL
```

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

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.

Starting a Foreground Partition (continued)

13

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.

Interrupting or Terminating Processing

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
   ```

In 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.

This sample procedure for initiating POWER and running jobs under the control of POWER is based on the assumption that the following I/O devices are available for POWER operation:

DA=direct access

	Device Address
IBM 2540 Card Reader	X'00C'
IBM 2540 Card Punch	X'00D'
IBM 1403 Printer	X'00E'
IBM 2314 DA Storage Facility	X'132'
IBM 2780 Data Transmission Terminal	X'060'

The direct access storage facility is used for the POWER files; the data trânsmission terminal for remote job entry.

This example also assumes that POWER resides in F1, the partition with the highest priority, and supports BG, F2, and F3.

Dummy devices

Dummy devices generated or added during IPL for the partitions supported by POWER:

BG	**F2**	**F3**	
X'01C'	X'02C'	X'03C'	for readers
X'01D'	X'02D'	X'03D'	for punches
X'01E'	X'02E'	X'03E'	for printers

Although the input and output from and to the unit record devices is temporarily kept on disk, you need these dummy devices for the assignment of the logical units SYSRDR, SYSIPT, SYSPCH, and SYSLST. The device type of the dummy devices must match the type of the devices used by the POWER routines.

The POWER program can be initiated only when no processing is taking place in the partitions that are to be supported by POWER. This presents no problems, since you will normally initiate POWER immediately after IPL. Should the partitions contain executing programs, you must wait until they have finished processing and then issue STOP commands for these partitions (see Procedure 14).

15

Initiating POWER

1. Mount the disk pack(s) used for the POWER files on the disk drive(s) and ready device(s).

2. Make the reader, the punch, and the printer available by effecting the necessary unassignments in the partition that 'owns' these devices:

```
ASSGN SYSIN,UA
ASSGN SYSPCH,UA
ASSGN SYSLST,UA
ASSGN SYSxxx,UA  (if necessary)
```

3. Stop all running partitions that are to be supported by POWER, using the STOP command after the jobs in those partitions have reached EOJ.

EOJ appears on SYSLOG

4. Allocate sufficient real storage to the POWER partition:

```
ALLOCR F1R=34K
```

   Your system programmer will tell you how much storage to allocate. Remember that POWER operates in real mode only.

5. Start the POWER partition by entering the attention command

```
BATCH F1
```

6. Assign SYSIN and SYSLST:

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

   Note that the assignment for SYSIN is required only if you use the control statements listed under 7 below from a reader. The assignment for SYSLST enables you to obtain a dump when POWER is canceled.

7. Place the POWER initiation cards in the card reader. These cards are available at your installation. In our example they look like this:

SYS001 used for QFILE

SYS002 used for DATAFIL

```
// JOB POWER
// ASSGN SYS001,X'132'
// DLBL QFILE,'QFILE POWER',99/365,DA
// EXTENT SYS001,111111,1,0,20,20
// ASSGN SYS002,X'132'
// DLBL DATAFIL,'DATAFIL POWER',99/365,DA
// EXTENT SYS002,111111,1,0,40,1000
// EXEC FGPSPOOL,REAL
```

   SYS001 and SYS002 are both assigned to X'132', which holds the POWER files QFILE and DATAFIL.

8. Ready the card reader. The job control program in F1 will now read the control statements and load POWER. The first POWER message on SYSLOG is:

```
1Q39D FORMAT QUEUES?
```

Queue formatting

9. There are four possible responses to this message:

Q = format the QFILE

D = format the DATAFIL and the QFILE

A = format the ACCTFIL

NO = no formatting is required. NO is assumed when you press END or ENTER

ACCTFIL=accounting file (optional)

You may enter one or more responses; if you enter more than one, separate them by commas. A NO response cannot be combined with other responses. If your system does not support job accounting, do **not** respond with 'A'.

The responses to the message FORMAT QUEUES? depend on whether you want a 'warm' or a 'cold' start. A warm start allows you to continue processing input data, and punching and printing output data contained in the POWER files. If you perform a warm start, you must not format the files. If you perform a cold start, for instance when disk packs are used for POWER for the first time, formatting of the files is required.

Warm and cold start

Autostart option

10. If the autostart option was specified when POWER was generated, the following message will appear on SYSLOG:

```
1Q40D AUTOSTART?
```

The two possible responses to this message are:

YES - The partition-independent reader and writer routines are started automatically at POWER initiation. Once stopped, you can start them yourself, using the S command. YES is assumed when you press END or ENTER.

NO - Automatic start is not performed.

When you have entered your responses to the messages 1Q39D FORMAT QUEUES? and 1Q40D AUTOSTART?, the POWER files are opened.

When you perform a cold start, the files will be formatted, which requires a certain amount of time, depending on the size of the files to be formatted. When formatting has completed, the following messages appear on SYSLOG:

```
1Q41I POWER CONTROLLED PARTITIONS MAY NOW BE STARTED
1Q14I xx PROGRAM BUFFERS
```

15

1Q15I xx DATA BUFFERS

These messages require no operator action. The last message enables you to check whether sufficient data buffers were allocated.

Three data buffers are necessary for each supported partition that is running under POWER. At least one data buffer is required for each reader or writer routine. An RJE routine requires two data buffers.

A reader routine using the 3540 reader uses as many data buffers as necessary to hold the data of one diskette track.

Starting a POWER-Supported Partition

Start the POWER-supported partitions, using the BATCH (or START) command. The following example shows what you have to do to start the background partition.

These are the dummy assignments for BG.

Output on SYSLOG	Operator Action
	Press REQUEST
AR READY FOR COMMUNICATIONS	
AR	`BATCH BG`
BG	`ASSGN SYSIN,X'01C'`
BG	`ASSGN SYSPCH,X'01D'`
BG	`ASSGN SYSLST,X'01E'`
BG	Press END or ENTER

If SYSPCH is assigned to a 2560 MFCM or 5425 MFCU, the second ASSGN statement should read: `ASSGN SYSPCH,X'01D',H2`

Start the other POWER-supported partitions in the same way, using the appropriate dummy devices.

If input for this partition is available in the POWER reader queue, processing will start immediately. If no job entry is available in the reader queue, the following message appears on SYSLOG:

F1 1Q65I WAITING FOR WORK BG

You must now start a reader routine to supply this partition with its card input (see *Starting a POWER Reader Routine*).

Issuing POWER commands

You have a set of POWER operator commands available, which enable you to control POWER operation. The POWER routine-management commands are:

	Command
Start a reader/writer routine	S
Cancel a reader/writer routine	C
Stop a reader/writer routine	P
Restart printer output from a specified page	T
End POWER support	E
Flush printer or punch output of the job entry being printed or punched	F
Reactivate a waiting routine or partition	G
Display/alter copy counter of the job entry being printed or punched	M

The POWER queue-management commands serve to manipulate the job entries in the POWER files; these commands are:

	Command
Alter priorities of job entries	A
Obtain status information on the running POWER system and the job entries contained in it	D
Place a job entry in the 'hold state', which means temporarily suspend processing of job entries	H
Process the job accounting file	J
Delete a job entry from a queue	L
Release a job from the 'hold' state	R

These commands are entered just as attention routine commands as follows (omit steps 1 and 2 for Models 115 and 125):

1. Press REQUEST.
2. Wait for message READY FOR COMMUNICATIONS.
3. Enter the command.
4. Press END or ENTER.

Full details of the POWER commands and their operands are contained in the section 'Reference Information'.

15

Whenever you enter a queue-management command related to a particular job entry, make sure that your command correctly identifies that job entry by the name under which it was logged by POWER and, possibly, also by the number which POWER assigned to the job entry. A number is required when job entries with identical names were entered.

D command

If you are in doubt about the correct identification of a job entry, obtain a status report by entering the D command with the ALL option prior to entering the queue-management command in question. The status report provides you with a list of all job entries logged by POWER in a specific queue.

Starting a POWER Reader Routine

A reader routine reads input for the associated partition, or, if it is independent, for all supported partitions.

S command

To start a reader routine, use the S command. In the example described under *Starting a POWER-Supported Partition*, you would enter

```
S BGRDR,00C
```

or, to start a partition-independent reader,

```
S RDR,00C
```

Starting a POWER Writer Routine (Print or Punch)

A writer routine prints or punches output for the associated partition, or, if it is independent, for all supported partitions.

Writer routines are started in the same manner as reader routines. To start a writer routine for the background partition in our example, you would enter one or both of the following commands:

S BGPRT,00E	if you want to print
S BGPUN,00D	if you want to punch

If background output is available to the POWER files, printing and punching starts immediately. The commands

```
S PRT,00E
S PUN,00D
```

start writer routines that process output from all supported partitions.

Starting a POWER RJE (Remote Job Entry) Routine

An RJE routine enables a user at a 2770, 2780, or 3780 remote terminal, which is equipped with a reader, a punch (optional for 2770 and 3780), and a printer, to

- submit jobs from his terminal to your system, and
- obtain output from your system at his terminal.

A set of POWER operator commands is used to control RJE functions. These commands are:

	Command
Start an RJE routine	S
Stop an RJE routine	P
Reactivate a waiting remote routine	G
Display the status of remotely submitted jobs	D
Broadcast information to terminals	B
Inquire about RJE activity	I
Change the output destination of job entries	O

Some of these commands are the same as those for local routine management, although they have different operands. They are entered in the same manner as routine and queue-management commands. Full details of the RJE commands are contained in the section *Reference Information.*

You would start an RJE routine in the example described under *Starting a POWER-Supported Partition* by entering the command

```
S RJE,060,2780
```

POWER acknowledges this command by issuing the message

```
1R24I 060 2780 HAS BEEN STARTED
```

When this message has appeared on SYSLOG, the operator at the terminal may start his RJE activities.

Stopping POWER Reader, Writer, and RJE Routines

POWER routines should be stopped only if they are not processing input or output and the last job entry has been read in completely.

As far as input processing is concerned, wait until the hopper of the card reader is empty.

As far as print output is concerned, check whether or not the printer is active. You must be sure that the last job entry has been printed completely. If the printer is active, the remaining job entries should be placed in the 'hold' state by means of the H command. When you have issued this command, wait until the job entry in progress has been completed.

H command

As far as punch output is concerned, use the P command with the EOJ option:

P command

15

```
P BGPUN,00D,EOJ
```

This command causes the punch routine to stop after completion of the job entry in progress.

I command

The I command enables you to obtain status information regarding an RJE routine.

Terminating POWER

When the POWER program is no longer required at the end of the day, or when jobs that may not be executed in the POWER environment must be run, you will terminate POWER.

Normal Termination

E command

Before you terminate POWER, inform the operators of remote work stations, if any, that the shutdown procedure is about to begin. Enter E. The E (end) command without an operand causes all POWER routines to complete processing their current job entries and then stop. After the message

```
1Q42I POWER HAS BEEN TERMINATED
```

normal processing without POWER can take place again.

Abnormal Termination

E KILL

When it is necessary to terminate POWER immediately because things are going wrong, use the E command with the operand KILL. This command causes POWER to terminate immediately and produce a dump (printout of storage) if SYSLST for the POWER partition is assigned.

As a result of the E KILL command, input and output of jobs may be broken off. A warm start enables you to restart output from the beginning of the interrupted job. Input of a job entry that was interrupted by the E KILL command must be resubmitted completely.

IBM 1442 Card Read Punch, 2560 MFCM, 5425 MFCU, and 3525 Card Punch with Reader Feature

If your installation uses either of these devices for POWER, remember that they cannot be used simultaneously for a reader **and** a punch routine.

If it is necessary to start the reader routine for one of these devices while the punch routine is still processing, you must stop the punch routine first, using the P command. Example:

```
        P xxPUN,00D,EOJ
   or   P xxPUN,00D,CHECKPOINT (2560 and 5425 only)
```

where xx = partition identifier

00D = device address of reader-punch

The first example causes the punch to stop on completion of the current job entry and the message

1Q77I STOPPED xxT00D

is issued on SYSLOG, where xx indicates the partition.

You may now start the reader routine, using the S command:

```
S xxRDR,00D
```

If you are using a 2560 MFCM or 5425 MFCU, you need not wait for completion of the current job entry in order to start the reader routine. You can interrupt the punch routine by specifying CHECKPOINT in the P command, start the reader routine, and - upon termination of the reader routine - restart the punch routine by means of the RESTART operand of the S command.

Assume, for example, that you are using a 5425 MFCU at address '00D' and you must stop punching cards of the job entry named LONG to read in more input to the background partition. Enter the following command:

```
P xxPUN,00D,CHECKPOINT
```

A checkpoint is taken, punching is stopped, and the following messages are issued:

```
1Q79I BG LONG 00027 HAS BEEN CHECKPOINTED
1Q77I STOPPED BGP00D
```

You can now start the desired reader routine by entering

```
S BGRDR,00D
```

After the reader routine has completed reading the new input, stop it and restart the interrupted punching by entering

```
S xxPUN,00D,,RESTART
```

This causes the message

1Q68I BG LONG 00027 HAS BEEN RESTARTED

to be issued, and the punch routine resumes where it left off when the punching was interrupted.

Using the IBM 3540 Diskette Input/Output Unit

You can use the 3540 unit as a POWER input device for SYSIN FILES AND DATA files. The diskette unit can be used either instead of a card reader or as supplement to the card reader. POWER can, for example, read the job control statements from the card reader and then switch to the diskette unit to read a data file, or it can read control state-

15

ments and card image data from the diskette unit, using no card reader at all.

Note that although a diskette file may be inserted into a card input stream, it is not possible to insert card data into the middle of a diskette file or to insert one diskette file into another.

If a diskette unit is the only input device, the information needed by POWER to read the input file must be specified in the S command. For example, to start a background reader routine to read a file named INPUT from two diskette volumes on device 00B, enter the command

```
S BGRDR,00B,,'INPUT',2
```

If you wish to insert a diskette file into the card input stream, you must indicate this to the reader routine by means of the * $$ RDR statement provided by the programmer. For example, to insert a data file INDATA contained on one diskette on device 00B into the card input stream on device 00C, enter the following:

```
S BGRDR,00C,,00B
```

Input stream on card reader:

```
// JOB JOB1
// ASSGN SYS008,X'01B' (dummy 3540)
// DLBL FILEA,'INDATA',,DU
// EXTENT SYS008
// EXEC JOB1
* $$ RDR SYS008,'INDATA'
/*
/&
```

For more examples of the use of diskette input files refer to *DOS/VS System Control Statements, GC33-5376.*

RJE Terminal Operation

The Remote Job Entry (RJE) of POWER enables you to enter jobs for processing under DOS/VS from a remote 2770, 2780 or 3780 Terminal. Once the job has been entered into POWER files, it will be executed as though it had been submitted locally.

Input from the Terminal to the Central System

The input stream at the terminal consists of job entries and terminal commands.

Job Entries

Job Entry Control Language (JECL)

The job entry is the primary input to POWER RJE. It consists of the job to be executed at the central system and the optional Job Entry Control Language (JECL) statements. The JECL statements * $$ JOB and * $$ EOJ, for example, are used to identify and delimit a job entry to POWER RJE.

JECL also provides a convenient means of specifying how POWER is to handle a particular job entry. If JECL statements are not used, system options are assumed when the job entry is received by the central system.

Normally you will not be concerned with JECL; occasionally, however, your programmer might want you to change or correct his JECL cards. You will find details of all JECL statements in the section *Reference Information.*

Terminal Commands

You can make specific requests to POWER RJE by using terminal commands. These commands are entered from the card reader and may appear anywhere within the input stream, except within a job entry. The terminal commands enable you to:

- define the state of the terminal (for example active, inactive)
- manipulate job entries and their associated output
- provide job entry and system information
- communicate with the central system operator

The terminal commands are described in detail in the section Reference Information.

Output from the Central System to the Terminal

Two kinds of output are received at the terminal: job entry output and messages.

Job Entry Output

Options

Job entry output is the result of the execution of locally or remotely submitted jobs entries. A number of options can be specified in JECL statements and terminal commands, which enable you or the central operator to:

- have output printed or punched at your terminal or locally, at the central system.
- direct output to another user or, on a read-only basis, to all terminal users.
- hold output at the central system until an OUTPUT terminal command is issued.
- request multiple copies of your output.
- continue interrupted output (with the CONTINUE terminal command).
- delete job entries on the input or output queue (DELETE terminal command).
- specify type of forms to be used for punched and printed output.

Messages

Messages received at a terminal include responses to input from the terminal, diagnostic messages, and broadcast messages issued by the central operator. The broadcast messages include any information considered desirable by the central operator, for example, close-down of the central system, loss of a central device, or addition of a central device. These messages appear on the printer after the output of one job entry and before the output of the next.

Broadcast messages appear only when you request them by issuing a BRDCSTR terminal command, or by an RJSTART command with the BRDCST option. You may also send messages from the terminal to the central operator by means of the MSGR terminal command. These messages are printed on SYSLOG when they are received by the central system.

Initiating and Terminating POWER RJE Operation

The following paragraphs describe the startup and closedown procedures for POWER RJE operation.

Startup Procedure

To initiate Remote Job Entry from the terminal, you must submit an RJSTART and a LOGON command (after startup procedures have been completed at the central system and an RJE routine has been started for your teleprocessing line).

active state

The RJSTART command places the terminal in the **active** state, that is, it becomes logically attached to POWER RJE. The termid operand in the RJSTART command identifies the terminal to POWER RJE and prevents unauthorized access to the system.

