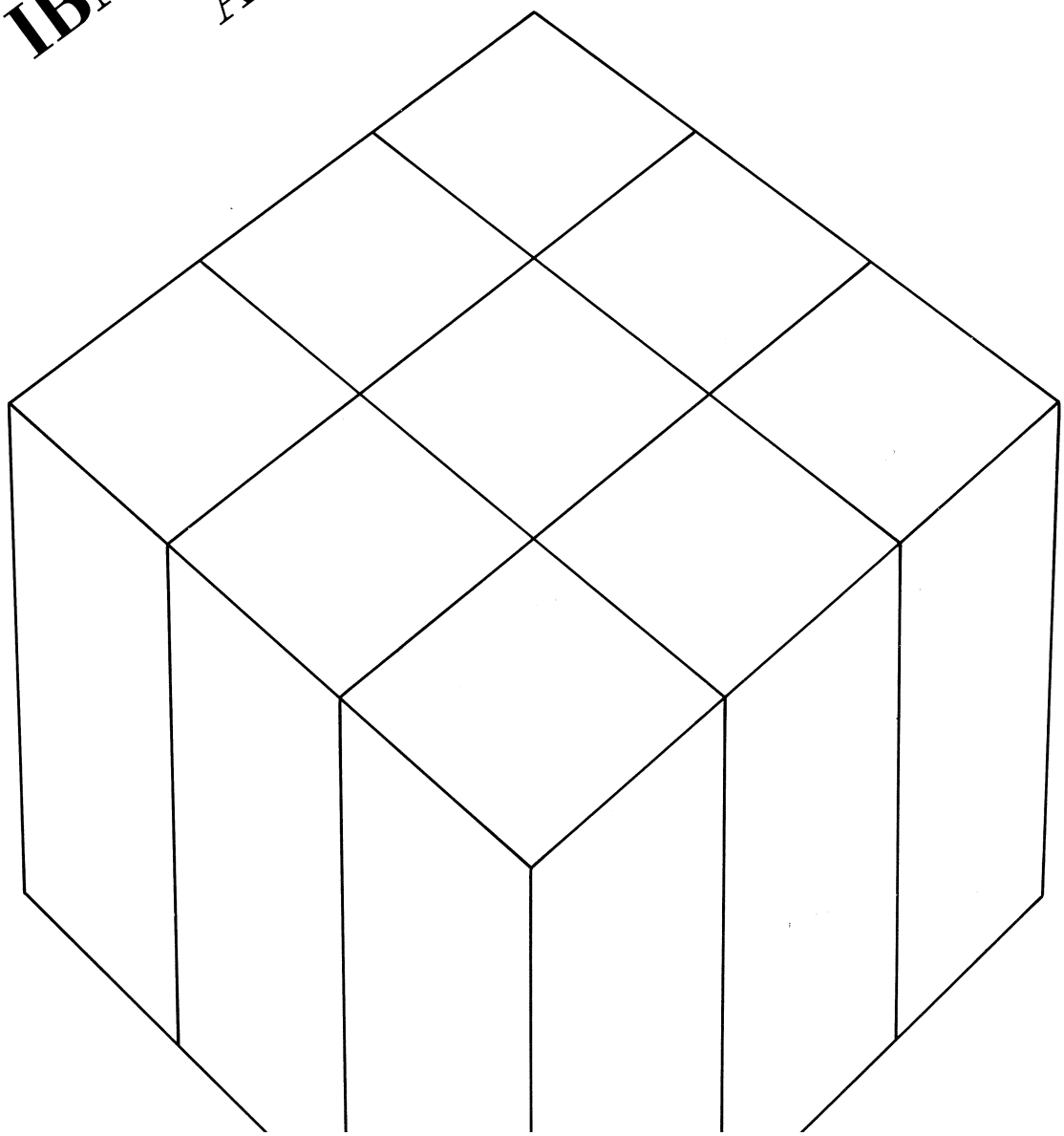
IBM Virtual Storage Extended Advanced Functions

Operation



IBM Virtual Storage Extended/ Advanced Functions

Operation

Version 3 Release 1

Program Number 5666-301

Order Number SC33-6194-02
File No. S370/4300-40

Third Edition (March 1987)

This edition is a major revision of SC33-6194-1. It applies to Version 3, Release 1 of IBM Virtual System Extended/Advanced Functions (VSE/AF), Program Number 5666-301, and to all subsequent releases until otherwise indicated in new editions or Technical Newsletters. Changes are made periodically to the information herein; before using this publication in connection with the operation of IBM systems, consult the latest *IBM System/370, 30XX and 4300 Processors Bibliography,* GC20-0001, for the editions that are applicable and current.