You may now access POWER RJE by logging on or wait for any broadcast messages from the central installation operator (provided you have specified BRDCST in the RJSTART command).

LOGON

The LOGON command identifies you, the terminal operator, to POWER RJE and allows you to access it. The userid operand in the LOGON command ensures that only authorized users have access to POWER RJE and it protects your output against unauthorized access. The LOGON command defines the beginning of a user session. You may now transmit input to the central system or request job output.

Whenever you begin a terminal session, a message appears on the central operator's console, containing your userid and the time you logged on.

Closedown Procedure

To terminate POWER RJE operation from the terminal, you must submit a LOGOFF and an RJEND command.

LOGOFF

Enter the LOGOFF command if you want to terminate a session. When the central system receives your LOGOFF command, you - as the current user - will be logically detached from POWER RJE; however, the terminal remains in the **active** state, that is, attached to the central system. You may restart a session by entering another LOGON command. Job entries can be transmitted, and output can be received, only during a user session.

If the central system receives a valid LOGON command from a terminal with a session in progress, the central system logs off the current user and logs on the user identified in the LOGON command. If the central system receives an RJEND command from a terminal with a session in progress, it logs off the user and logically detaches the terminal. In this manner, you may omit the LOGOFF command when changing user sessions or detaching the terminal.

Whenever you end a terminal session, a message appears on the central operator's console giving your userid and the time you logged off.

RJEND

inactive state

The RJEND command terminates your RJE activities and logically detaches your terminal from POWER RJE. The central system places the terminal in the **inactive** state. You will receive a message that the closedown procedure has completed, after which the terminal may be used for local processing. RJE activity can be resumed with an RJSTART command only.

If the terminal is connected to the central system via a switched line, the connection is broken, and the line is available for another terminal.

A closedown may also be effected by the central operator issuing a P (stop) command, in which case you receive a message to that effect.

This procedure describes how to operate a 2770 terminal.

Transmitting Input

1. Turn on the power switches at the 2772 Control Unit and the 545 Card Punch.
2. Set the Job Select switch at the 2772 console to VARIABLE. If one of the five predefined job setups corresponds with the settings of steps 2 through 8, omit these steps and set the Job Select switch to the corresponding position.
3. Set the Term Mode switch to LINE.
4. Set the Answer switch to MANUAL.
5. Turn off the Inquiry Mode and the Selection Required switches.
6. Turn on the Input switch which is associated with the card reader.
7. Turn on the Output Printer switch.
8. Turn on the Output switch which is associated with the card punch, if it is available.
9. At the card punch:
   a) Place blank cards in the hopper.
   b) Set the Auto Pch/Key Pch switch to KEY PCH.
   c) Punch upper-case D in columns 2-80 of a card, insert the card in the program drum, and lower the starwheels.
   d) Feed three cards by pressing FEED twice and RELEASE once.
   e) Set the Auto Pch/Key Pch switch to AUTO PCH. For Model 4, be sure that the Print and LZ Print switches are set on.
10. Press CARRIAGE RESTORE on the 2203 or 2213 Printer (Model 2) to align forms.
11. Press NPRO (Non-process Runout) on the 2502 Card Reader to flush cards.
12. Place the input cards (after RJSTART and LOGON cards) in the reader.
13. Press CHECK RESET and TERM RESET on the 2772 console.
14. If transparent data is being transmitted or received in this session, turn on the TRNSPCY switch on the 2772 console. Do not change the switch position during a session. (If the 2770 is equipped with the Buffer Expansion feature, buffer capacity increases to 256 or 512 bytes from 128 bytes, and variable length, multiple-card records will be stored from card reader input if in nontransparent mode.)
15. If the space compress feature is to be used, turn on the SPACE COMPRESS EXPAND switch on the 2772. Note that this feature can only be used in non-transparent mode, that is, the TRNSPCY switch must be turned off.

16. Set the EOF (End-of-File) switch to ON (if no more cards are to be transmitted) and press START on the 2502 reader panel to ready the card reader.
17. In case of a switched line, dial the number and switch the data set to data mode. The actual procedure depends on the type of data set at your location, and can normally be found in the operator's instructions for the data set.
18. Press START on the 2772 console to transmit data.
19. The alarm sounds when:
    a) the last card in the hopper has been read and transmitted. If the EOF switch was set to ON to indicate that this was also the last card of the input stream, transmission ends normally. If the EOF switch was left in the OFF position because more cards had to be transmitted, refill the hopper, set the EOF switch to ON if this is the last group of cards for this job and press START on the card reader and then on the 2772 console to resume the input operation.
    b) If a message is to be printed during input, your terminal will receive an RVI (reverse interrupt) signal from the CPU. The PROCSR IRPT (Processor Interrupt) light is turned on at the 2772 console, when line turnaround is triggered by RVI. Press START on the 2772 console to turn the light off and continue reading cards after the message has been printed.

Receiving Output

1. When print or punch output is available to be received at the terminal and the output device is not in the ready state, the alarm will sound.
2. Correct error indicators on the printer and/or punch according to error-recovery procedures in *Operator's Guide: IBM 2770 Data Communication System, GA27-3038.*
3. Press CHECK RESET and START on the 2772 console to return the terminal to ready state.
4. If it is necessary to submit more input while output is in progress, discontinue output (see *Discontinuing Output* later in this chapter).
5. The end of output is signaled by the audible alarm, at which time you may submit more input (see *Preparing More Input* later in this chapter).

Discontinuing Output

1. a) Press STOP on the printer control panel if you want to discontinue print output in progress.
   b) Set the Auto Pch/Key Pch switch to KEY PCH if you want to discontinue punch output in progress.
2. When output has been discontinued, ready the printer again, upon which the central system will send you a message requesting a CONTINUE command.

17

3. Enter a CONTINUE command (see Reference Information). If the next card submitted is not a CONTINUE command, the central system will flush the reader and write a message stating that the job stream has been flushed.
4. In the CONTINUE command you may specify (with the HOLD operand) that the job output is to be held until requested via an OUTPUT command at some later time. In the meantime you may submit more input. For the other options of the CONTINUE command, see Reference Information.

Special Forms Output

1. If the programmer has, as is recommended for RJE, specified special forms or cards in his * $$ PRT or * $$ PUN statements and the output is ready to be transmitted, the central system will send you a message indicating the required form or card number.
2. For printed or punched output, place the required forms or cards in the printer or punch and transmit a CONTINUE command from the reader. Upon receipt of this command the central system transmits the output.

Preparing More Input

1. While output is printed or punched, you may load and ready the reader to send more input. (If you ready the reader **after** output has been completed, you must do so within approximately 30 seconds multiplied by the number of retries specified in the RETRY generation option to prevent the central system from disconnecting the line.)
2. When print or punch output has completed, input will be transmitted.

Error Recovery Procedures

When the central system detects a permanent line I/O error, it will usually disconnect your terminal. To reestablish the line connection, redial (for a switched line), and resubmit RJSTART and LOGON commands. Issue the STATUS command to get your input acknowledged and resubmit the interrupted job entry.

Failure During Input

When the card reader fails (because of a card jam, for example), correct the condition according to the error-recovery procedures defined in *Operator's Guide: IBM 2770 Data Communication System, GA27-3038.*

Whenever the terminal stops transmitting input, a time-out occurs after 30 seconds. The system will retry the read operation as many times as is specified in the RETRY system generation option. The default is six. Each retry takes 30 seconds.

If the retries are unsuccessful, the user session is terminated and the terminal is detached from POWER RJE.

Resume input operation

To resume input operations, redial (if the connection is via a switched line) and resubmit the RJSTART and LOGON cards. Issue the STATUS command to get your input acknowledged and resubmit the interrupted job entry.

Automatic Answering Feature

Since the Automatic Answering feature, if active on a switched line, disconnects the line if recovery is not made within 21 seconds, it should be disabled during RJE operation.

Input Stream Flushed

Error messages

Upon detection of an invalid command or a user error, the central system will return one of the following messages to the terminal (see also *DOS/VS Messages, GC33-5379*):

1R65A	INVALID/MISSING TERMID/USERID
1R63A	INVALID RJE WORK STATION COMMAND
1R64A	INVALID RJE WORK STATION COMMAND - NO USER LOGGED ON
1R74A	TWO userid USERS ARE ALREADY LOGGED ON
1R53A	CARD FORMS formsid NEEDED FOR JOB ENTRY name. CONTINUE COMMAND REQUIRED
1R54A	CONTINUE COMMAND IS REQUIRED FOR PRINTER/PUNCH
1R70A	PRINT FORMS formsid NEEDED FOR JOB ENTRY name. CONTINUE COMMAND REQUIRED

After this the remaining cards in the hopper are flushed to enable you to correct the error. This is indicated by the additional message

1R67I	JOB STREAM FLUSHED

When you receive an error message on the printer as a result of an input error, adopt the following procedure:

a. Press STOP on the card reader.

b. Remove the telephone receiver and switch the data set to talk mode (switched line only).

c. Press NPRO on the card reader.

d. Locate the error card and correct it.

e. Load the corrected card and the remaining cards into the reader.

f. Press CHECK RESET and TERM RESET on the 2772 console.

g. Press START on the card reader.

h. Switch the data set to data mode and hang up the telephone receiver.

i. Press START on the 2772 console to continue transmission.

Failure During Output

Failure during output may occur as a result of a failing printer or punch, a device that is not ready, or failure of the central system.

Unit Failure

When the printer or punch fails, correct the condition according to the error-recovery procedures defined in *Operator's Guide: IBM 2770 Data Communication System, GA27-3038.*

The system will retry the output operation as many times as is specified in the RETRY generation option; the default is six retries.

If you can ready the printer or punch within 30 seconds multiplied by the number specified in the RETRY option, the system will issue a message requesting a CONTINUE card. Submit a CONTINUE command (see Reference Information), and the output will be resumed according to the options specified in the CONTINUE command.

If the RETRY count is exhausted before recovery can be made, the terminal is disconnected from POWER RJE. To resume output operation, redial (if the connection is via a switched line) and resubmit the RJSTART and LOGON commands. The output of the interrupted job entry will be retransmitted upon receipt of an OUTPUT command.

If the unit failure causes an irrecoverable error, the session is ended and the output is placed in the hold state. Submit an OUTPUT command in a later session (see Reference Information) to retrieve the output. (The central operator may use the O (output) command to redirect output for a terminal to a local device.)

Unit Not Ready

Device not ready

If the central system tries to transmit to a device that is not in the ready state, the audible alarm sounds, and the TERM ADD light comes on (see *Error-Recovery Procedures* in *Operator's Guide: IBM 2770 Data Communication System, GA27-3038).*

Print error

If a print error occurs, you can normally place the printer in the ready state by pressing CHECK RESET and TERM RESET on the 2772 console as well as NEW LINE or CARRIAGE RESTORE on the printer.

Punch error

If a punch error occurs (because of a card jam, for example), remove the cards that caused the failure, feed three new cards, and place the punch in the ready state by pressing CHECK RESET and TERM RESET on the 2772 console.

If you have a terminal without a card punch (NOPUNCH parameter was specified in the RJSTART card), or you are operating in non-transparent mode, and punch output is to be sent to the terminal, the output is redirected for local punching and the following information message is listed on the printer:

```
1R79I jobname DIRECTED TO LOCAL PUNCH
```

Central Failure

If the central system fails while output is in progress, the terminal is placed in the inactive state. When the central system comes back online, submit an RJSTART command followed by a LOGON command, and retrieve the output, using an OUTPUT command.

If the central system fails while input is in progress, the terminal is placed in the inactive state. Submit an RJSTART and a LOGON command when the system comes back online. Begin the interrupted user session with a STATUS command to obtain acknowledgement of jobs entered prior to the central system failure. When a failure occurs, the central system will attempt to notify you by sending the message

```
1R51I     ABNORMAL CENTRAL SHUTDOWN.
```

Testing the RJE System

Online Terminal Test

If you want to test the system to check whether transmissions are being received, use the Online Terminal Test facility. This is an optional service provided by BTAM to ensure proper operation of the system, and it may be used in the diagnosis and correction of a terminal malfunction. For details refer to *DOS/VS Basic Telecommunications Access Method, GC27-6989.*

If the online test facility is not present, you may send a message (via an MSGR command) to the central operator, requesting him to send you a broadcast message. If you receive it on the printer upon submitting a BRDCSTR command the system may be assumed to be working properly.

2780 Terminal Operation

18

This procedure describes the specific action required to operate a 2780 terminal.

Detailed information about the 2780 is contained in *IBM 2780 Data Transmission Terminal, GA27-3005.*

Transmitting Input

1. Turn on the power switch.
2. Set mode switch first to OFFLINE, then to TSM (transmit) or TSM TRSP (transmit transparent) - if available. Transparency is required if you want to submit EBCDIC data as object decks with your jobs, or if you expect to receive such data. Do not change from transparent to non-transparent mode during the same session.
3. Insert your card:

   ```
   *. . RJSTART termid,[BRDCST],2780,,[NOPUNCH]
   ```

   followed by:

   ```
   *. . LOGON userid
   ```

   and your input cards into the card reader.
4. Ready the printer.
5. If you expect punched output and the Automatic Turnaround feature is available, place blank cards behind your input and press AUTO TURNAROUND.

   Automatic Turnaround

   The Automatic Turnaround feature enables the terminal to switch automatically to receive mode and ready the punch for the output to be received without operator intervention. The card reader automatically changes to a punch-ready status after sensing the first blank card that follows the data cards.

   If you expect punched output and the Automatic Turnaround feature is not installed, ignore 5 and go to 6.
6. If you expect printed output or punched output and the Automatic Turnaround feature is not installed, press END-OF-FILE if no more cards are to be transmitted. (This turns on the END-OF-FILE light.)
7. If the connection is via a switched line, dial the appropriate number and switch the data set to data mode. The actual procedure depends on the type of data set at your location, and can normally be found in the operator's instructions for the data set.
8. Press START twice. The READY light turns on and the data will now be transmitted.
9. The alarm sounds (and the READY light turns off) when:

   a) a blank card has been sensed by the Automatic Turnaround feature.

   b) the last card in the hopper has been read and transmitted. If END-OF-FILE was pressed to indicate that this was also the last card of the job stream, transmission ends after the last card has been transmitted. If END-OF-FILE was not pressed

Resume input operation

because more cards had to be transmitted, refill the hopper and press START to resume operation.

c) Upon receiving an RVI (reverse interrupt) signal from the CPU, the 2780 has terminated transmission by sending an EOT (end of transmission) signal.

10. Press STOP to turn the alarm off (the alarm is also turned off when the terminal receives output from the central system). This removes the card punch and the printer from the ready state. Ready the device again.

Carriage Control Limitation

The range of carriage control functions for the IBM 2780 is smaller than that provided for local printers. Specifically, the IBM 2780 does not provide skips past channel 8, nor space suppression. When the IBM 2780 is used for remote output, these carriage control functions should be avoided. If they are requested, they will result in a single space.

Receiving Output

1. a) To receive punched output, place blank cards in the hopper (if the Automatic Turnaround feature is not installed or not activated).
   b) Set the mode switch to REC (if the Automatic Turnaround feature is not installed or not activated). Do not ready the printer.
   c) Press START on the card read/punch.
2. a) To receive printed output, the mode switch may be set to TSM, TSM TRSP, or REC.
   b) Press START on the printer.
3. If it is necessary to submit more input while output is in progress, discontinue output (see *Discontinuing Output* later in this chapter).
4. The end of output is signaled by the audible alarm (unless the reader is in the ready state), at which time more input may be submitted (see *Preparing More Input* later in this chapter).
5. Press STOP to turn the alarm off (the alarm is also turned off when the terminal receives more output from the central installation). This removes the card reader and the printer from the ready state. Ready the device again.

Discontinuing Output

1. Press STOP if you want to discontinue output.
2. After output has been discontinued, ready the printer; the central system will send you a message requesting a CONTINUE command.
3. Enter a CONTINUE command (see *Reference Information*). If the next card submitted is not a CONTINUE command, the central system

will flush the reader and write a message, stating that the job stream has been flushed.

4. In the CONTINUE command you may specify (with the HOLD operand) that the job output is to be held until requested via an OUTPUT command at some later time. In the meantime you may submit more input. For the other options of the CONTINUE command, see *Reference Information.*

Special Forms Output

1. If the programmer, as is recommended for RJE, has specified forms or cards in his * $$ PRT or * $$ PUN statement and the output is ready to be transmitted the central system will send you a message indicating required form or card number.
2. If you expect printed output, place the required forms in the printer and enter a CONTINUE command. Upon receipt of this command the central system transmits the output.
3. a) If you expect punched output, any cards in the card reader are flushed. EOF must be pressed.
   b) If the Automatic Turnaround feature is available, place a CONTINUE command and the required blank cards in the hopper. Press AUTO TURNAROUND and START.
   c) If the Automatic Turnaround feature is not installed, make sure that the printer is not in the ready state, place a CONTINUE command (without operand) in the hopper, press END-OF-FILE once and START twice. After the card has been read, put the required blank cards in the hopper, set the mode switch to REC, and press START on the reader. When punching has been started, press START on the printer.