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM licensed program in this document is not intended to state or imply that only IBM's licensed program may be used. Any functionally equivalent program may be used instead.

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

A form for readers' comments is provided at the back of this publication. If the form has been removed, comments may be addressed either to:

IBM Corporation
Dept. 6R1BP
180 Kost Road
Mechanicsburg, PA 17055, USA

or to:

IBM Deutschland GmbH
Dept. 3248
Schoenaicher Strasse 220
D-7030 Boeblingen, Federal Republic of Germany

IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Preface

This manual contains the information required by system operators to run jobs under VSE/Advanced Functions. 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 matter and followed by back matter.

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

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

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 tells the operator how to do typical tasks. The procedures are illustrated by examples.

Control Statements contains a summary of all operator commands and control statements.

Back Matter contains a glossary, a bibliography, and the index.

Contents

Figures

Summary of Amendments

A complete overview of functions new since Version 1, Release 3, Modification Level 5 of VSE/Advanced Functions is given in the IBM publication *VSE/Advanced Functions Planning and Installation.*

The changes reflected in this manual cover:

- A new library structure, associated with the VSE libraries
- Conditional Job Control
- Parameterized Job Control
- A printer train cleaning procedure
- Support of the IBM 4248 Printer in native mode
- Optional suppression of recatalog commands in the output of the librarian PUNCH command.

Section 1. Introduction

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

The Operator's Responsibilities

All data-processing systems perform three major functions: they accept input, process this input, and produce output. You, as the operator, must see to it that your computing system performs these functions efficiently. This section tells you, in short, what you have to do to achieve this.

References in this manual to *the system* mean the operating system and the associated IBM and user-written programs. *The location* is the combination of the system, the hardware on which it runs, and the organization needed to make everything work efficiently.

The operator's tasks are divided into six groups. Not every operator has to do all of these tasks. Exactly what you will have to do depends on the way your location is organized. Regard this section as a general introduction to the work of an operator. Do not assume that all the duties outlined here will be expected of you. The possible groups of tasks are:

1. Starting up the system;
2. Preparing input for the system;
3. Controlling the processing in the system;
4. Handling output from the system;
5. Collecting information about the state of the system;
6. Stopping the system.

Starting the System

First, you have to switch the power on. You can find out how to do this in the manuals for the various devices you have to operate. You must also ready the input/output (I/O) devices (disk drives, tape and diskette units, printers, and card devices.)

Before work can begin, you must load the operating system. This is the set of programs which supervise all other activities in the system. It is stored on the system residence (or SYSRES), a disk volume. The operating system you are dealing with is VSE/Advanced Functions. To load it into the processor, do the initial program load (IPL) procedure. The IPL procedure you have to follow is laid down in detail for your particular location.

Preparing Input

The input consists of programs, data, and control information on tape, disk, diskette or cards. You will also enter control information at the console.

Programs are supplied by the programmers. The data with which the programs work may be stored on tape or disk by other programs, or entered by hand on diskette or cards.

As an operator, you are not concerned with programs or data. You only have to make them available to the system. You do this by mounting the appropriate volume on a tape, disk or diskette drive, or inserting cards is the card reader. You will be told which volume to mount by your system programmer or by a message at the console.

Most of the control information is supplied by the programmer. Sometimes this control information may be in error. If you receive a console message informing you of such an error, tell the programmer. He or she may then instruct you how to correct the input.

Controlling Processing

Processing takes place in the processor. The supervisor takes care of how the processing is done. However, you can stop processing when an error occurs, and start it again when the problem has been fixed. As directed by your system programmer, you can change system control values. These in turn affect what happens in the processor.

Handling Output

The output may be in the form of a printed report or punched cards, or it may be on diskette, magnetic tape or disk, or at your console.

Console output affects you as an operator directly. It may instruct you to mount a particular volume, or ask you to make a decision, or inform you of an error situation.

Output on magnetic storage devices merely requires you to do something at the appropriate device. You may have to insert additional diskettes in the diskette unit, or change tape reels or mount disk packs.

For printed output, you will sometimes have to change the paper in a printer. It may then be necessary to reload either the forms control buffer (FCB) or the universal character set buffer (UCB). The FCB controls the skipping of forms, the UCB controls printing with a universal character set print chain (or print train). You load both of these buffers by entering control statements at the console, as instructed by a programmer.

Handling output also means archiving tapes and diskettes, and distributing printed reports. How this is done varies from location to location.

Collecting Information

Errors may occur during input, processing, or output. 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 reel for a program. Programs or data that are required for processing may not be found on the specified disk packs or diskettes.

When such errors occur, the system usually issues a message. The person responsible for correcting the error will need these messages. You will find

them on the console hard copy log, if you have one. If not, use the D-command to browse back through the messages on your console display. You will find details on how to use this in the section "Communicating With the System" later in this manual. You can get a hard copy by printing the Hard Copy File.

During processing, the system may enter a wait state for no apparent reason. In this case, you may have to check to see if any replies to system messages are still outstanding. Here again, you can look at the console hard copy log, if you have one, or enter the attention routine command REPLID at the console.

You as the operator may also have to provide planning information for the programmers and administrators. This information may consist of listings of the contents of disk volumes or of libraries, or of the devices at present in use by the system. In the case of trouble with the hardware, you may have to run a program to gather data about the various devices. Your programmers will tell you what to do. There may be procedures specially tailored to your location.

Program checkpoints are another important kind of information. Long-running programs record the status of the job and the system on disk or tape at set intervals. If a program terminates abnormally and has to be run again, it can be restarted from one of these checkpoints.

Each time a checkpoint is written during the execution of a program, a message appears on SYSLOG. No action is required for these messages. However, be sure to save the messages, because the programmer needs the checkpoint numbers to restart the program.

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.

Stopping the System

Your system programmer will normally decide if and when the system should be shut down. From time to time, a shutdown will be necessary for maintenance purposes. The shutdown procedure depends on the size and complexity of your system. You may, for example, have to record or print the statistical data accumulated during the day.

The overview given in the preceding paragraphs, although simplified, applies to all data-processing locations. You are concerned with VSE/Advanced Functions and the integrated IBM and user-written programs, all of which make up your VSE system. To help you understand why you should perform certain operations, you should be familiar with some concepts of VSE/Advanced Functions. These are discussed in "System Concepts and Components" on page 5.

System Concepts and Components

A data processing system is a tool which can do many kinds of work. It can store information, or write bills, or keep track of items in a store. Exactly what the system does, depends on what programs run on it. There may be information retrieval programs, billing programs, or inventory programs, for example. These are called application programs.

Besides application programs, the system consists of control programs. These programs allocate resources to the application programs, and read the input and write the output for them. All these control programs make up your *VSE system.*

You, as the operator, are concerned mainly with the operating system. But remember, it is there to support the application programs, not as an end in itself.

Before we look at the operating system in more detail, let us look at the work it has to do.

Programs

As we have seen, the programs do all the real work. This may be printing the monthly pay checks, or it may be reorganizing data on the system's disk storage. Each program must go through three stages before it can be used. First, the programmer must write it in a programming language, for example, PL/I or COBOL. This is the **source program**. It must be in a machine-readable form.

In the second stage, the source program is processed by a **language translator** to produce an **object module**. This consists of instructions that the processor can execute.

In the last stage, the **linkage editor**, one of the system programs, prepares the program for loading into the processor. This final version of the program is called a **phase**. It is stored on disk until it is needed.

When a program is to run, it must be loaded, in phase format, into the address space of the **processor**. Here, the operating system causes the program to be executed one instruction at a time. When input has to be read or output written, the operating system attends to this, too.

Initial Program Load (IPL)

Before the operating system can start controlling programs, it must be loaded into the processor.

This is done by the IPL program, which means that this program must be executed each time the system is started. Executing the IPL program is one of your responsibilities as an operator. To prepare for IPL, you must power-on the system, make certain devices ready, and enter the address of the disk on which the IPL program is stored. Exactly how you do this with

your type of console is described in the "Procedures" section of this manual. The automated system initialization facility (ASI) then loads the system programs, defines I/O devices and starts up the system. ASI reads the necessary commands and control statements from a sublibrary. ASI is described in "3. Automated System Initialization (ASI)" on page 52.

Supervisor

The supervisor, which is loaded into processor storage by the IPL program, controls overall system operation. It coordinates the use of the system resources and maintains the flow of operations.

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 processing time, and handles much of the communication between system and operator.

Multiprogramming

Multiprogramming means using the idle time of one program to execute instructions of another program.

In a VSE system, you can have up to 12 programs running concurrently. The switching from one program to another is handled by the operating system. To make multiprogramming easier to handle, the address space of the processor is divided into **partitions**. One program can run in each partition at a time. To stop or cancel a program, you must give the appropriate command for the partition in which it is running.

Virtual Storage

The amount of **processor storage** on any system is limited. Often a system with 12 partitions has to deal with large programs. If the programs have large data areas, they will not all fit into processor storage.