Preparing More Input

1. While output is being written on the printer, you may load and ready the card reader to send more input (unless Automatic Turnaround is active). Wait until the printer has finished.
2. a) If the output is being received at the punch (or the printer if Automatic Turnaround is active, or if the mode switch is set to REC), wait until the output has been completed.
   b) Remove punched output and blank cards (if any).
   c) Reload and prepare the reader to transmit input. (The reader must be readied within approximately 30 seconds multiplied by the number of retries specified in the RETRY generation option to prevent the central system from disconnecting the line.)
3. Set the mode switch to either TSM or TSM TRSP and continue as described under *Transmitting Input* earlier in this chapter.

Error Recovery Procedures

When the central system detects a permanent line I/O error, it will usually disconnect your terminal. To reestablish the line connection, redial (for a switched line), and resubmit RJSTART and LOGON commands. Issue the STATUS command to get your input acknowledged and resubmit the interrupted job entry.

Failure During Input

When the card reader fails (because of a card jam, for example), correct the condition according to the error-recovery procedures defined in *IBM 2780 Data Transmission Terminal, GA27-3005.*

Whenever the terminal stops transmitting input, a time-out occurs after 30 seconds. The system will retry the read operation as many times as is specified in the RETRY system generation option; the default is six retries, which amounts to three minutes.

If the retries are unsuccessful, the user session is terminated and the terminal is disconnected from POWER RJE.

To resume input operations, redial (if the connection is via a switched line) and resubmit an RJSTART and a LOGON card. Issue the STATUS command to get the input acknowledged and resubmit the interrupted input.

Automatic Answering feature

Since the Automatic Answering feature, if active on a switched line, disconnects the line if recovery is not made within 21 seconds, it should be disabled during RJE operation.

Input Stream Flushed

Error messages

Upon detection of an invalid command or a user error, the central system will return one of the following messages to the terminal (see also *DOS/VS Messages, GC33-5379):*

1R65A	INVALID/MISSING TERMID/USERID
1R63A	INVALID RJE WORK STATION COMMAND
1R64A	INVALID RJE WORK STATION COMMAND - NO USER LOGGED ON
1R74A	TWO userid USERS ARE ALREADY LOGGED ON
1R53A	CARD FORMS formsid NEEDED FOR JOB ENTRY name. CONTINUE COMMAND REQUIRED
1R54A	CONTINUE COMMAND IS REQUIRED FOR PRINTER/PUNCH
1R70A	PRINT FORMS formsid NEEDED FOR JOB ENTRY name. CONTINUE COMMAND REQUIRED

After this the remaining cards in the hopper are flushed to enable you to correct the error. This is indicated by the additional message

1R67I	JOB STREAM FLUSHED

When you receive an error message on the printer as a result of an input error, adopt the following procedure:

a. Press STOP.
b. Press NPRO for a non-process runout of the card reader.
c. Locate the card in error, as indicated by the message.
d. Correct the invalid command.
e. Load the corrected card and the remaining cards into the reader.
f. Turn the mode switch to OFFLINE and then back to TSM TRSP or TSM (to drop ready status on the printer and the punch).
g. Press START on the printer.
h. Press END-OF-FILE.
i. Press START twice.

Transmission will then continue.

Failure During Output

Failure during output may occur as a result of a failing printer or punch, a device that is not ready, or failure of the central system.

Unit Failure

When the printer or punch fails, correct the condition according to the error-recovery procedures defined in *IBM 2780 Data Transmission Terminal, GA27-3005.*

The system will retry the output operation as many times as is specified in the RETRY generation option; the default is six retries.

If you can ready the printer or punch within 30 seconds multiplied by the number specified in the RETRY option, submit a CONTINUE command (see *Reference Information*), and the output will be resumed according to the options specified in the CONTINUE command. (If punched output is to be received and the Automatic Turnaround feature is being used, blank cards must follow the CONTINUE command to trigger Auto Turnaround.)

If the RETRY count is exhausted before recovery can be made, the terminal is disconnected from POWER RJE. To resume output operation, redial (if the connection is via a switched line) and resubmit the RJSTART AND LOGON commands.

The output of the interrupted job entry will be retransmitted upon receipt of an OUTPUT command.

If the unit failure causes an irrecoverable error, the output is placed in the hold state. Submit an OUTPUT command in a later session (see *Reference Information*) to retrieve the output. (The central operator may use the O (output) command to redirect output for a terminal to a local device.)

Unit Not Ready

If the central system tries to transmit to a device that is not in the ready state, the audible alarm sounds, and the TERM ADD light comes on (see *Error-Recovery Procedures* in *IBM 2780 Data Transmission Terminal, GA27-3005*). If the punch is not ready when initially selected, place blank cards in the hopper, turn the mode switch to REC and press START to initiate punch output.

If you have a terminal without a card punch (NOPUNCH parameter was specified in the RJSTART card), or you are operating in non-transparent mode, and punch output is to be sent to the terminal, the output is redirected for local punching and the following information message is listed on the printer:

```
1R79I jobname DIRECTED TO LOCAL PUNCH
```

Central Failure

If the central system fails while output is in progress, the terminal is placed in the inactive state. When the central system comes back online, submit an RJSTART command followed by a LOGON command, and retrieve the output using an OUTPUT command.

If the central system fails while input is in progress, the terminal is placed in the inactive state. Submit an RJSTART and a LOGON command when the system comes back online. Begin the interrupted user session with a STATUS command to obtain acknowledgement of jobs entered prior to the central system failure. When a failure occurs, the central system will attempt to notify you by sending the message

```
1R51I     ABNORMAL CENTRAL SHUTDOWN
```

Testing the RJE System

Online Terminal Test

If you want to test the system to check whether transmissions are being received, use the Online Terminal Test facility. This is an optional service provided by BTAM to ensure proper operation of the system, and it may be used in the diagnosis and correction of a terminal malfunction. For details refer to *DOS/VS Basic Telecommunications Access Method, GC27-6989.*

If the online test facility is not present, you may send a message (via an MSGR command) to the central operator requesting him to send you a broadcast message. If you receive it on the printer upon submitting a BRDCSTR command, the system may be assumed to be working properly.

3780 Terminal Operation

19

This procedure describes how to operate a 3780 terminal.

Detailed information about the 3780 is contained in the *Component Information for the IBM 3780 Data Communication Terminal, GA27-3063.*

Transmitting Input

1. Turn on the power switch at the control unit. If Power is already on, set OFFLINE switch to OFFLINE, then press terminal reset.
2. Set the Term Mode switch on the console to LINE.
3. Set the Answer switch on the console to MANUAL.
4. Turn the Inquiry Mode switch on the console off.
5. Turn AUTO RESTART switch on.
6. At the card punch: remove cards, press NPRO, then place blank cards in the hopper and press START.
7. Press CARRIAGE RESTORE on the printer to align forms and press START.
8. Press NPRO (Non-process Runout) on the card reader to flush cards.
9. Place the input cards (preceded by your * . . RJSTART termid,[BRDCST],3780,,[NOPUNCH] and your * .. LOGON userid cards) in the reader.
10. Press CHECK RESET and TERM RESET on the console.
11. If transparent data is to be transmitted or received, turn on the TRNSPCY switch on the console.
12. If the space compress feature is available, turn on the SPACE COMPRESS EXPAND switch on the console. Note that this feature is only active in non-transparent mode.
13. Set the EOF (End-of-File) switch to ON (if no more cards are to be transmitted) and press START on the reader panel to ready the card reader.
14. In case of a switched line, dial the number and switch the data set to data mode. The actual procedure depends on the type of data set at your location, and can normally be found in the operator's instructions for the data set.
15. Press START on the console to transmit data.
16. The alarm sounds when:

    a) the last card in the hopper has been read and transmitted. If the EOF switch was set to ON to indicate that this was also the last card of the input stream, transmission ends normally. If the EOF switch was left in the OFF position because more cards had to be transmitted, refill the hopper, set the EOF switch to ON if this is the last group of cards for this job and press START on the card reader and then on the console to resume the input operation.
    b) If a message from the central installation is to be printed during input, your terminal receives an RVI (reverse interrupt)

signal from the CPU. The PROCSR IRPT (Processor Interrupt) light is turned on at the console when line turnaround is triggered by RVI. When the AUTO RESTART switch is on, the processing of input resumes automatically after the message is printed.

Receiving Output

1. When print or punch output is available to be received at the terminal and the output device is not in the ready state, the alarm will sound.
2. Correct error indicators on the printer and/or punch according to error-recovery procedures in *Component Information for the IBM 3780 Data Communication Terminal, GA27-3063.*
3. Press CHECK REQUEST and START on the console to return the terminal to ready state.
4. If it is necessary to submit more input while output is in progress, discontinue output (see *Discontinuing Output* later in this section).
5. The end of output is signaled by the audible alarm, at which time you may submit more input (see *Preparing More Input* later in this section.

Discontinuing Output

1. Press CARRIAGE STOP on the printer or card punch control panel if you want to discontinue print or punch output in progress.
2. When output has been discontinued, ready the printer or card punch again, upon which the central system will send you a message requesting a CONTINUE command.
3. Enter a CONTINUE command (see *POWER RJE Terminal Commands*). If the next card submitted is not a CONTINUE command, the central system will flush the reader and write a message stating that the job stream has been flushed.
4. In the CONTINUE command you may specify (with the HOLD operand) that the job output is to be held until requested via an OUTPUT command at some later time. In the meantime you may submit more input. For the other options of the CONTINUE command, see Reference Information.

Special Forms Output

1. If the programmer, as recommended for RJE, has specified special forms or cards in his * $$ PRT or * $$ PUN statements and the output is ready to be transmitted, the central system will send you a message indicating the required form or card number.
2. For printed or punched output, place the required forms or cards in the printer or punch and transmit a CONTINUE command from the reader. Upon receipt of this command the central system transmits the output.

Preparing More Input

1. While output is printed or punched, you may load and ready the reader to send more input. (If you ready the reader after output has been completed, you must do so within approximately 30 seconds multiplied by the number of retries specified in the RETRY generation option to prevent the central system from disconnecting the line.)
2. When print or punch output has completed, input will be transmitted.

Error Recovery Procedures

When the central system detects a permanent line I/O error, it will usually disconnect your terminal. To reestablish the line connection, redial (for a switched line), and resubmit RJSTART and LOGON commands. Issue the STATUS command to get your input acknowledged and resubmit the interrupted job entry.

Failure During Input

When the card reader fails (because of a card jam, for example), correct the condition according to the error-recovery procedures defined in *Component Information for the IBM 3780 Data Communication Terminal, GA27-3063.*

The terminal maintains the connection with the central system as long as you press STOP on the card reader at intervals of not longer than 30 seconds (by sending repeated WACKS). Otherwise a timeout occurs after 60 seconds, followed by another after 30 seconds.

The system will retry the read operation as many times as is specified in the RETRY system generation option. The default is six. Each retry takes 30 seconds.

If the retries are unsuccessful, the user session is terminated and the terminal is detached from POWER RJE.

To resume input operations, redial (if the connection is via a switched line) and resubmit the RJSTART and LOGON cards. Issue the STATUS command to get your input acknowledged and resubmit the interrupted job entry (and any other unacknowledged input).

Automatic Answering feature

Since the Automatic Answering feature, if active on a switched line, disconnects the line if recovery is not made within 21 seconds, it should be disabled during RJE operation.

Input Stream Flushed

Error messages

Upon detection of an invalid command or a user error, the central system will return one of the following messages to the terminal (see also *DOS/VS Messages, GC33-5379):*

1R65D	INVALID/MISSING TERMID/USERID

1R63D	INVALID RJE WORK STATION COMMAND
1R64D	INVALID RJE WORK STATION COMMAND - NO USER LOGGED ON
1R74D	TWO userid USERS ARE ALREADY LOGGED ON
1R53A	CARD FORMS formsid NEEDED FOR JOB ENTRY name. CONTINUE COMMAND REQUIRED
1R54A	CONTINUE COMMAND IS REQUIRED FOR PRINTER/PUNCH
1R70A	PRINT FORMS formsid NEEDED FOR JOB ENTRY name. CONTINUE COMMAND REQUIRED

After this the remaining cards in the hopper are flushed to enable you to correct the error. This is indicated by the additional message

1R67I	JOB STREAM FLUSHED

When you receive an error message on the printer as a result of an input error, adopt the following procedure:

a. Press STOP on the card reader.
b. Lift the telephone handset and switch the data set to talk mode (switched line only).
c. Press NPRO on the card reader.
d. Locate the error card and correct it.
e. Load the corrected card and the remaining cards into the reader.
f. Press CHECK RESET and TERM RESET on the console.
g. Press START on the card reader.
h. Switch the data set do data mode and hang up.
i. Press START on the console to continue transmission.

Failure During Output

Failure during output may occur as a result of a failing printer or punch, a device that is not ready, or failure of the central system.

Unit Failure

When the printer or punch fails, correct the condition according to the error-recovery procedures defined in *Component Information for the IBM 3780 Data Communication Terminal, GA27-3063.*

During recovery procedures the central system will not detect this situation if you press the STOP key at intervals of not longer than 30 seconds (the terminal will maintain the connection with the central system by transmitting WACKs). Otherwise the system will retry the output operation every 30 seconds until it is successful or until the retry count is exhausted. When the retry count is exhausted, a message is issued requesting a CONTINUE command.

The system retries the output operation as many times as is specified in the RETRY generation option; the default is six retries.

If you can ready the printer or punch within 30 seconds multiplied by the number specified in the RETRY option, submit a CONTINUE command (see *POWER RJE Terminal Commands*), and the output will be resumed according to the options specified in the CONTINUE command.

If the RETRY count is exhausted before recovery can be made, the terminal is disconnected from POWER RJE. To resume output operation, redial (if the connection is via a switched line) and resubmit the RJSTART and LOGON commands. The output of the interrupted job entry will be retransmitted upon receipt of an OUTPUT command.

If the unit failure causes an irrecoverable error, the session is ended, and the output is placed in the hold state. Submit an OUTPUT command in a later session (see *POWER RJE Terminal Commands*) to retrieve the output. (The central operator may use the O (output) command to redirect output for a terminal to a local device.)

Unit Not Ready

If the central system tries to transmit to a device that is not in the ready state, the audible alarm sounds, and the TERM ADD light comes on (see *Error-Recovery Procedures* in *Component Information for the IBM 3780 Data Communication Terminal, GA27-3063.*

If a print error occurs, you can normally place the printer in the ready state by pressing CHECK RESET and TERM RESET on the console and START on the printer.

If a punch error occurs (because of a card jam, for example), remove the cards that caused the failure. Place the punch in the ready state by pressing CHECK RESET and TERM RESET on the console.

If you have a terminal without a card punch (if NOPUNCH parameter was specified in the RJSTART card), or you are operating in non-transparent mode, and punch output is to be sent to the terminal, the output is redirected for local punching and the following information message is listed on the printer:

```
1R79I jobname DIRECTED TO LOCAL PUNCH
```

Central Failure

If the central system fails while output is in progress, the terminal is placed in the inactive state. When the central system comes back online, submit an RJSTART command followed by a LOGON command, and retrieve the output, using an OUTPUT command.

If the central system fails while input is in progress, the terminal is placed in the inactive state. Submit an RJSTART and a LOGON command when the system comes back online. Begin the interrupted user session with a STATUS command to obtain acknowledgement of jobs entered prior to the central system failure. When a failure occurs, the central system will attempt to notify you by sending the message

```
1R51I     ABNORMAL CENTRAL SHUTDOWN
```

Testing the RJE System

Online Terminal Test

If you want to test the system to check whether transmissions are being received, use the Online Terminal Test facility. This is an optional service provided by BTAM to ensure proper operation of the system, and it may be used in the diagnosis and correction of a terminal malfunction. For details refer to *DOS/VS Basic Telecommunications Access Method, GC27-6989.*

If the online test facility is not present, you may send a message (via an MSGR command) to the central operator, requesting him to send you a broadcast message. If you receive it on the printer upon submitting a BRDCSTR command the system may be assumed to be working properly.

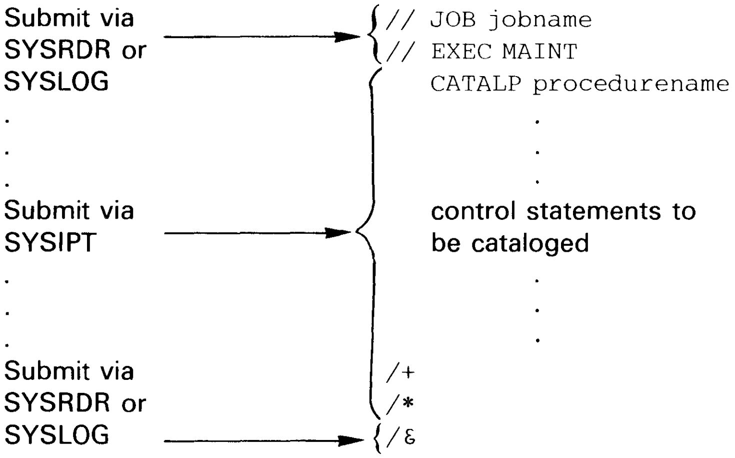
DOS/VS allows you to catalog procedures, that is sets of job control and linkage editor control statements with or without SYSIPT data in the procedure library. You can access this library whenever you need a procedure to complete or set up a job stream.

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).

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 important rush job in a partition which is specifically reserved for such jobs, and which is not working at present.
2. To insert an important 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

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

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.

Symbolic names must be unique.

To enable changes 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.

You cannot modify an unnamed statement.

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.

To determine the symbolic names of statements in a procedure, print the procedure on SYSLST (see Procedure 23).

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

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

blank 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.

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

```
// ASSGN SYS005,X'193'   NAME4
```

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

```
// ASSGN SYS005,X'193'   NAME4     D
                         ↑         ↑
                         column 73 column 80
```

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
```

Examples

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:

20

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

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

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

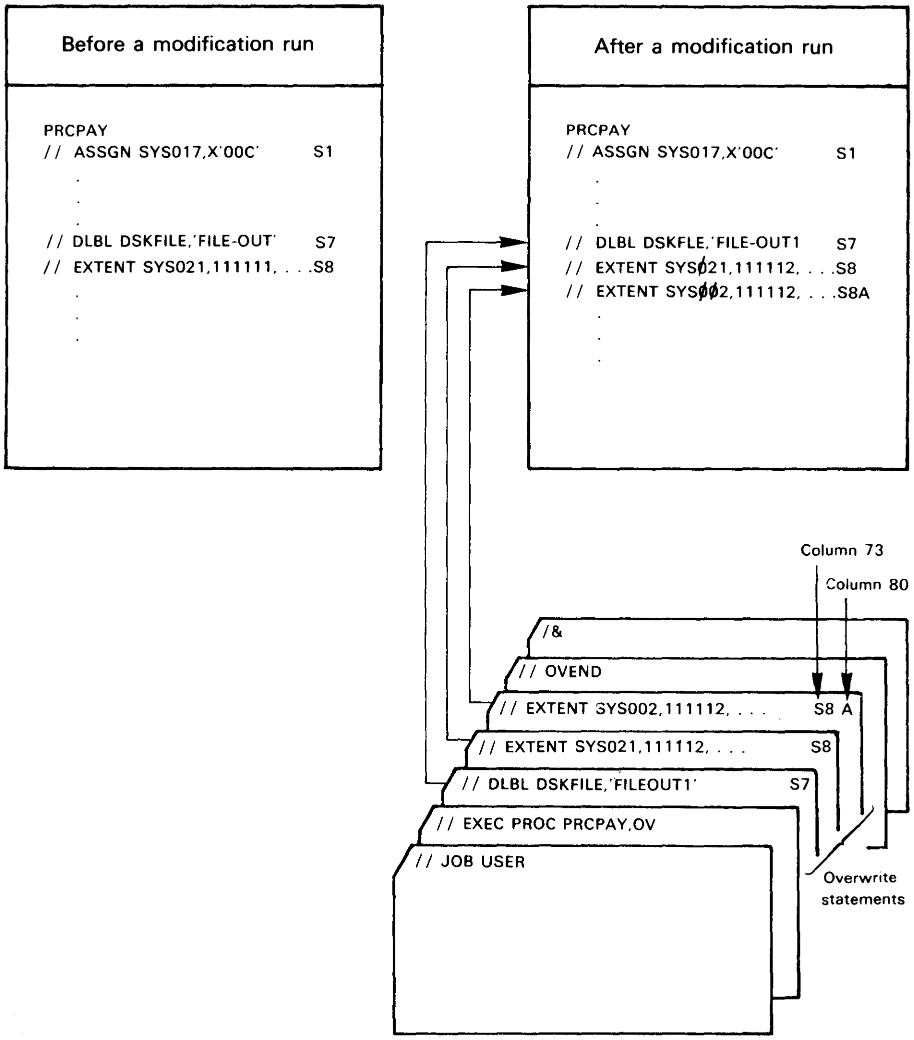

Before a modification run
PRCPAY
// ASSGN SYS017,X'00C' S1
// DLBL DSKFILE,'FILE-OUT' S7
// EXTENT SYS021,111111, . . .S8
After a modification run
PRCPAY
// ASSGN SYS017,X'00C' S1
// DLBL DSKFLE,'FILE-OUT1 S7
// EXTENT SYS021,111112, . . .S8
// EXTENT SYS002,111112, . . .S8A
Column 73
Column 80
/&
// OVEND
// EXTENT SYS002,111112, . . . S8 A
// EXTENT SYS021,111112, . . . S8
// DLBL DSKFILE,'FILEOUT1' S7
// EXEC PROC PRCPAY,OV
// JOB USER
Overwrite statements

Figure III-2. **Example of overwrite statements.**

Displaying the VTOC

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 the card reader
2. Using the printer-keyboard or DOC

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

Using the card reader

```
// 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 DOC

To display the VTOC from the printer-keyboard or DOC, 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

22

The 'operator action' for a number of DOS/VS messages includes the listing of the label information cylinder 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, you use the LSERV program. You can do this in one of two ways:

1. Using the card reader
2. Using the printer-keyboard or DOC

LSERV prints the contents of the label information cylinder on the device assigned to SYSLST.

Using the card reader

The job control statements required to execute the LSERV program from a card reader are:

```
// JOB name
// EXEC LSERV
/*
/&
```

Using the printer-keyboard or DOC

To execute the LSERV program from the printer-keyboard or DOC, enter

```
// EXEC LSERV
```

If you want to display the label information cylinder 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

23

Recommendations on when to use these programs are given in *DOS/VS SADP, GC33-5380.*

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.

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 cards, be sure to replace the words phase1, module1, etc., by the actual phase or module names.

Core image library

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 either DSPLY or DSPCH. DSPLY causes the output to be printed; DSPCH causes the output to be printed and punched in cards.

Displaying and Punching the Libraries (continued)

23

Examples

To punch the phases CORPHA and CORPHB from the core image library, enter

```
// JOB PUNCHPH
// EXEC CSERV
   PUNCH CORPHA,CORPHB
/*
/&
```

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.

Procedure library

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 in cards.

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
/*
/&
```

Relocatable library

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 in cards.

Examples

To punch the modules MODA and MODC from the relocatable library, enter

```
// JOB PCHMOD
// EXEC RSERV
   PUNCH MODA,MODC
/*
/&
```

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)

23

Source Statement Library

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 in cards.

Function	Control Statements Required

Punch one or more books	`// JOB jobname`
	`// EXEC SSERV`
	`   PUNCH sublib.book1[,sublib.book2,...]`
	`/*`
	`/&`

Punch one or more sub- libraries	`// JOB jobname`
	`// EXEC SSERV`
	`   PUNCH sublib1.ALL[,sublib2.ALL,...]`
	`/*`
	`/&`

Punch entire library	`// JOB jobname`
	`// EXEC SSERV`
	`   PUNCH ALL`
	`/*`
	`/&`

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)

24

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	// 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, relocatable, and 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.
- The **procedure library** can be condensed from the background partition unless
  - it is being accessed by the job control program
  - 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.

Automatic Condense

You can also specify that the condense function be performed automatically each time the available free space in a library drops below a specified minimum, referred to as the condense limit. Automatic condense is requested by the CONDL control statement indicating the library or libraries to be condensed and the condense limit(s).

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.

NEVER CANCEL A CONDENSE RUN

If a condense run is interrupted by a hardware error or by operator intervention, the library being condensed is inusable and must be reconstructed.

25 Your programmer will tell you for which libraries automatic condense is to be established and the condense limits (in number of library blocks) to be specified.

Example:

```
// JOB  NAME
// EXEC MAINT
   CONDL CL=10
/*
/&
```

The CONDL statement in the above example indicates that the core image library is to be condensed automatically 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 DOC 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.

To print the hard copy file, proceed as follows:

1. Type `// JOB name` and press ENTER.

2. Type `// EXEC PRINTLOG` and press ENTER.

   The following message appears on the screen:

   ```
   ENTER ONE OR MORE OF THE FOLLOWING OPTIONS:
   ALL OR NEW, BG OR F1 OR F2 OR F3 OR F4 OR AR,
   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.

3. 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 on the hard-copy file.

   NEW — specifies printing of the messages that have accumulated since the previous run of PRINTLOG.

   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.

   BG - F4 — specifies printing of messages issued by a particular partition; only **one** partition identifier may be specified.

   AR — specifies printing of messages issued by the Attention routine.

   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.

26

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

28

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.*

Debugging aids
Alter/Display
Trace
Dump
EREP

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.

Shutting down the System

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 an IBM System/370 Model 115 or 125, 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.

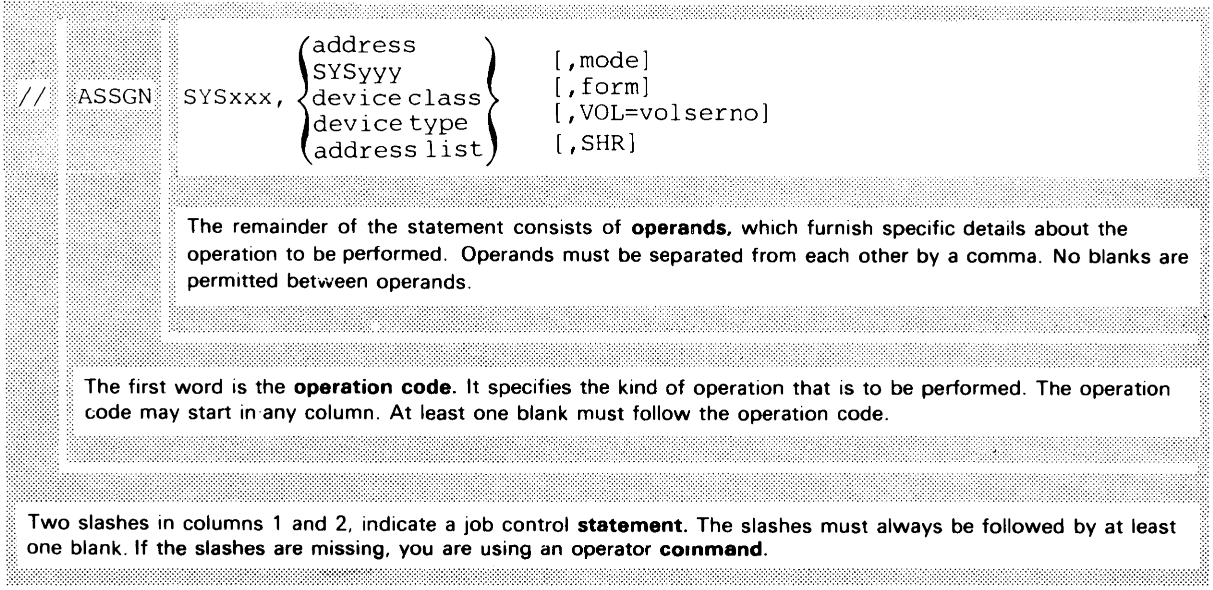
Reference Information

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. We use the ASSGN statement/command to illustrate these rules (see Figure IV-1).



```
                  (address      )   [,mode]
                  )SYSyyy       (   [,form]
// ASSGN SYSxxx, <device class   >  [,VOL=volserno]
                  )device type  (   [,SHR]
                  (address list )
```

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,

```
 (address      )
 )SYSyyy       (
<device class   >
 )device type  (
 (address list )
```

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 SYSREC,X'191'
// ASSGN SYSRDR,X'193',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.

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 CANCEL command:

$$\text{CANCEL}\ \begin{bmatrix}\underline{\text{BG}}\\ \text{Fn}\end{bmatrix}$$

Decimal values

For the letter n, you have to specify a decimal value which, in this specific case, may be either 1, 2, 3, or 4, indicating one of the four foreground partitions.

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 CANCEL 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 canceled by default.

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-6, which shows all DOS/VS job control statements, is followed by a description of each job control statement.

Operator Commands

There are three types of operator command:

- IPL commands
- job control commands
- attention commands

The main difference between these 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 either from the console printer-keyboard, from the DOC, or from a card reader.

When entering IPL commands from the **console printer-keyboard,** press END after every command. When you press END after the last command, the system issues the message:

DOS/VS IPL COMPLETE

You may then enter job control commands or statements.

When entering IPL commands from the **DOC,** press ENTER after every command. The procedure is the same as described for the Console Printer-Keyboard, but the name of the key you press is ENTER instead of END.

When all IPL commands have been entered from a **card reader,** the system issues the message

DOS/VS IPL COMPLETE

as described above for the console printer-keyboard and for the DOC.

Then, however, job control immediately reads the job control statements/commands, if any, that follow the IPL commands, and initiates the first job in the background partition.

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 DOC, or from a card reader.

When entering job control commands from the console printer-keyboard or from the DOC, press END or ENTER, respectively, after typing in a command.

Job control commands to be read from a card reader are simply placed in the job stream on the card reader, i. e., 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 **DOC.** Before entering an attention command from the printer-keyboard, press REQUEST. The system responds by issuing the message

READY FOR COMMUNICATION

Then type in the command and press END or ENTER, respectively.

Attention commands from the DOC 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.

If you enter an attention command or job control command without specifying a partition in the operand, the partition in which the command is issued will be addressed. This is always the partition identified by the most recent partition identifier on SYSLOG.

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
LISTIO	List current I/O assignments		X	
LOG	Log (print) job control statements and commands		X	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 (part 1 of 2). Alphabetical list of all operator commands.

Command	Meaning	Type of Command		
		IPL	Job Control	Attention
MSG*	Communicate with a foreground program			X
MTC	Perform magnetic tape control operations		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	
UCS	Load Universal Character Set buffer		X	
UNBATCH*	Deactivate a partition		X	

*   only if your system has more than one partition

** date and clock only

Figure IV-2 (part 2 of 2). Alphabetical list of all operator commands.

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.

ADD
add a device to the PUB table

See also Procedures 2, 3, and 4

Operation	Operands	Type
ADD	X'cuu'[k],devicecode [,X'ss' ,X'ssss' ,X'ssssss']	IPL

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.

devicecode 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	
1403	1403 Printer	Printers
1403U	1403 Printer with UCS feature	
1443	1443 Printer	
3211	3211 Printer	
3525P	3525 Card Punch (with optional print feature)	
3203	3203 Printer	
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 Units
125DP	Model 115 or 125 integrated Display Operator Console with attached 5213 console printer	
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	
3340	3340 Disk Storage	
3540	3540 Diskette I/O Unit	Diskette

Figure IV-3 (part 1 of 2). Device codes used in the ADD command.

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	
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	
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 Printer	
3277B	3277 Display Station or 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	
2955	2955 Data Adapter Unit	Data Link for RETAIN
7770	7770 Audio Response Unit	Audio Response Unit
UNSP	Unsupported Device	No burst mode on multiplexor channel
UNSPB	Unsupported Device	Burst mode on multiplexor channel

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

ALLOC
allocate virtual storage

See also Procedure 11

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.

Operation	Operands	Type
ALLOC	[F1=nK][,F2=nK][,F3=nK][,F4=nK]	AR, JC

The minimum of storage allocated to an active (batched) virtual foreground partition is 64K. The value n must always be an even number. 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. 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 11

The ALLOCR command allocates storage to real partitions (or modifies the amount of storage allocated to real partitions).

Operation	Operands	Type
ALLOCR	[BGR=nK][,F1R=nK][,F2R=nK][,F3R=nK][,F4R=nK]	AR, JC

n (the amount of storage allocated to a real partition) must be an even number. It may also be zero.

The ALLOCR command must have at least one operand. 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 10

The ASSGN command assigns a logical unit name to a physical I/O device. It can be used to modify any assignment previously established. The assignment established via an ASSGN command is normally permanent, i.e. remains in effect until the next ASSGN command or the next IPL procedure, unless it is overridden for the duration of the job or job step by an ASSGN job control statement. An exception is if the TEMP option is included.

Operation	Operands	Type
ASSGN	address SYSyyy SYSxxx, device class [,mode][,form] device type [,VOL=volserno][,SHR] address list	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[1,2]
  SYSIN
  SYSIPT
  SYSLNK
  SYSLOG
  SYSLST
  SYSOUT[2,3]
  SYSPCH
  SYSRDR
  SYSREC[2]
  SYSRLB
  SYSSLB

  [1] Only if the system permits private core image libraries.
  [2] Must be permanent assignments, i.e. the TEMP option is not permitted for these logical units.
  [3] Valid for a tape unit only.

- 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.

If a system logical unit is assigned to a diskette, a tape or DASD, it must be closed (using the CLOSE command) before it can be reassigned. A system or programmer logical unit must be permanently unassigned before it can be assigned to another partition (e.g., from BG to F3).

address the address can be the hexadecimal device address (X'cuu'), UA, or IGN:

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

ASSGN (continued)

UA (UnAssigned) indicates that the logical unit is to be unassigned. Any operation attempted on an unassigned 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). If you use one of these words, the system will search for an available device in that category. A successful scan is completed by message 1T20I, informing you of the assigned device address.

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. The system will search for an available device of the desired type. When found, message 1T20I is issued, telling you to which device address the symbolic unit has been assigned.

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.

mode can be ALT, X'ss', H1, or H2.

ALT indicates an alternate magnetic tape unit that is used when the capacity of the original assignment is reached. 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.

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

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). H2 must also not be specified if the 2560 or 5425 is assigned to a partition supported by POWER.

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.

form

indicates the nature of the assignment and can be either TEMP (temporary) or PERM (permanent). If included, this option overrides any other given assign form, standard assignment as well as the // indication. A temporary assignment is effective only for the current job, and is reset to the last permanent assignment upon job termination.

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 you had not yet mounted the disk pack, you would receive message 1T50A, telling you to mount pack 222222 on the indicated drive.

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 want to use a disk which is already owned by a partition.

Density (b.p.i.)	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.

BATCH
initiate or continue processing

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	[BG Fn]	AR

See also Procedure 12

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

INVALID STATEMENT

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 specified partition. The job will be canceled as soon as all outstanding interruptions have been handled.