The virtual storage concept allows you to define a storage area larger than the processor storage of your system. This larger area is the **virtual storage**. It is this virtual storage which is divided into partitions in the VSE system.

Remember that, even under multiprogramming, only one instruction of one program can be executed at one time. The virtual storage concept makes use of this fact. Only that part of virtual storage which contains the instructions to be executed at present is brought into processor storage. When this set of instructions is no longer needed, it is copied to a disk storage device.

Programs and data are not moved byte by byte between processor storage and disk storage. Virtual storage is divided into blocks called **pages**. Processor storage is divided into areas of the same size, called **page frames**. Programs and data are moved a page at a time. When loaded into processor storage, each page of virtual storage takes up one page frame.

The disk storage containing the pages which are not at present in processor storage is the **page data set**. The transfer of pages between the page data set and processor storage is **paging in**, the reverse is **paging out**. We use the word **paging** to mean transfer in both directions.

Job Control Program

The job control program is loaded into a partition automatically when no other program is being executed. It is loaded for the first time after IPL to prepare execution of the first program. Then it is loaded each time a program has completed execution or is terminated. This makes transition from executing one program to executing the next one automatic.

The job control program has many different functions. Not all of them are needed for every program execution. The programmer, or sometimes the operator, must therefore inform the job control program of the specific requirements for each program. These instructions to the job control program are given using job control statements and job control commands. The difference between these two types of job control information is explained in "Operator Commands" on page 100. The entire set of job control statements and job control commands is called the **job control language**.

The job control language is one of your chief means of telling the system what you want it to do. Full details of the job control commands and statements are given in *VSE/Advanced Functions System Control Statements*.

Jobs and Job Steps

Programs cannot simply be loaded into virtual storage and just run. They need input of some kind to work on, and produce some kind of useful output.

When a program is started, the system has to be told where the input is to come from, and where the output is to go to. The programmer responsible for the program does this, using **Job Control statements**. These statements, and a statement specifying which program is to be loaded (an EXEC statement), make up a **job**.

The start of a new job is indicated by a **JOB statement**, the end of the job by a **/& (end-of-job)** statement. The operating system creates certain statistics associated with the name given in the JOB statement.

Often programs work hand in hand. For example, a program may update a list of addresses. Another program may sort the list in alphabetical order, and a third may print the sorted list.

In such cases, the three programs are usually run in one job. This job consists of the input and output definitions and EXEC statements for each of the three programs. The input and output definitions and EXEC statement for one program are called a **job step**. So a job to update, sort and print the address list would have three steps.

A multi-step job also has a JOB statement at the start and a /& statement at the end. If you cancel the partition while an earlier step is running, none of the following steps are started. The operating system looks for the next /& (end-of-job) statement, and finishes work on that job. In our example, this makes sense. If the sort program is interrupted, there is no point in trying to print its output.

Sometimes the programmer may want a particular program called if one of the job steps is canceled, or if some other abnormal condition arises. He can prevent the job from being canceled and skip to the required step by using conditional job control.

Physical and Logical Units

Your computing system consists of more than just the processor with its processor storage. There are also **input and output (I/O) devices**. Your console is one of them. Other I/O devices are tape, disk and diskette drives, card units and printers.

Each I/O device has an address, so that the processor can access it. The address consists of a 3-digit hexadecimal number. The first digit indicates which channel the device is attached to, the second digit the control unit, and the last digit identifies the particular unit. This is the **physical unit address**. Each I/O device has a plate showing the physical unit address of the device.

In a program, the instructions dealing with I/O do not refer directly to these physical unit addresses. The programmer has no way of knowing, for example, which disk drives will be available when the program is run. Programs therefore address so-called **logical units** for input and output.

Logical units have 6-character names beginning with SYS-. There are **programmer logical units** named SYS001, SYS002, and so on up to SYS254. They are used mainly in application programs.

System logical units are named SYSRDR, SYSLOG, SYSLST, and so on. These are used mainly by the operating system. The names of the system logical units and their use are shown in Figure 1.

Logical Unit	Device Type	User for
SYSRDR	card reader, magnetic tape unit, diskette or disk extent	reading job control statements or commands
SYSIPT	card reader, magnetic tape unit 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 (see note)
SYSLST	printer, magnetic tape unit, diskette or disk extent	"printed" output of the system (see note)
SYSLOG	console printer keyboard or display console	communication between the operator and the system, and logging of job control statements
SYSLNK	disk extent	input to the linkage editor
SYSRES	disk extent	system residence device
SYSREC	three disk extents	storing error records collected by the recovery management support recorder (RMSR) functions. Also for storing messages to the operator for subsequent printing, and for the System History File.
SYSCAT	disk extent	VSE/VSAM catalog

Figure 1. System Logical Units and Their Use

Note: "Punched" and "printed" refer only to the format of the respective output, and are independent of the storage medium used.