Operation	Operands	Type
CANCEL	[BG Fn]	AR
CANCEL	blank	JC

The CANCEL command without operand can be used as a job control command or as an attention command. If it is used as a job control command, it cancels the job in the corresponding partition. If it is used as an attention command, it cancels the job in the background partition.

If CANCEL is used as an attention command and an operand is specified (BG, F1, F2, F3, or F4), it cancels the job in the specified partition.

The CANCEL command must not be used to cancel the partition in which POWER is running.

CAT
assign logical unit for VSAM catalog

See also Procedures 2 and 3

The Virtual Storage Access Method (VSAM) uses a catalog that describes all VSAM files and indexes.

This catalog must be available to the system whenever VSAM is used.

The CAT command serves to assign the system logical unit SYSCAT, on which the VSAM 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 catalog volume. SYSCAT may only be assigned to a 2314, 2319, 3330, 3333, or 3340.

CLOSE
close a logical unit

See also Procedure 13

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]	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.

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, 3, and 4

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.

DEL X'130'

DV [illegible] 2314

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 the definition has been changed in the meantime. 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

Detailed information on dumps is contained in *DOS/VS SADP, GC33-5380*

The DUMP command allows you to print part or all of the contents of the real and/or virtual address area on SYSLST. SYSLST may be assigned to any partition; it must be a printer. The printer should not, at the time of the dump, be used since this would result in interspersed partition and dump output.

Operation	Operands	Type
DUMP	{ blank S BG Fn BGS FnS PDAREA addr,addr } [,BG ,Fn]	AR

blank — if the dump area operand is omitted, 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.

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.

in first operand field

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

DUMP (continued)

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.

in second operand field — BG/Fn — 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 20 and 27

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] $\left[\text{,SIZE=}\left\{\begin{matrix}\text{nk}\\\text{AUTO}\\\text{(AUTO,nk)}\end{matrix}\right\}\right]$	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.

(a) 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.

(b) 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).

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	[F1][,F2][,F3][,F4]	JC

You may enter from one to four operands 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

The IGNORE command has no operand.

LISTIO
list current I/O assignments

See also Procedure 10

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 10.

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

Operation	Operands	Type
LISTIO	ALL BG Fn PROG SYS SYSxxx DOWN UNITS X'cuu' UA	JC

ALL lists all logical units and the physical units assigned to each.

BG lists all logical units of the background partition and the physical units assigned to each.

Fn lists all logical units of the specified foreground partition (F1, F2, F3, F4) 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 13

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.

MAP
produce map of real and virtual storage

See also Procedure 11

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 concerning the MODE command are given in *DOS/VS SADP, GC33-5380.*

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	{ BG Fn }	AR

BG or Fn specifies the partition with which communication is desired (where Fn = F1, F2, F3, or F4).

MTC
magnetic tape control

The MTC (magnetic tape control) command enables the operator to perform tape operations such as rewinding, unloading, skipping files, etc. The MTC command is a useful tool to perform tape control operations from the console printer-keyboard or from the DOC.

Operation	Operands	Type
MTC	opcode, $\left\{\begin{matrix}\text{SYSxxx}\\ \text{X'cuu'}\end{matrix}\right\}$ [,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.

NOLOG
suppress logging

See also Procedure 13

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 JOB, PAUSE, STOP, ALLOC, ALLOCR, MAP, DVCDN, DVCUP, /*, and / &, nor of control statements in error.

Operation	Operands	Type
NOLOG	blank	JC, AR

The NOLOG command has no operand.

OVEND
end of overwrite statements

See also Procedures 20 and 27

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 14

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, etc.

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 DOC. The PAUSE command is therefore ignored if no console printer-keyboard or DOC is available.

Operation	Operands	Type
PAUSE	[comments]	JC
PAUSE	BG [,EOJ] Fn [,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, and the partition in which processing is to be interrupted can be specified (BG, F1, F2, F3, or F4). If no partition is specified, the program in the background partition will be interrupted.

If the optional operand EOJ is specified, the interruption will occur at the end of the current **job.** 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	sequence	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. Sequence consists of the partition identifiers (BG, F1, F2, F3, F4) in 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.

RESET
reset I/O assignments

See also Procedure 10

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 5

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.

When and where DATE and CLOCK should be specified, depends on your system. For details refer to *DOS/VS System Control Statements, GC33-5376.*

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.

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.

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 multivolume capability there is no need for a warning message.

n5 defines the status of the system recorder file on SYSREC. It can be:

See also Procedure 6

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 reflect 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 7

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 SVA from that specified in the supervisor. The format of this option is SVA=(nK,nK). 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 size should be a multiple of 2K and smaller than the size of the SVA.

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.

START
initiate or continue processing

See also Procedure 13

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	[BG / Fn]	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 14

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.

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 8 and 9

A character sent to the printer for printing is compared with the characters in the UCS buffer. When a match occurs, the corresponding chain/train character is printed in the print-line position that was occupied by the output character. The UCS buffer and the variety of chains/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

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 upper-case or lower-case 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 14

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 are unassigned, and the partition is no longer active. Because logical units are unassigned, all tape and disk files must previously have been closed (via a CLOSE command).

The partition can subsequently be made active again (i.e., 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 are exactly the same as of the ASSGN job control command with the TEMP option.

Operation	Operands
// ASSGN	SYSxxx, { address SYSyyy device class device type address list } [,mode][,form] [,VOL=volserno][,SHR]

// 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, etc.

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.

// DLAB

The // DLAB job control statement should no longer be used. See the // DLBL job control statement.

The // DLAB job control statement is entered in the form of two cards. The format is:

Operation	Operand	Col. 72
// DLAB blank	'label-fields 1-3', xxxx,yyddd,yyddd,'system-code'[,type]	X

label-fields consists of 51 characters. The first operand in the second card must start in column 16.

// 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],[data-security]

// EXEC
execute a program or a procedure

See also Procedure 20

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 timer feature, the time of day appears in positions 73-100 when the // JOB statement is printed on SYSLST. The 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 characters; blanks, equal signs, and commas are not permitted.

Accounting information may contain 16 characters of user-specified job accounting information. 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	{ TAPE[(nn)] }
	{ NSD(nn) }

// LISTIO
list current I/O assignments

Format and effect of the // LISTIO job control statement are exactly the same as those of the LISTIO job control command.

Operation	Operands
// LISTIO	SYS PROG BG Fn ALL SYSxxx UNITS DOWN UA X'cuu'

// MTC
magnetic tape control

Format and effect of the // MTC job control statement are the same as those of the MTC job control command.

Operation	Operands
// MTC	opcode,SYSxxx[,nn]

// 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 and of the virtual or real partition on SYSLST in case of abnormal program termination.
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.

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.

// OPTION (continued)

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 used by programmers only.
NOSYM 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 used by programmers only.
NOXREF 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 DOC. The // PAUSE statement is therefore ignored if no console printer-keyboard or DOC 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	$\left\{\begin{array}{l}\text{SYS}\\ \text{PROG}\\ \text{ALL}\\ \text{SYSxxx}\end{array}\right\}$

// 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 (e.g., 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	filename,['file-id'],[date], ⎡file-serial-number⎤ ⎣set-identifier⎦ ⎡volume-sequence-number⎤ ⎣file-section-number⎦ ,[file-sequence-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 11-13'}

// 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 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	{EAST WEST} /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:

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

// ZONE (continued)

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.

POWER Commands

The POWER central installation console commands allow you to control POWER and POWER RJE operation from the central operator's console. They are entered in the same way as attention routine commands.

The POWER central installation console commands may be divided into the following four groups:

1. Commands for routine and system management:

S	start a routine
P	stop a routine
G	reactivate a routine or a partition
C	cancel a routine
F	flush a writer routine
M	display or alter copy counter
T	restart a writer routine
E	terminate or cancel POWER

2. Commands for queue management:

A	alter job-entry priority
D	display status of a job entry
H	hold a job entry
L	delete a job entry
R	release a job entry
J	process the accounting file

3. Commands for RJE management:

B	send, delete, or display messages
I	inquire about status of RJE routines
O	change output destination of a job entry
S	start an RJE routine
P	stop an RJE routine
G	reactivate an RJE routine
D	display the status of an RJE job entry

4. Command for diagnostic purposes:

Z	trace TIB, QFILE, or DATAFIL access by POWER

The following pages describe all the above commands in alphabetical order.

A (alter)

This command must not be used if POWER was generated with the option PRIORTY=NO

The A command alters the priorities of jobs in the queue.

Operation	Operands
A	queue,name,[number],priority

queue — specifies the queue to which the command is directed in the xxyyy format, where

xx	=	BG, F4, F3, F2, or F1
yyy	=	RDR, PRT, or PUN

name — specifies the name of the job entry.

number — specifies the number assigned to the job entry by POWER when it is inserted in the input queue. The job number may be obtained by using the D command. It may consist of up to five numeric characters and allows addressing of specific job entries with identical names.

priority — specifies the priority the job entry is to assume. It is specified by a numeric character from 0 to 9. Nine is the highest priority.

The A command changes the priority of a job entry after it has been inserted in the POWER queues. For a job entry in the input queue (RDR), the change determines when the job entry will be processed by DOS/VS. For output, the command changes the sequence in which the job entries are processed by the writer routines.

Examples

```
A   BGRDR,ASSEMBLY,016,3
A   F2PUN,OBJECT,,0
```

B (broadcast)

The B command enables the central installation operator to communicate with remote terminals. Messages may be submitted for one specific terminal, or for all terminals and are held in the message queue.

In the message queue, the messages are grouped and are given numbers within the groups. There is one group for each 'userid' and 'termid'. An additional group, called 'ALLUSERS' consists of messages addressed to all users.

The B command permits you to add a message to one group, to delete a message from one group, or to delete all messages of a group or all messages, and to display a message or all messages of a group on SYSLOG.

When a message is added, it is given the next higher number within its group. When a message is deleted, the messages of the group are renumbered. Therefore, the actual message numbers can be obtained only by displaying an entire group.

Operation	Operands	Type
B	M, termid userid ALLUSERS ,'msg'	Message option
B	L, termid [,nr] userid [,nr] ALLUSERS [,nr] ALL	Delete option
B	D, termid [,nr] userid [,nr] ALLUSERS [,nr]	Display option

M
L
D

specify the B command option desired. M causes the specified message to be added to the specified message group. L causes the specified message or all of the messages of a group or all messages to be deleted. D causes the specified message or all of the messages in a specified group to be displayed on SYSLOG.

termid
userid
ALLUSERS
ALL

specify the group in the RJE message queue to which the B command option applies. 'termid' and 'userid' are valid if they are entries in the userid list (see *Userid List Generation* in *DOS/VS System Generation, GC33-5377*), which is created when POWER RJE is generated. ALLUSERS specifies those messages to be broadcast to all users. ALL specifies the entire message queue.

B (broadcast)
(continued)

msg — specifies the message to be added to the specified group. The message may consist of from one to 40 characters, enclosed in single quotation marks. If the message text contains a quotation mark, type in two quotation marks for each that is to appear in the message.

Operator messages can be retrieved at the terminal by the

* . . RJSTART termid,BRDCST, . . .
and by the * . . BRDCSTR

commands. Messages addressed to a termid are deleted after delivery.

nr — specifies the number of a specific message in the RJE message queue. nr is specified as a number from 1 to 99. If this operand is omitted, all messages for the specified userid, termid, or ALLUSERS are assumed.

To perform its functions, the B command requires a data buffer. If a data buffer is not available when the B command is used, a message to this effect will appear on SYSLOG and the command will be ignored.

Examples

```
B  M,NEWYORK,'RJE CEASES AT 14.00 TODAY, 4/18/72'
```

This is a message to one specific work station.

`B  D,NEWYORK`	Display all messages to New York
`B  D,DENVER,3`	Display message No. 3 to Denver
`B  D,ALLUSERS`	Display messages to all users
`B  D,ALLUSERS,4`	Display message No. 4 to all users
`B  L,CHICAGO`	Delete all messages to Chicago
`B  L,LONDON,1`	Delete message No. 1 to London
`B  L,ALLUSERS`	Delete all messages to all users
`B  L,ALLUSERS,6`	Delete message No. 6 to all users
`B  L,ALL`	Delete the entire message queue

C (cancel)

The C command cancels a reader or writer routine.

Operation	Operands
C	queue[,ioaddr] ioaddr

queue — specifies the reader or writer routine in the xxyyy or yyy format, where

xx = BG, F4, F3, F2, or F1
yyy = PRT, PUN, or RDR

A specification in the yyy format specifies a partition-independent reader/writer routine.

ioaddr — specifies the address of the I/O device for which the routine was started in the cuu format.

If you wish to cancel a combined card and 3540 reader routine, specify the card device.

When a reader routine is reading input, the C command terminates processing and the job entry affected by the command must be re-entered.

Note: If you stop a reader routine using a 5425, press the NPRO key to clear the card path before starting another routine. Otherwise results with the next routine will be unpredictable.

When a writer routine is punching or printing output, the C command terminates output processing and the output of the job entry currently being processed is deleted from the output queue.

POWER acknowledges the C command with the message

```
1Q77I STOPPED xxycuu
```

where y = R for a reader routine
T for a print writer routine
P for a punch writer routine

Examples

Reader Routine

```
C  BGRDR,00C
C  RDR,00C
C  X'00C'
```

Writer Routine

```
C  BGPRT,00E
C  F2PUN,X'00D'
C  00E
```

D (display)

The D command displays the status of POWER or POWER RJE job entries in a specified queue or provides you with status information on the running POWER system. The information is displayed on SYSLOG.

Operation	Operands	Type
D	queue,name[,number]	Job option
D	queue[,ALL]	ALL option
D	queue,HOLD	HOLD option
D	queue,FREE	FREE option
D	queue,priority	Priority option
D	queue,class	Class option
D	queue,LOCAL	LOCAL option
D	A	Active task option
D	B	Free buffer option
D	Q	Queue status option
D	T	Time/day option
D	queue,RJE	RJE option

queue specifies the queue to which the command is directed in the xxyyy or yyy format, where:

xx = BG, F4, F3, F2, or F1
yyy = RDR, PRT, or PUN

A specification in the yyy format specifies a partition-independent display.

name specifies the name of the job entry.

number specifies the number assigned to the job entry by POWER when it is entered in the input queue. Job numbers may consist of from one to five numeric characters; it allows addressing of specific job entries with identical names.

ALL specifies a general request for status information on all job entries in the specified queue.

HOLD specifies a general request for status information on all job entries in the 'hold' state in the specified queue.

FREE — specifies a general request for status information on all job entries that are not in the 'hold' state in the specified queue.

priority — specifies a general request for status information on all job entries in the queue with the specified priority. Priority is specified by Pn, where n is a numeric character from 0 through 9. Nine is the highest priority.

class — specifies a general request for status information on all job entries belonging to the specified class. Class is specified by CLASSn, where n is any alphabetic character from A through Z.

LOCAL — specifies, in an RJE system, that status information be displayed on all job entries which do not send their output to a terminal user.

A — specifies a request for a list of all reader, writer and RJE routines currently active. Each entry in the list appears in the format 1Q20I xxycuu where:

```
xx    =  BG, F4, F3, F2, or F1
         XX for partition-independent routines
y     =  R for a reader routine
         T for a print writer routine
         P for a punch writer routine
cuu   =  device address of unit record device
```

If no routines have been started, the message

```
1Q12I NO ACTIVE ROUTINES
```

is issued.

B — specifies a request for a display of the number of program and data buffers currently not used in the following format:

```
1Q14I xx PROGRAM BUFFERS
1Q15I xx DATA BUFFERS
```

Q — specifies a request for the number of free QFILE records and track groups. The report is in the format

```
1Q10I xxxx TRACK GROUPS
1Q11I xxx QFILE RECORDS
```

T — specifies a request for the DOS/VS system time and date. These are issued in the format

```
1Q16I TIME = xx/xx/xx, DATE = xx/xx/xx
```

RJE — specifies that the command applies to RJE job entries in the specified queue.

D (display)
(continued)

The general format of the report obtained by means of the D command is

```
Report   1Q27I  INTD   00022  5   *   A   xx 000020000
                |      |      |   |   |   |   |
                |      |      |   |   |   |   no. of records[1]
                |      |      |   |   |   |
                |      |      |   |   |   partition ID
                |      |      |   |   |
                |      |      |   |   job entry class
                |      |      |   |
                |      |      |   *   =   in process
                |      |      |   D   =   ready to run
                |      |      |   H   =   in 'hold' state
                |      |      |   R   =   output for RJE terminal
                |      |      |   W   =   output for RJE
                |      |      |           terminal in 'hold' state
                |      |      |
                |      |      priority
                |      |
                |      job entry number
                |
                job entry name
```

[1] This number is normally equal to the number of printed or punched cards. If cards are printed **and** punched, however, 'no. of records' is equal to the number of punch and print operations performed.

Examples

```
Command:   D BGRDR,ASSEMBLY,10
Report:    1Q27I ASSEMBLY 00010 2  00000103
Command:   D BGPRT,CLASSA
Report:    1Q27I REPORTA 00022 5 * A 00002000
```

E (end)

The E command terminates the POWER program, and frees the POWER partition and the POWER-supported partition. The POWER-supported partitions will have SYSIPT, SYSRDR, SYSLST, and SYSPCH unassigned. The following messages appear on SYSLOG for each supported partition that has been active under POWER:

```
0S01I JOB CANCELED DUE TO OPERATOR INTERVENTION
1I00A READY FOR COMMUNICATIONS
```

If no further activities are planned for such partitions, they may be stopped by means of the DOS/VS command STOP.

Operation	Operands
E	[KILL]

no operand — terminates the POWER program, frees the POWER partition for another program, and restores the system to normal DOS/VS operation. POWER-supported partitions continue until processing of their current job entries has completed. After completion, POWER issues the message

```
1Q42I POWER HAS BEEN TERMINATED
```

KILL — cancels the POWER program immediately and causes a dump to be produced if SYSLST is assigned in the POWER partition. All running partitions that are supported by POWER are canceled, too. The active POWER routines are stopped immediately.

When the dump has been printed, POWER issues the message

```
1Q42I POWER HAS BEEN TERMINATED
```

The E command has no effect on partitions that have been stopped by the DOS/VS command STOP. If a stopped partition is supported by POWER, it must therefore be activated by means of the BATCH command **before** issuing the E command to ensure that the last job entry in the partition is terminated.

F (flush)

The F command terminates processing of the job entry output for a specified writer routine and deletes the job entry output from the output queue.

Operation	Operands
F	queue[,uraddr] queue,[uraddr],ALL uraddr[,ALL]

queue — specifies the queue for which the writer routine was started in the xxyyy or yyy format, where

xx = BG, F4, F3, F2, or F1
yyy = PRT, or PUN

uraddr — specifies the I/O address for which the routine was started in the cuu format.

ALL — specifies that, in addition, all output in the specified queue is to be deleted.

The F command permits the specified writer routine to remain active to process job entry output as it becomes available in the output queue.

POWER does not acknowledge the F command with a message. However, the specified device should stop printing or punching the current job entry. If ALL was specified, the device will wait until new output is available in the specified queue. If ALL was not specified, the device will, after a short pause, begin the output of the next job entry in the specified queue.

Examples

```
F  BGPRT,00E
F  00E
F  F2PUN,00D,ALL
F  PRT,00E
F  X'00D',ALL
```

G (go)

The G command reactivates a POWER or a POWER RJE routine that is waiting for an operator action.

Operation	Operands	Type
G	queue[,uraddr] uraddr partition	Routine management command
G	RJE,lineaddr	RJE command

queue — specifies the queue for which the reader or writer routine was started in the xxyyy or yyy format, where

xx = BG, F4, F3, F2, or F1
yyy = RDR, PRT, or PUN

A queue specification in the yyy format indicates a partition-independent routine.

uraddr — specifies the address of the I/O device for which the routine was started in the cuu format.

partition — specifies the partition in which execution was halted.

RJE — specifies that the command is directed to an RJE routine.

lineaddr — specifies the line for which the RJE routine was started in the cuu format, where

c = channel
uu = unit

POWER does not acknowledge the G command with a message. However, reactivating a reader or writer routine causes the specified device to resume operation.

With DOC support you must, after starting a routine with the G command, delete any E-type messages referring to the routine, as the deletion is not performed automatically.

Examples

Reactivate Reader or Writer Routine

```
G  BGPRT,00E
G  PRT,00E       (partition-independent routine)
G  F2PUN,00D
G  X'00E'
```

Reactivate POWER-Supported Partition

```
G   BG
G   F2
```

Reactivate an RJE Routine

```
G   RJE,022
```

H (hold)

The H command places the specified job entry in the 'hold' state. It prevents processing of one or all job entries in the specified queue. Job entries may be removed from the hold state only by the R command.

Operation	Operands	Type
H	queue,name[,number]	Job option
H	queue[,ALL]	ALL option
H	queue,priority	Priority option

queue specifies the queue to which the H command is directed, in the xxyyy format, where

xx = BG, F4, F3, F2, or F1
yyy = RDR, PRT, or PUN

name specifies the name of the job entry. 'name' may be from one to eight alphameric characters.

number specifies the number assigned to the job entry when it is entered in the input queue. Use the D command to obtain the job number. 'number' may be from one to five numeric characters; it allows addressing of specific job entries with identical names.

ALL specifies that all job entries in the specified queue are to be placed in the hold state.

priority specifies that all job entries in the specified queue with the specified priority are to be placed in the hold state. Priority is specified by Pn, where n is a numeric character from 0 through 9. Nine is the highest priority.

Examples

```
H  BGRDR,ASSEMBLY,2
H  BGPUN,P3            (priority option)
H  F2RDR,ALL
H  F2PRT,TEST
```

I (inquire)

The I command provides the central installation operator with status information on a specific RJE routine or on all RJE routines. POWER responds with a report which includes, for each specified routine, the routine designation, termid, userid, and status. The status is one of the following: processing, active, inactive, not attached, not initiated, not supported.

Operation	Operands
I	U,userid T,termid L,lineaddr ALL

U,userid specifies a valid 'userid' as defined during POWER generation. 'userid' may be from one to eight alphameric characters.

T,termid specifies a valid 'termid' as defined during POWER generation. 'termid' may be from one to eight alphameric characters.

L,lineaddr specifies the teleprocessing line for which an RJE routine is or may be started in the cuu format, where

c = channel
uu = unit

ALL specifies that a status report is required for all RJE routines.

Examples

```
I  U,SMITH
I  T,NEWYORK
I  L,030
I  ALL
```

J (job entry accounting)

The J command causes processing of the POWER accounting file. It is accepted only if ACCOUNT=YES was specified as a POWER generation option.

The J command allows you to retrieve the accumulated account file records. POWER informs you when the account file is full by issuing a message on SYSLOG. You must then use the J command to empty the account file.

You may use the J command to punch the account file to cards, to write the account file to tape, or to delete the account file records. The file is then reusable from its beginning.

The account file may be processed at any time while POWER is active.

Operation	Operands	Type
J	punchaddr	punch cards option
J	tapeaddr	write tape option
J	DEL	delete option

punchaddr — specifies the address of the punch that is to punch the accounting records. It may be specified as cuu or X'cuu'.

tapeaddr — specifies the address of the tape device for the accounting records. It may be specified as cuu or X'cuu'.

When processing is completed, a tape mark is written and the tape will be rewound and unloaded.

DEL — specifies that the accounting file records are to be deleted.

Examples

```
J  00D    punch accounting file
J  180    write the accounting file to tape
J  DEL    delete the accounting file
```

L (delete)

The L command deletes a job entry from the specified queue. Job entries being processed are not affected by this command. If there are no job entries in the specified queue, the message

1Q06I QUEUE IS EMPTY

appears on SYSLOG.

Operation	Operands	Type
L	queue,name[,number]	Job option
L	queue,ALL	ALL option

queue specifies the queue to which the DELETE command is directed in the xxyyy format, where

xx = {BG, F4, F3, F2, or F1}
yyy = {RDR, PRT, or PUN}

name specifies the name of the job entry. It may contain from one to eight alphameric characters.

number specifies the number assigned to the job entry by POWER when the job entry is in the input queue. Use the DISPLAY command to obtain the number of the job entry. 'number' may be from one to five numeric characters; it allows addressing of specific job entries with identical names.

ALL specifies that all job entries in the specified queue are to be deleted.

Examples

```
L  BGPRT,RUN1,3
L  F2RDR,ALL
```

M (multiple copy)

The M command displays or alters the copy counter for the job entry output currently being processed by the specified writer routine.

Operation	Operands	Type
M	[queue,]uraddr	Display option

M	{ queue,[uraddr],additional }
	{ uraddr,additional }
		Alter option

queue specifies the queue for which the writer routine was started in the xxyyy or yyy format, where

xx	=	BG, F4, F3, F2, or F1
yyy	=	PUN, or PRT

A queue specification in the yyy format indicates a partition-independent routine.

uraddr specifies the address of the I/O device for which the writer routine was started in the cuu format.

additional specifies the number of additional copies to be made. It is entered in the form of one or two numeric characters.

Display option

The display option is used to determine the number of copies left to be processed, including the one in progress.

The alter option adds the specified value to the current copy counter. It is not possible to decrease the copy counter. The copy counter cannot be higher than 255.

Alter option

If you have used the display option and wish to increase the copy counter, make sure that you have enough time to complete the alter option. For instance make the device temporarily not ready. Otherwise, the alter option will be applied to the next job entry.

Examples

Display Option

```
M  BGPRT,00E
M  F2PUN,00D
M  PRT,00E
M  X'00D'
```

Alter Option

```
M  BGPRT,00E,12
M  F2PUN,00D,3
M  00E,21
```

O (output)

The O command changes the output destination of a job entry.

Operation	Operands
O	name,[number], { userid ALLUSERS LOCAL,[queue] }

name specifies the name of the job entry. It may be from one to eight alphameric characters.

number specifies the number assigned to the job entry by POWER when the job entry is logged in the input queue. Use the D command to obtain the number of the job entry. Job number may be from one to five numeric characters; it allows addressing of specific job entries with identical names.

userid specify the destination of the output. 'userid' is valid if it is in the Userid List, which is created when POWER is generated.

ALLUSERS ALLUSERS specifies that output will be available to all users on a read-only basis. Only the central installation operator may delete job entries with the ALLUSERS destination.

LOCAL LOCAL specifies that the output will be directed to the local output queues.

queue may be specified only together with LOCAL

queue specifies the queue to which the local output is to be directed and can be either PRT or PUN. It allows separate handling of print and punch output.

yyy = PRT or PUN

If queue is not specified, all output is directed to the local output queues.

Examples

```
O  BGJOB1
O  BGJOB2,2,ALLUSERS
O  BGJOB3,,NEWYORK
O  BGJOB4,,LOCAL,PRT
```

P (stop)

The P command stops a reader, writer, or RJE routine.

Operation	Operands	Type
P	{ queue[,ioaddr] { queue,[ioaddr],EOJ { ioaddr[,EOJ]	Routine management command
P	queue,[ioaddr],CHECKPOINT ioaddr,CHECKPOINT	CHECKPOINT option (only for 2560 and 5425)
P	RJE,lineaddr	RJE command

queue — specifies the queue for which the reader or writer was started in the xxyyy or yyy format, where

xx = { BG, F4, F3, F2, or F1}
yyy = { RDR, PRT, or PUN}

A queue specification in the yyy format indicates a partition-independent routine.

ioaddr — specifies the address of the I/O device for which the routine was started in the cuu format. If you have to stop a combined card and diskette reader routine, specify the card device.

EOJ — causes the routine to stop at the end of the job entry.

RJE — specifies that the command is directed to an RJE routine.

lineaddr — specifies the teleprocessing line for which the RJE routine was started in the cuu format, where

c = channel
uu = unit

CHECKPOINT — can be specified for a punch routine if a 2560 or 5425 is used for punch output from a disk spool file. It causes the punch routine to take a checkpoint and terminate processing, in order to allow a reader routine to be started. After the reader routine has terminated, punching of the interrupted job entry can be resumed from the checkpoint by means of the RESTART option of the S command.

Note: If you stop a reader routine using a 5425, press the NPRO key to clear the card path before starting another routine. Otherwise results with the next routine will be unpredictable.

If a reader or RJE routine was preparing a job entry in the input queue when a P command was issued, further input processing is terminated immediately. Such an interrupted job entry must be re-entered. If a writer or RJE routine was punching or printing output when the P

P (stop)
(continued)

command was issued, further output processing is terminated immediately. The output is not deleted from the output queue, but will be printed or punched when the routine is started again.

Unless an emergency exists, you should use the I command to determine the status of the remote terminal prior to stopping an RJE routine.

POWER acknowledges the P routine-management command with the message

```
                1Q77I STOPPED xxycuu
```

where y = R for a reader routine,
T for a print routine, or
P for a punch routine

POWER acknowledges the P RJE command with the message

```
                1R29I device type ROUTINE HAS BEEN
                TERMINATED RJE line address
```

where device type = 2770, 2780, or 3780
line address = cuu

Examples

```
P   BGRDR,00C
P   F2PUN,00D,EOJ
P   F2PRT,00E
P   PRT,00E
P   00E
P   00E,EOJ
P   RJE,031
P   BGPUN,CHECKPOINT
P   00C,CHECKPOINT
```

R (release)

The R command releases a job entry from the hold state in the specified queue. If there are no job entries in the queue, POWER issues the following message on SYSLOG:

1Q06I QUEUE IS EMPTY

Note: Make sure that the partition is started before releasing a job.

Operation	Operands	Type
R	queue,name[,number]	Job option
R	queue[,ALL]	ALL option
R	queue,priority	Priority option

queue — specifies the queue to which the RELEASE command is directed in the xxyyy format, where

xx = BG, F4, F3, F2, or F1
yyy = RDR, PRT, or PUN

name — specifies the name of the job entry. It may contain from one to eight alphameric characters.

number — specifies the number assigned to the job entry by POWER when the job entry is logged in the input queue. Use the D command to obtain the number of the job entry. 'number' may be from one to five numeric characters; it allows addressing of specific job entries with identical names.

ALL — specifies that all job entries in the specified queue are to be released from the hold state.

priority — specifies that all job entries in the specified queue with the specified priority are to be removed from the hold state. Priority is specified by Pn, where n is a numeric character from 0 to 9. Nine is the highest priority. This operand is not valid if PRIORTY=NO was specified during POWER generation.

Examples

```
R  BGPRT,ASSEMBLY,26
R  F2PRT,ALL
R  BGPUN,P3               (priority option)
```

Note: Make sure that the partition is started before releasing the job.

S (start)

The S command starts a reader, writer, or RJE routine.

Operation	Operands	Type
S	queue,[uraddr],[buffers], {tapeaddr class}	Routine management command
S	queue,[uraddr],[buffers],[RESTART	RESTART command
S	queue,[duaddr],[buffers],['filename'], [vols],[S]	diskette reader
S	queue,craddr,[buffers],duaddr	combined card and diskette reader
S	RJE,lineaddr[,type]	RJE command

queue specifies the queue of the routine in the xxyyy or yyy format, where

xx = { BG, F4, F3, F2, or F1}
yyy = { RDR, PRT, or PUN }

A queue specification in the yyy format indicates a partition-independent routine.

uraddr specifies the I/O device for which the routine is to be started in the cuu format.

Special treatment for 5425 MFCU. Prior to starting a reader or punch routine with a 5425 MFCU empty both hoppers, press NPRO twice, then put cards in hopper 2, then press START key. A reader routine with 5425 cannot be started if the 5425 is not in the ready state.

buffers specifies the number of I/O buffers (1 or 2) to be assigned to the routine.

For diskette input, the reader routine will attempt to get sufficient buffer space to hold all records from one diskette track, regardless of the number of buffers specified.

For punch routines, buffers may be specified as two numeric characters. The first character specifies the number of buffers (1 or 2) to be assigned to the routine. The second character specifies a pause code (0, 1, or 2). Zero indicates that the forms number specified in the * $$ PUN card is not to be printed out on SYSLOG. Specifying 1 indicates that each form number should be printed on SYSLOG and that the routine should pause before processing each job entry. Specifying 2 indicates that every other form number should be printed and that the routine should pause before processing every other job entry. If the second character is omitted, zero is assumed. The first character may not be omitted.

tapeaddr specifies the address of the tape drive that contains output. It is entered in the cuu format.

This parameter is necessary if output spooled on tape is to be printed or punched by this routine.

Before starting a routine for a given device, any other routine using that device must be stopped or canceled. **Note:** Tape must be positioned correctly before starting a tape writer task.

class specifies the class with which the writer routine is to be associated. Up to four classes may be specified in the form of an alphabetic character string.

RESTART specifies that the punch routine is to resume processing at the point where it was interrupted by a P command with the CHECKPOINT option (for IBM 2560 MFCM and 5425 MFCU only).

filename is the name of the diskette file to be read; it must be identical to the filename in the HDR1 label on the diskette. One to eight alphameric characters, including blanks, may be entered between the quotes.

vols is the maximum number of diskette volumes to be read. Any number from 1 to 255 may be specified; the default value is 1.

S specifies that volume sequence checking is desired. If S is entered, the sequence number of the first volume must be 1, the sequence number of the second volume must be 2, and so on.

craddr specify the card reader (craddr) or a combined reader routine.

duaddr specify the 3540 diskette unit.

craddr specify the card reader (craddr) in the cuu format for a reader routine that is to read from both a card reader and a diskette.

duaddr specify the 3540 diskette unit in the cuu format.

RJE specifies that the command was given to start an RJE routine.

lineaddr specifies the line for which the RJE routine is to be started in the cuu format.

Examples

type	2780	-	for 2780 terminal
	3780	-	for 3780 terminal
	2770,128	-	for 2770 with 128 byte buffer
	2770,256	-	for 2770 with 256 byte buffer
	2770,512	-	for 2770 with 512 byte buffer

POWER acknowledges the S RJE command by issuing the message

```
1R24I lineaddr type HAS BEEN STARTED
```

```
S  BGRDR,00C
S  BGPRT,00E,2
S  F2PUN,00D,20
S  RDR,00C                (partition-independent
                          routine)
S  BGPRT,00E,,285         (tape writer routine)
S  PRT,00E,,ABCD          (partition-independent
                          routine of classes A-D)

S  RJE,020
S  RJE,030,2770

S  BGRDR,00C,,RESTART
S  RDR,00B,,'INFILE',6,S  (diskette input)
S  RDR,00C,,00B           (combined card and
                          diskette reader)
```

T (restart)

The T command restarts the print output of a job entry from the beginning or from a specified page.