When a program is to be run, its logical units are assigned to actual I/O devices of the appropriate types. This assignment is done with the **ASSGN** job control command. This command connects the logical unit name with a physical unit address, for example:

```
ASSGN SYSLST,00E
```

Here, the system logical unit SYSLST is assigned to the printer with the physical unit address 00E.

Sometimes, it may not matter which physical unit SYSLST is assigned to, as long as it is a printer of the PRT1 type. In this case, you can enter:

```
ASSGN SYSLST,PRT1
```

The logical unit SYSLST is then assigned to the physical address of the first PRT1 printer that the operating system finds to be free. PRT1 is the *device type.*

If it does not matter what type of printer is used for SYSLST, you can use the command:

```
ASSGN SYSLST,PRINTER
```

SYSLST is then assigned to the first free printer of any type. PRINTER is the *device class*.

There is one other way to assign a logical unit. You can assign it to the same physical unit address as another, already assigned, logical unit. For example, a program might direct its printed output to logical unit SYS005. To have this output directed to the same printer as the system output, you do not even have to know which unit SYSLST is assigned to. Simply enter:

```
ASSGN SYS005,SYSLST
```

and the job control program associates SYS005 with the physical unit assigned to SYSLST.

The same holds true for other kinds of device, of course. Tape, disk, and other devices can be assigned by physical address, device type, device class, or by reference to another logical unit, too.

Logical units can be assigned **permanently** or **temporarily** to physical units. A permanent assignment remains in effect until the next IPL, or until another assignment is made for the same logical unit. A temporary assignment remains in effect until the end of the job in which it is made, or until another temporary assignment is made for the same logical unit in the same job. At the end of the job, the permanent assignment for the logical unit takes effect again.

An ASSGN command is valid only for the partition in which it is given.

Records and Files

Let us have a look at what programs do with the physical units assigned to their logical units. When a program accesses a disk, diskette or tape, it reads or writes data. The sets of data which programs work with are called **files**. Programs do not read or write a whole file at once. Files consist of **records**, and programs process them record by record.

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 terms are best explained by an example. Think of names and addresses stored on tape. The name, street and city may be thought of as items of data. Each complete name and address is a record, and the collection of names and addresses is a file.

Labels

It is part of your task as an operator to make data available to programs. Data is stored in files on disk packs, diskettes, and reels of magnetic tape (also called volumes). These volumes are normally labeled on the outside for easy identification. You must mount the required volume on the device assigned to the appropriate logical unit.

However, volumes often contain more than one file. It is therefore necessary for each file on a volume to be identified so that the operating system can find the right one. **File labels** provide this identification. Whenever a programmer creates a file, he can specify the contents of the file label. This label is then written onto the disk, diskette, or tape just like a data record. When the file is processed as input, the programmer specifies the contents of the label in a job control statement. The operating system compares the specified data with the actual label. If the system does not find an actual label matching the label information supplied by the programmer, you get an error message at the console.

File labels also **protect** files. They contain the date of creation and expiration of the file. Normally, a file cannot be destroyed, that is, overwritten by another file, before the expiration date. If a non-expired file is to be overwritten, you, the operator, must enter the appropriate response to a system message.

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.

In addition to a physical label (usually a sticker), each volume (disk pack, diskette, or tape reel) is further identified by the **volume label**. This label is written by a utility program when the volume is prepared for use for the first time.

Disk and diskette volumes must have labels. For tapes, labeling is optional. If a tape is to be labeled, a utility program, the Initialize Tape Utility, is used to write a volume label on the tape. File labels are written before and after 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, MTC, is used to position the tape correctly.

System Utilities

Utilities are supplied by IBM to **help you** run your VSE system. For example, there are utilities to initialize tapes or disk packs. Others print system information, or help you to clean your printers.

Utilities are **programs**. Like application programs, they run in the partitions, and are started with ASSGN and EXEC job control statements.

Some of the utility programs help you in your work as an operator. The utilities for **operation** are described in Section 3 on page 47.

Job Control Procedures

Programs need job control statements to start them. This is true of application programs and utility programs. For applications, writing the job control is the task of the programmer responsible for the program.

What about utility programs that you as the operator have to use? This manual gives you some help, but you will probably not have to write the job control at all. Your system programmer can do this for you, and you can use his job control whenever necessary.

The job control statements for a frequently-used program can be stored as a **procedure**. The procedure consists of all the statements needed to define the I/O for the program, and execute it. A procedure may contain one or more job steps.

(Here, we are talking about job control procedures. In data processing, we refer to any hard and fast sequence of actions as a procedure. The IPL procedure, for example, calls for you to mount the SYSRES volume, ready certain devices, and enter certain values at the console, always in the same order. Each time it is run, a job control procedure makes the operating system take the same actions in the same order.)

The job control procedures written for your installation are **stored on disk**. How to use them is described in "18. Using Cataloged Procedures" on page 80.

Libraries

We have seen that job control procedures are stored on disk. So are the programs they call. Data are often stored on disk, too. There are differences between a program or procedure and a data file. The main one is, that programs and procedures are used by the operating system, whereas data files are usually for use by application programs.

The purpose of the operating system is to control the work as quickly as possible. Its input must be stored so as to be available quickly whenever it is needed. The **libraries** serve this purpose in the VSE system.

A library is basically a file on disk storage. But it does not contain just records, like a data file. It is divided into **sublibraries**, which contain **members**. The members may be, for example, job control procedures, programs or dumps. Procedures and programs are **cataloged** into members, and are **searched** for using the **member name**. Besides the name, each member has a **type**. To find a procedure, for example, the operating system searches through all members of the type PROC until it finds the specified name.

The possible member types and their use are as follows:

PROC for job control procedures;

PHASE
for phases (programs processed by the linkage editor);

OBJ for object modules (programs processed by a language translator);

DUMP for dumps;

A..Z,0..9,$,#,@
(a 1-character type) for source programs. Members of these types are referred to collectively as "members of the type SOURCE." In certain job control statements, using the keyword **SOURCE** is the only way to refer to source programs.

In addition, any freely-chosen type different from the above may be used for your own data. It must be no longer than 8 characters.

This is how a library is built up. First, label information for a disk file is entered in the system. This information includes the chosen size and the filename. Files defined using the access method VSAM can have a basic size that can be extended when more space is needed. Then this file is defined to the operating system as a library. Before the new library can be used, at least one sublibrary must be defined in it. The sublibrary contains no data. It is just a directory for finding the members. Members can now be cataloged in the sublibrary.

A library always keeps its original size. The size of a sublibrary is the sum of the sizes of its members, plus the space needed for its directory.

The library space not taken up by the sublibraries is **free space**. It can be used to catalog members in any of the sublibraries.

The Librarian

The component of the operating system which handles the libraries is the **librarian**.

The librarian performs such functions as:

- Creating a library or sublibrary
- Displaying the contents of a library or sublibrary
- Adding members (phases, modules, etc.) to a sublibrary
- Deleting members from a sublibrary.

The librarian is a program. To use it, you call it like any other program:

```
EXEC LIBR
```

Then you enter the appropriate librarian commands for the functions you need. You can do this in the form of a job, as with other programs, or you can enter the EXEC LIBR statement at your console. At the console, the librarian then prompts you to enter the commands one at a time.

As an operator, you will probably use the librarian commands most frequently to gather information about the libraries. For this, you will need the LIST and LISTD commands. You may also use the CATALOG command to catalog job control procedures for routine jobs that have to be done frequently. The MOVE, COPY and UPDATE commands also help you to manipulate the sublibrary members you will be using.

The librarian commands and their functions are described in detail in *VSE/Advanced Functions System Control Statements.*

Section 2. Communicating with the System

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

For details on the syntax of statements and commands, refer to *VSE/Advanced Functions System Control Statements.*

System-to-Operator Communication

The system communicates with you by issuing messages on SYSLOG, the logical unit to which the console is assigned.

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.

Replies to messages requesting an operator response need not be given in the same sequence in which these messages were displayed on the console. Thus, you need not respond to a message immediately; you can defer your reply while receiving or answering other messages or entering a command.

Each message is preceded by a three-digit reply-ID. To enter a reply to a message (either immediately or later), you must type in the reply-ID that the system has assigned to this message. You may omit leading zeros in the reply-ID for convenience. The reply-ID is part of the message prefix, which consists of:

- An asterisk (*) if the message still has a reply outstanding;
- The partition identifier;
- One character indicating whether a reply is required. The character may be:
  - A blank, indicating that no reply is required,
  - A minus sign (-), indicating that a reply is required,
  - A plus sign (+), indicating that an immediate reply is required to avoid performance degradation.
- The reply-ID.