Operation	Operands
T	{ writer[,uraddr] writer,[uraddr],count uraddr[,count] }

writer — is specified in the xxPRT or PRT format, where

xx = BG, F4, F3, F2, F1

A specification of PRT (without the partition identifier) indicates a partition-independent writer routine.

uraddr — specifies the I/O device for which the print writer routine was started in the cuu format.

count — is specified as a signed or unsigned value from 0 to 999. A plus sign indicates pages forward, a minus sign indicates pages backward from the interrupted page. No sign indicates a specified page count from the beginning. If the operand is omitted, printing resumes from the first page.

If the page count with a minus sign is higher than the number of printed pages, printing also resumes from the first page.

POWER does not acknowledge the T command with a message. However, the device with which the routine is associated should continue with the output after positioning to the new page.

Page numbering is based on the count of the 'skip to channel 1' commands and not on the actual headline numbers.

T BGPRT,00E	restart from the beginning
T F2PRT,00F,10	restart from page 10
T 00E,+100	restart 100 pages past the last page previously printed

Z (trace)

The diagnostic trace command Z conditionally monitors TIB contents, QFILE record format and contents, as well as DATAFIL records for valid disk address and chain pointers. These aids are generated as a user option with DIAG=YES and are useful for error analysis. To obtain the result of the trace, refer to any POWER dump, for example the dump produced by the E KILL command.

Operation	Operands	Type
Z	DUMP, { TIB[,QFL][,DFL] QFL[,DFL][,TIB] DFL[,TIB][,QFL] END LST }	DUMP option

Z	TRACE, { SVC
	TSK[,SVC]
	END
	LST }
		TRACE option

TIB specifies that the contents of the TIB are to be written in the wrap-around buffer.

QFL specifies that all write operations to QFILE are to be written in the wrap-around buffer.

DFL specifies that all write operations to DATAFIL are to be written in the wrap-around buffer.

SVC specifies that any active trace is terminated, and SVC 0 trace is started.

TSK specifies that any active trace is terminated, and routine selection trace is started. All routines are traced except 'NO WORK', 'QFILE', and 'DATAFIL' routines.

If TSK and SVC are specified, both traces are active in the wrap-around buffer.

LST specifies that the current status of the diagnostic operation is to be listed on SYSLOG. This option does not change the status itself.

END specifies that the diagnostic operations are to be terminated.

Examples

```
Z  DUMP,TIB,DFL
Z  DUMP,QFL,DFL,TIB
Z  DUMP,LST
Z  DUMP,END
Z  TRACE,SVC
Z  TRACE,TSK
Z  TRACE,SVC,TSK
Z  TRACE,LST
Z  TRACE,END
```

POWER and POWER RJE JECL Statements

For POWER, a job entry consists of the input that is read by a POWER reader routine and logged by that routine under a single name. A job entry may be created by means of either of the following:

DOS/VS Job Control Language (JCL)
POWER Job Entry Control Language (JECL)

A job entry under POWER

If JCL is used, a job entry is logged under the name specified in the //JOB statement, and this statement, together with the /& statement, delimits the job entry. If JECL is used, the * $$ JOB and * $$ EOJ statements delimit a job entry, and the name specified in the * $$ JOB statement is used to log the particular job entry.

When logging job entries, POWER assigns a number to each job entry name that it logs in order to distinguish between job entries for which identical names were specified. You must use that name and, possibly, the number of a particular job entry whenever you issue an operator command to control the processing of that job entry.

Purpose of JECL

JECL provides a convenient means for the programmer to specify how POWER is to handle a particular job entry. He may use JECL statements for the following:

1. Under POWER and POWER RJE

   - assign priorities
   - hold a job entry in any or all queues
   - spool output to tape rather than to disk
   - suppress spooling of unit record print or punch output
   - specify the partition in which the job entry is to be executed.

2. Under POWER RJE only

   - direct output to another user or, on a read-only basis, to all terminal users
   - direct output to be returned to a terminal or to be processed locally at the central system
   - hold output until an OUTPUT terminal command is issued.

Normally you will not be concerned with the completion of JECL statements; however, you may be called upon to change or correct JECL statements that your programmer prepared and inserted in the job stream.

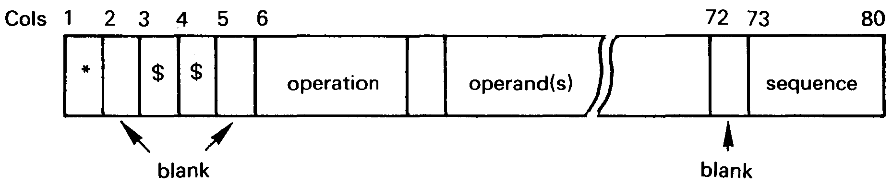
List of JECL statements

Figure IV-6 lists all JECL statements. This list is followed by a brief description of the function and format of each statement, including a brief discussion of the various operands. For a detailed discussion of these statements and their operands, refer to *DOS/VS System Control Statements, GC33-5376.*

JECL Statement	Function
* $$ JOB	Indicate the beginning of a job entry and provide job entry handling information.
* $$ EOJ	Indicate the end of a job entry.
* $$ DATA	Insert data from the input reader (or the reader of a terminal) into data from a source library.
* $$ RDR	Insert a diskette file into the input stream from the card reader.
* $$ PRT	Provide disposition and handling information for printed output.
* $$ PUN	Provide disposition and handling information for punched output.
* $$ SLI	Insert data from a sub-library into the job stream (from the input reader or the reader of a terminal).

Figure IV-6. **List of JECL statements.**

Any JECL statement must be completely contained in columns 1-71 of a card; column 72 must be blank. Columns 73-80 may be used as a sequence field; continuation cards are not permitted. The format is:



* $$ DATA

Insert data from the input reader (or the reader of a remote work station) into data from a library.

When the programmer requests POWER to insert data from a source library into his job entry by a * $$ SLI card, he can use the * $$ DATA statement to intersperse the data retrieved from the library with data from the input reader (or the reader of a terminal).

Operation	Operands
* $$ DATA	name

The name to be specified will be provided by your programmer.

* $$ EOJ

Indicate the end of a job entry.

This statement marks the end of a job entry that is delimited and controlled by means of JECL statements.

Operation	Operands
* $$ EOJ	blank

No operand is required.

POWER and POWER RJE JECL Statements

* $$ JOB

Indicate the beginning of a job entry and provide job entry handling information.

Operation	Operands
* $$ JOB	[name],[H],[priority],[partition], ⎡userid⎤ ⎣ALLUSERS⎦

When one or more of the operands are omitted, you must indicate this by a comma for each omitted operand, except for omitted operands at the end of the statement, for example

```
* $$ JOB  CONVERT,H,,F2
```

name — A string of one to eight characters, the first one of which must be alphabetic. Normally your programmer will provide the name to be specified. If you omit name, POWER assigns AUTONAME as the name of the job entry.

H — This operand indicates that the job entry is to be held in the input queue until you issue an R (release) command for that entry. If H is not specified, the particular job entry is processed with other job entries of the same priority on a first-in first-out basis.

priority — Any value from 0 to 9 specifying the priority with which the particular job entry is to be processed within the pertinent partition. Nine is the highest priority.

partition — You can specify:

BG, F1, F2, F3, F4

This parameter is only used by partition independent reader routines and by RJE routines.

userid — Used only in conjunction with remote job entry. The parameter specifies who is authorized to retrieve the output of the job entry. Your specification for userid is valid only if it matches with a userid entry in your system.

ALLUSERS — Specify ALLUSERS if the output of the particular job entry is to be available to all RJE users on a read-only basis.

*** $$ PRT**

Provide disposition and handling information for printed output.

If you use this statement, it must follow either the * $$ JOB or the * $$ PUN statement. Only one * $$ PRT statement is permitted for a job entry.

Operation	Operands
* $$ PRT	[disposition[class]],[forms-number], ⎡number-of-copies⎤ ⎣devaddr-of-tape ⎦ ,[number-of-lines-before-msg],[linetab]

When one or more of the operands are omitted, you must indicate this by a comma for each omitted operand, except for omitted operands at the end of the statement.

```
* $$ PRT D,,3,5
```

disposition — This can be one of the following:

D — spool output to disk.

T — spool output to tape (only valid if POWER at your installation includes the tape output writer function).

N — do not intercept print requests. Specifying N suppresses spooling of printed output.

H — hold the output until requested by means of an R command (for output on a local printer) or an OUTPUT terminal command (for output on a terminal).

R — (only for remote job entry) return printed output to the authorized terminal user.

class — Any letter from A through Z can be specified. Class can be used in conjunction with form number to group printed output by type. A print writer may then be started by class to process printed output of the same type. Class can be specified only together with a disposition specification of D. Class is ignored if the job entry is submitted from a terminal.

forms-number — This operand specifies the type of forms to be used. The form number must be a string of one to four alphameric characters.

number-of copies — You can specify a number of one or two digits. If you specify 0 or 00, or do not specify any number at all, POWER produces one copy of the printed output for the particular job entry. Number-of-copies cannot be specified when the disposition is specified as T or N. If additional copies are required when the disposition is

T, you can start the tape writer as many times as copies are needed.

devaddr-of-tape

Use this operand when the disposition is specified as T. The format of this specification is cuu, the hexadecimal characters representing the channel and unit address of the tape drive to be used. If this parameter is omitted, you must specify cuu at the console on demand of the system.

number-of-lines-before-msg

You can specify a number of one to six digits. The operand indicates how many print records are to be intercepted by POWER before a warning message is sent to the central installation operator. This operand is used only for disk spooling.

linetab

This operand specifies the carriage control tape format to be used when print requests are intercepted by POWER.

* $$ PUN

Provide disposition and handling information for punched output.

If you use this statement, it must follow either the * $$ JOB or the * $$ PRT statement. Only one * $$ PUN statement is permitted for a job entry.

Operation	Operands
* $$ PUN	[disposition[class]],[card-number], [number-of-copies devaddr-of-tape] ,[number-of-cards-before-msg]

When one or more of the operands are omitted, you must indicate this by a comma for each omitted operand, except for omitted operands at the end of the statement.

```
* $$ PUN DC,,,238
```

disposition — This can be one of the following:

D — spool output to disk.

T — spool output to tape (only valid if POWER at your installation includes the tape output writer function).

N — do not intercept punch requests. Specifying N suppresses spooling of punched output.

H — hold the output until requested by means of an R command (for output on a local punch) or an OUTPUT terminal command (for output on a terminal).

R — (only for remote job entry) return punched output to the authorized terminal user.

class — Any letter from A through Z can be specified. Class can be used in conjunction with card number to group punched output by types. A punch writer may then be started by class to process punched output of the same type. Class can be specified only together with a disposition specification of D. Class is ignored if the job entry is submitted from a terminal.

card-number — This operand specifies the type of cards to be used. The card number must be a string of one to four alphameric characters.

* $$ PUN (continued)

number-of-copies
You can specify a number of one or two digits. If you specify 0 or 00, or do not specify any number at all, POWER produces one copy of the punched output for the particular job entry. Number-of-copies cannot be specified when the disposition is specified as T or N. If additional copies are requested when the disposition is T, you can start the tape writer as many times as copies are needed.

devaddr-of-tape
Use this operand when the disposition is specified as T. The format of this specification is cuu, the hexadecimal characters representing the channel and unit address of the tape drive to be used.

number-of-cards-before-msg
You can specify a number of one to six digits. The operand indicates how many punch records are to be intercepted by POWER before a warning message is sent to the central installation operator. This operand is used only for disk spooling.

* $$ RDR Insert input data from diskette

The * $$ RDR statement indicates that a diskette file is to be inserted into the input stream read from a card reader. This statement is placed in the input stream at the point where the diskette file is to be inserted.

Operation	Operands
* $$ RDR	[SYSnnn],['filename'],[vols],[S]

When one or more operands are omitted, you must indicate this by a comma for each omitted operand, except for omitted operands at the end of the statement. Example:

```
* $$ RDR ,'INPUTFLE',3
```

SYSnnn is specified only if the diskette file is a data file (not a SYSIN file). nnn is the programmer logical unit number which will be used to read the file when the job entry is executed.

filename is the name of the diskette file to be read; it must be identical to the filename in the HDR1 label on the diskette. One to eight alphameric characters, including blanks, may be entered between the quotes.

vols is the maximum number of diskette volumes to be read. Any number from 1 to 255 may be entered; the default value is 1.

S specifies that volume sequence checking is desired. If S is entered, the sequence number of the first volume must be 1, the sequence number of the second volume must be 2, and so on.

* $$ SLI

Insert data from a sublibrary into the job stream.

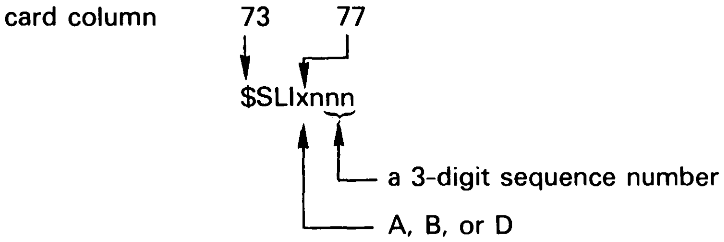
* $$ SLI statements are normally prepared by your programmer. He also determines where they should be placed within a job entry.

Operation	Operands
* $$ SLI	[sublibrary,]bookname

Sublibrary is specified as one character. Example of a statement with a complete operand:

```
* $$ SLI  A.INVOICE
```

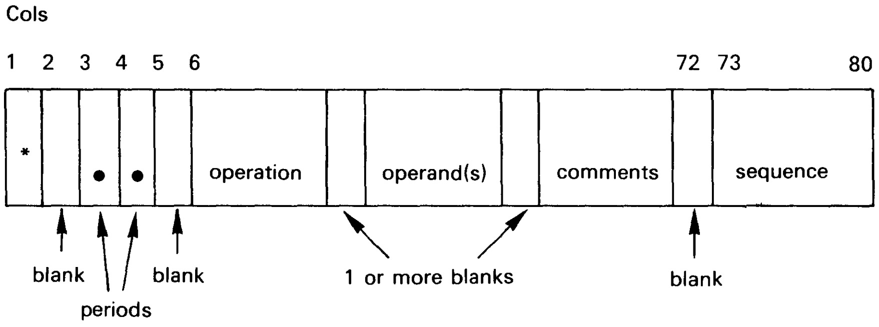
An * $$ SLI statement may be followed by one or more SLI update cards. Your programmer will provide those cards, if they are required. The format of SLI update cards is:



POWER RJE terminal commands are required to control jobs that are entered at a terminal for processing at a central installation. These commands are described here to facilitate direct communication between you and an operator at a remote terminal.

General format

POWER RJE terminal commands are entered on punched cards through the reader at the terminal. Each of these commands is composed of fields as shown below:



Fields ID and Operation are always required; most of the commands require at least one operand; the use of comments and sequence numbers is optional.

Sequence numbers are useful when a number of commands with the same operation code are submitted from one terminal. POWER RJE returns the sequence number in all responses and diagnostics relating to terminal commands. This allows the terminal operator to correlate the responses he rpceives with the commands he entered.

List of terminal commands

Figure IV-7 lists all terminal commands and shows their functions. Following this list is a detailed description of each command, including a discussion of the operands. The commands are described in alphabetical order.

Placement

POWER RJE terminal commands must be placed either ahead of the first job entry made during a terminal session, or between job entries, or after the last job entry of a terminal session.

Examples of commands

Figure IV-8 (at the end of this section) shows at least one example of each POWER RJE terminal command.

POWER RJE Terminal Commands

Command	Function
RJSTART	Attach a terminal to POWER RJE at the central installation.
RJEND	Detach a terminal from POWER RJE at the central installation.
LOGON	Begin a terminal session for remote job entry under POWER RJE.
LOGOFF	End a terminal session started by LOGON.
OUTPUT	Request the output of processed job entries.
CONTINUE	Specify the disposition of interrupted output.
DELETE	Remove a job entry from the specified queue.
STATUS	Obtain a status report on one or more RJE job entries in the specified queue.
BRDCSTR	Request a copy of broadcast messages.
MSGR	Send a message to the central operator.

Figure IV-7. List of Terminal Station Commands.

* . . BRDCSTR

Request a copy of broadcast messages

The BRDCSTR command enables the terminal operator to obtain a copy of broadcast messages for his terminal. POWER RJE responds to this command by returning a copy of the system broadcast messages to the terminal. These messages contain information of general interest to all RJE users or to specific users. They are retrieved only on request from a terminal.

Broadcast messages are entered into the POWER message queue by you, the central operator, by means of a B (broadcast) command. Messages for a specified user are deleted after a copy has been printed at the terminal.

ID	Operation	Operand
* . .	BRDCSTR	blank

No operand is required.

* . . CONTINUE

Specify the disposition of interrupted output.

Output may be interrupted when operator intervention at the terminal is required because, for example, forms on the terminal printer must be changed. The interrupted output is held until the appropriate CONTINUE command is transmitted.

Interruptions that require a CONTINUE command to be submitted cause a message to this effect to be printed. A card is read. If it is not a CONTINUE card, the input stream is flushed in preparation for the transmission of the CONTINUE command, and a message is written to the terminal printer stating that the job stream was flushed. If then the CONTINUE command is not transmitted within 28 seconds, a timeout will occur.

Following a time-out, the system re-issues the read request until the generated RETRY count is exhausted, at which time the work station is detached from the RJE system. If the terminal is detached, the interrupted output is saved, and the terminal operator may or may not receive the appropriate ending message, depending upon the output component involved in the interrupted transmission.

If the terminal is detached, the terminal operator must resubmit an RJSTART command and a LOGON command to reinstate his user session. The interrupted output must be requested via an OUTPUT command. The output is then transmitted from the beginning, but the terminal operator can skip printed output to any subsequent page by interrupting the output and subsequently submitting an appropriate CONTINUE command.

The CONTINUE command provides six options:

1. Transmission of the output starting with the block of records which was being written when the output was interrupted (the command does not require an operand).
2. Transmission of the interrupted output from the beginning.
3. Deletion of the output.
4. Hold the output until it is requested by an OUTPUT command.
5. Change the destination of output to a local output writer.
6. Page forward or backward from the interrupted page or from the beginning of the printed output.

* . . CONTINUE
(continued)

ID	Operation	Operand
* . .	CONTINUE	[BEGIN NO HOLD LOCAL PAGE [,count]]

no operand specifies that transmission of data is to be resumed with the output data block that was being transmitted when the output was discontinued.

BEGIN specifies that the terminal operator desires transmisstion of the job entry's output from the beginning.

NO indicates that the terminal operator desires the deletion of the job entry's output.

HOLD specifies that the job entry's output is to be placed in the hold state and will be requested via an OUTPUT command at some later time. This option has particular significance when a change of forms is required and the forms are not available for immediate delivery of the output.

LOCAL changes the destination of a job entry's output to a local output writer.

PAGE allows the terminal operator to page forward or backward up to 999 pages from the interrupted page or forward up to 999 pages from the beginning of the job entry's printed output.

count is specified as a signed or unsigned value of from 0 to 999. A plus (+) sign indicates page forward, minus (-) sign indicates page backward from the interrupted page. No sign indicates a specified page count from the beginning. If the operand is omitted, printing is resumed from the interrupted page. A page count beyond the limits of the job being printed causes the following:

a. If the user specifies too great a backward page count, printing starts from the beginning.

b. If the user specifies a page count greater than the last page, a message is printed informing him of the maximum number of pages in the job entry and requesting another CONTINUE card.

Note: To avoid the loss of printlines when the output is interrupted, use the PAGE option of the continue card with a count specification of -2.

* . . DELETE

Remove a job entry from the specified queue.

The terminal operator can use the DELETE command to cause either of the following:

1. Delete a specific job entry previously submitted by him.
2. Delete all job entries previously submitted by him.

The DELETE command does not remove named data files created by the job entry specified for deletion. The command is not necessary to remove output which has already been returned to the terminal user. All references to a job entry processed by POWER RJE are normally removed after the output is returned to the user. A job entry can be deleted only if it is on the input queue or if it is on an output queue.

ID	Operation	Operand
* . .	DELETE	queue, {name[,number] {ALL

queue — specifies the queue to which the DELETE command is directed in the xxyyy format, where

xx = partition to which the job entry was assigned for execution: BG, F4, F3, F2, or F1

yyy = RDR for reader
PRT for printer
PUN for punch

name — specifies the name by which the job entry is known to POWER. The specified name may be from one to eight alphameric characters long. Each job entry name is qualified by a number assigned to it by POWER. Therefore, if the possibility of duplicate job entry names exists, that number should be specified also. If a duplicate name exists and the number is not specified, the first job entry found with the specified name is deleted.

number — specifies the 1 to 5-digit number assigned to the job entry by POWER RJE.

The terminal operator may use the STATUS command to inform himself about the correct specification for number.

ALL — requests to delete all job entries in the specified queue that were submitted under the userid specified in the LOGON command for the current session.

* . . LOGOFF

End a terminal session started by LOGON.

The LOGOFF command indicates that the terminal user currently logged on has completed his session. After receiving a LOGOFF command, POWER RJE refuses job entries from the terminal until another LOGON command is submitted.

Output to be returned to the current terminal user can be returned only during a session that was started with his userid in the LOGON command. Whenever a terminal user ends a session, a message appears on the central operator's console, identifying the user and indicating time he logged off.

ID	Operation	Operand
* . .	LOGOFF .	blank

No operand required.

If the central system receives a valid LOGON command from a terminal with a session in progress, the central system logs off the current user and logs on the user identified in that LOGON command. if the central system receives an RJEND command from a terminal with a session in progress, it logs off the user and logically detaches the terminal. In other words, the terminal operator may omit the LOGOFF command when changing user sessions or detaching the terminal.

POWER RJE Terminal Commands

* . . LOGON

Begin a terminal session for remote job entry under POWER RJE.

Following an RJSTART command, a user at a terminal which is logically attached to POWER RJE issues a LOGON command to start his session. The command identifies the user to POWER RJE and allows him access to the system. By checking the userid in the LOGON command, POWER RJE guarantees that only authorized users can submit input or request output from the system. The LOGON command remains in effect until another LOGON, a LOGOFF, or an RJEND command is entered by the terminal operator.

A terminal user who has more than one terminal available may log on at two terminals at one time to permit the use of one line for input and another for output, two lines for output, or two lines for input.

ID	Operation	Operand
* . .	LOGON	userid

userid — specifies the name assigned to the terminal user by the installation. If the userid specified in this parameter is not valid, POWER RJE rejects the command and a corrected statement must be submitted. The userid may consist of from one to eight alphameric characters. Whenever a terminal user begins a session, a message appears on the central operator's console identifying the user and indicating the time the terminal operator logged on.

* . . MSGR

Send a message to the central operator.

Messages sent to the central operator by means of the MSGR command are displayed on SYSLOG when they are received.

ID	Operation	Operand
* . .	MSGR	M,'text'

M,'text' specifies the message text to be sent. The message text must be enclosed in single quotation marks. The text itself may include up to 40 printable characters and blanks. Continuation lines are not permitted. Any messages requiring more than 40 characters of text must be sent by multiple commands.

Quotation marks contained within the message text must be doubled, and each quotation mark counts as one text character.

* . . OUTPUT

Request the output of a processed job entry.

The OUTPUT command can be used to request the output of job entries whose output destination is either the terminal operator's userid or ALLUSERS.

A job entry's output destination can be specified

1. explicitly in the * $$ JOB statement,

2. implicitly if, in the * $$ JOB statement for a remotely submitted job entry, no termid or userid was specified (in this case, POWER RJE assumes that the job entry's output is to be made available only on the terminal user who was logged on when the entry was submitted), or

3. explicitly by means of an O (output) command issued by the central operator.

If processing of a particular job entry has not finished when the output is requested, the job entry is not in the output queues and POWER RJE returns a message indicating that the job entry was not found. The OUTPUT command must be resubmitted after processing of the job entry has completed. The OUTPUT command provides the following options:

1. The terminal operator may request the output of a specific job entry if that job entry's output destination is either the requesting operator's userid or ALLUSERS.

2. The terminal operator may request all output whose destination is identical with the requesting operator's userid.

3. The terminal operator may request the output whose destination is ALLUSERS.

If the terminal operator is not an authorized recipient of the requested output, POWER RJE returns either of the following messages:

1R66I	JOB ENTRY name NOT FOUND
1R69I	NO OUTPUT WAITING

The same messages are returned to the terminal operator if he requests the output of a job entry that was not entered previously.

A user is automatically a valid recipient of a job entry's output if he submits the job entry, unless the destination field in the * $$ JOB statement specified another userid. Job entries with the destination ALLUSERS will not be purged when the output function has completed.

* . . OUTPUT
(continued)

No OUTPUT command is needed to retrieve a job entry's output if the entry is completely processed and

1. the user who submitted the job entry is still logged on and
2. that output was neither placed into the hold state nor directed to another terminal user.
3. the transmission of another job entry ends.

ID	Operation	Operand
* . .	OUTPUT	name[,number] ALL ALLUSERS

name indicates that the request is for the output of the job entry named in the parameter.

number specifies the number assigned to the job entry by POWER RJE when the job entry was logged in the input queue. The terminal operator may use the STATUS command to obtain the correct specification for number. Number, which may be specified as a decimal number of one to five digits, is needed only if job entries with identical names were submitted.

ALL specifies that POWER RJE is to return to the terminal operator all the available output destined for him in accordance with his userid.

ALLUSERS specifies that POWER RJE is to return to the terminal operator all the output available on a read-only basis with destination ALLUSERS.

* . . RJEND

Detach a terminal from POWER RJE at the central installation.

The RJEND command allows a terminal user to logically detach his terminal from POWER RJE. No output of the job entry is returned to the terminal after the RJEND command has been received. POWER RJE transmits the message

IR71I RJEND PROCESSED

on receipt of the RJEND command.

If the terminal is connected to the central system via a switched connection, the connection is broken. To ensure successful termination, the operator should not disconnect his terminal before he has received the message indicating that the RJEND command was accepted. No further communication occurs until the terminal resumes RJE activity with an RJSTART command.

ID	Operation	Operand
* . .	RJEND	blank

No operand is required.

* . . RJSTART

Attach a terminal to POWER RJE at the central installation.

The RJSTART command logically attaches a terminal to POWER RJE, that is, it indicates to POWER RJE that remote job entry is intended from the specified terminal.

The RJSTART command must be the first statement received from an inactive (logically not attached) terminal. In addition to identifying the terminal to POWER RJE, this command allows the terminal operator to request broadcast messages before continuing RJE processing. Communication proceeds between the central system and the terminal after a valid RJSTART command has been received. Once the terminal is logically attached, a terminal operator may gain access to the central system by logging on at the terminal with the LOGON command.

> If the terminal operator desires to resume RJE activity after the terminal has been logically detached from the system (after an RJEND command or a system failure), he must resubmit the RJSTART command.

> Only the LOGON command or the RJEND command may immediately follow the RJSTART command.

ID	Operation	Operand
* . .	RJSTART	termid,[BRDCST],[termtype], [bufsize],[NOPUNCH]

termid — specifies the name of the terminal. The specified name must be contained in POWER RJE's list of authorized user identifications. The name may consist of from one to eight alphabetic and/or numeric characters. If the specified termid is not recognized as a valid name, POWER RJE rejects the command, and a corrected statement must be submitted.

BRDCST — specifies that a copy of the broadcast messages for the specified termid is desired before RJE operation is continued. Messages for the specified termid are deleted after a copy has been printed at the terminal. If this parameter is omitted, no messages are broadcast at this time.

termtype — is the type of the terminal: 2770, 2780, or 3780 may be specified. Be sure that termtype is specified, as automatic distinction between the three terminals by the RJE routine is not possible.

bufsize is the size of the buffer of a 2770 terminal: 128, 256, or 512 may be specified. Default is 128. This operand must not be specified for 2780 and 3780 terminals.

NOPUNCH notifies the system that the terminal has no punch.

If the Space Compress/Expand feature is to be used, the RJSTART card must contain at least two consecutive blank columns.

* . . STATUS

Obtain a status report on one or more RJE job entries in the specified queue.

RJE returns the status of only those job entries which are currently in the specified queue and which have the terminal operator's userid or ALLUSERS specified as output destination. The STATUS command provides the following options:

1. The terminal operator can request the status of a specific job entry.
2. The terminal operator can request the status of all job entries which are in the specified queue and for whose output he is an authorized recipient.

For each job entry that satisfies the command, POWER RJE transmits the following:

- name of job entry and number
- priority
- number of logical records associated with the job entry in the specified queue
- disposition of the job entry in the RDR, PRT, and PUN queues as specified by the JECL statements for the entry.

If POWER RJE is unable to provide the above information in response to a STATUS command, it transmits either of the following messages:

1R66I JOB ENTRY name NOT FOUND
1R76I queuename QUEUE IS EMPTY

POWER RJE immediately returns the status of only those job entries which satisfy a STATUS command at the time the command is received. POWER RJE does not automatically return the new status of a job entry when a change of status occurs.

ID	Operation	Operand
* . .	STATUS	queue [,name[,number] ,HOLD ,priority ,ALLUSERS ,<u>ALL</u>]

queue specifies the queue to which the display command is directed in the xxyyy format, where

xx = partition to which the job entry was assigned for execution: BG, F4, F3, F2, or F1

yyy = RDR for reader
PRT for printer
PUN for punch

name specifies the name of the job entry about which status information is required. Each job entry name is qualified by a number assigned to it by POWER RJE. Therefore, if duplicate job entry names occur, number should be specified also.

number — specifies the number assigned to the job entry by POWER RJE when the job entry was logged in the input queue. This (1 to 5-digit) number is also used with output queue entries for this job entry.

HOLD — indicates that status information is requested only for those job entries which satisfy the command and are in the hold state.

priority — indicates that status information is requested only for those job entries which were assigned the same priority number. Priority is specified in the form Pn, where n = any number from 0 to 9. Nine is the highest priority.

ALLUSERS — indicates that status information is requested for those job entries which are in the specified queue and whose output destination is ALLUSERS.

ALL — indicates that status information is requested for all job entries which are in the specified queue and which the current terminal operator is authorized to receive.

Examples of commands

```
      * . .   RJSTART     NYTERM5,BRDCST
      * . .   LOGON       SHIFT2
 1            .
              .  Jobs from NYTERM5 entered under userid SHIFT2
              .

      * . .   STATUS      F4RDR,ALL
 2    * . .   STATUS      F4PRT,ALL
      * . .   STATUS      F4PUN,ALL

 3    * . .   DELETE      F4RDR,EXERCSE

 4    * . .   CONTINUE    PAGE,21

      * . .   OUTPUT      ASSYJD,2
      * . .   OUTPUT      ASSYJD,5
      * . .   OUTPUT      PL1COMP,3
 5    * . .   OUTPUT      ASSYKT
      * . .   OUTPUT      ASSYKG
      * . .   OUTPUT      PL1COMP1
      * . .   OUTPUT      PL1COMP6

      * . .   BRDCSTR
 6    * . .   MSGR        M,'DON''T UNDERSTAND LAST BROADCAST'
      * . .   MSGR        M,'MESSAGE. PLS HAVE ITS TEXT VERIFIED'
      * . .   BRDCSTR

 7    * . .   LOGOFF
      * . .   RJEND
```

1 The two commands indicate to POWER RJE that remote job entry is intended from terminal NYTERM5 under userid SHIFT2.

2 Upon entry of all jobs the terminal operator requests a status report.

3 If EXERCSE is to be processed after all, it must be resubmitted.

4 The printed output of a job entry was interrupted because, for example, the forms on the printer had to be re-aligned. Printing of the output is to continue with page 21.

5 Assemblies and compilations were entered with the output disposition specification H in the * $$ JOB statement. The terminal operator requests their output in accordance with installation defined priorities.

6 Before the terminal operator ends his session, he requests the transmission of messages whose destination is either his userid or ALLUSERS. Since the text of the last broadcast message was unclear to the terminal operator, he made an inquiry to the central operator. Two MSGR commands were necessary because his inquiry exceeded 40 characters. Two BRDCSTR commands are necessary because he expects retransmission of the last message.

7 Normal log off procedure.

Figure IV-8. **Examples of POWER RJE terminal commands.**

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 ,F1,F2,F3,F4,REL,NOREL}

IPL Reason Codes and ID-Codes

For details on how to enter these codes, see Procedure 5 in the section *Procedures*.

Reason Code	Comments
CE	IBM CE/SE has control of the system for maintenance or testing purposes.
DF	Default.
EN	Environmental problems (power, overheating, etc.) caused failure.
IE	IBM hardware or IBM-supplied programm error that did not require an IBM CE/SE.
IM	IBM hardware or IBM-supplied program error that required an IBM CE/SE.
ME	Media - hardware error caused by faulty disk pack, reel of tape, cards, etc.
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, etc. failure.
20	DASD - A failure occurred in a DASD unit or its associated control unit (2311, 2314, 2841, etc.).
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 (2401, 2803, 3400, etc.).
50	A failure occurred in a card reader/punch, a printer, or in its associated control unit (2540, 1403, 2821, etc.).
60	MICR/OCR - A magnetic ink character reader (1417, 1419, etc.) or an optical character reader (1285, 1287, etc.) failure.
70	A teleprocessing failure occurred in a teleprocessing control unit (2701, 2702, etc.).
80	Graphic - A video display unit (2260, etc.) 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.

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, 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 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 Models 115 and 125 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	,SEG=n	Delete n lines at a time	K SEG=4	When you enter K E,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.

Examples of the K Command (continued)

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 Models 115 and 125 only.

Controlling redisplay operation

Terminating redisplay mode

Command	Meaning
D D L D L,ALL	Enter redisplay mode for all messages
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.

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.

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.

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.

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, a cylinder 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: A program that traces a specified event when it occurs during the operation of a program. Abbreviated PDAID.

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.

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.

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 System Management Guide, GC33-5371

DOS/VS Serviceability Aids and Debugging Procedures, GC33-5380, abbreviated in this manual as DOS/VS SADP

System/370 Model 125 Procedures, GA33-1509

System/370 Model 135 Procedures, GC38-0005

System/370 Model 145 Procedures, GC38-0015

Index

Index

Index

Operator's Library
DOS/VS
Operating Procedures

GC33-5378-1

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

GC33-5378-1

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

Operator's Library DOS/VS Operating Procedures Printed in U.S.A. GC33-5378-1

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)

GC33-5378-1

Operator's Library DOS/VS Operating Procedures Printed in U.S.A. GC33-5378-1

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)

GC33-5378-1

Operator's Library DOS/VS Operating Procedures Printed in U.S.A. GC33-5378-1

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)