For example, the prefix:

```
* BG - 001
```

would indicate that the message is from the background partition (BG) and requires a reply (-). The reply has not yet been entered (*), and you must type in the reply-ID '001' when you do enter it.

Grouped according to the action indicators, there are four types of message.

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

The last position of each message number indicates which type of message it is. For example:

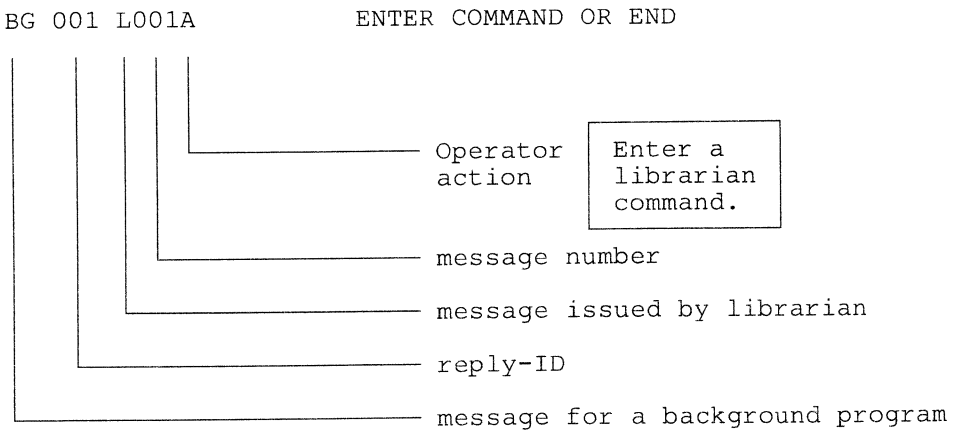
```
L001A ENTER COMMAND OR END
```

is an **action** (A) 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 messages, including operator action and responses, are contained in VSE/System Package Messages and Codes.

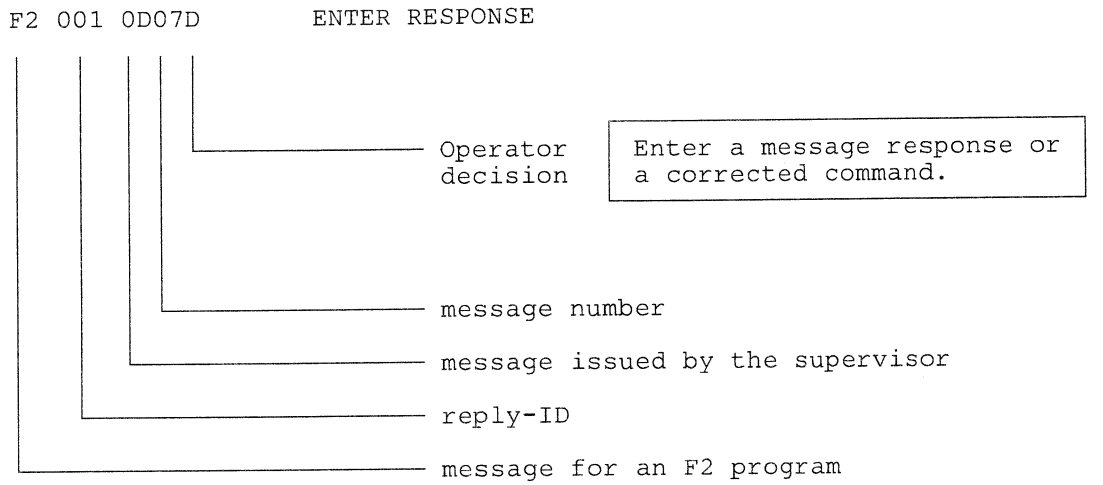
Action Message

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



Decision Message

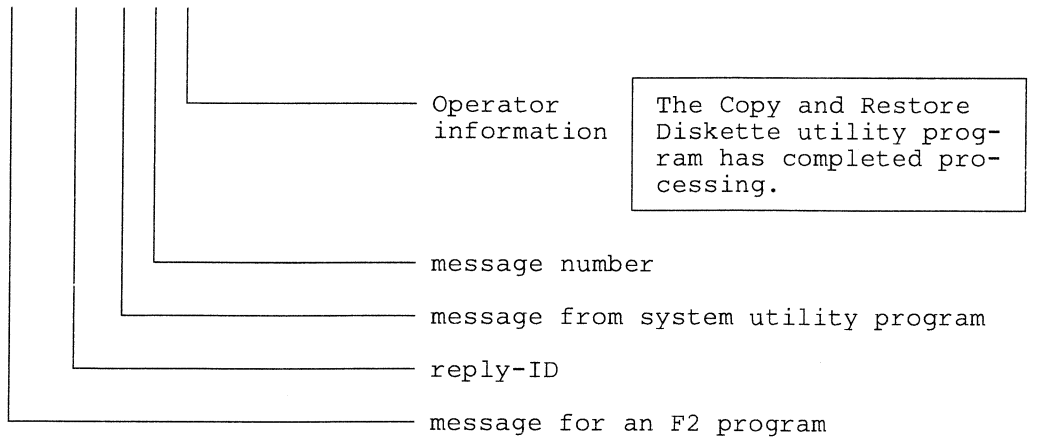
The message number is followed by the letter D, indicating that you must make a choice between alternative courses of action.



Information Message

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

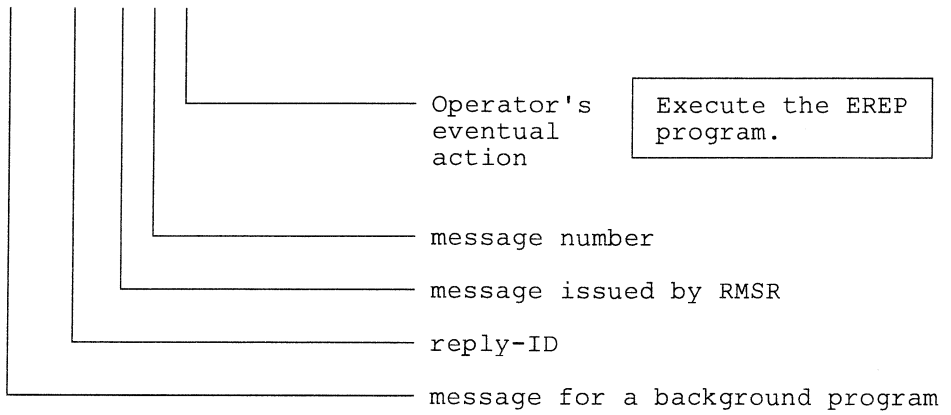
```
F2 001 8512I          END OF COPY/RESTORE
```



Eventual-Action Message

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

```
BG 001 0T05E          RECORDER FILE FULL. RUN EREP
```



Hard Wait Codes

Occasionally, the processor stops executing instructions, and the WAIT indicator at the console comes on. This happens, for example, when the system is waiting for the completion of an I/O operation, or for the reply to an action or decision -type message, or when no programs are active. The system still accepts interrupts from the console. This is a normal wait state, and not an error condition.

However, there are errors that put the system in a hard wait state. The WAIT indicator on the processor console is on, and the system does not accept any interrupts from the console. Usually, a code in bytes 0-3 of processor storage identifies the type of error that is causing the hard wait. Retrieval of the code from bytes 0-3 is described in detail in *VSE/Advanced Functions Service Aids*.

Operator-to-System Communication

You can communicate with the system by means of either job control statements or 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 may be punched in cards and entered at a card reader (SYSRDR), or be in card-image format on tape, disk, or diskette. You can also enter them at the console. They are accepted by the system only between jobs or job steps.

Operator commands are entered from the system console (SYSLOG). All commands must be preceded by a partition identifier and followed by pressing the ENTER key.

In addition, **job control** commands must have a reply-ID after the partition identifier. The system displays the valid reply-id's on the console. Do not forget to type in the appropriate reply-ID in front of your job control commands, even when a message prompts you to "Just press ENTER".

The following types of operator commands are summarized briefly in this manual:

- IPL commands, which are accepted only while you perform the IPL procedure. These commands prepare the operating system for operation.
- Attention routine commands, which are accepted at any time from the console. (If an AR command is not accepted, enter the command **RC** (Request Communication), after which a message informs you that the attention routine is available.) Attention commands from a display console may be entered without pressing the PA1 key (on the 3277, 3278), or PF1 key (on the Model 158 and 3031 consoles.)
- Job control commands, which are accepted only between jobs or job steps.

These commands are described in detail in *VSE/Advanced Functions System Control Statements.*

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

- **Set system values.**
  - The ADD command makes devices available to the system;
  - The DEL command makes devices unavailable to the system;
  - The DEF command identifies the disk drive on which the disk volume containing the VSE/VSAM catalog (SYSCAT) is mounted, and the physical device to which SYSREC is assigned;

- The SET command sets the values for date and time;
- The DPD command defines the page data set;
- The DLA command is used to define a separate label area;
- The DLF command to define the system lock file;
- The SVA command allocates space within the shared virtual area;
- The SYS command allows you to influence the paging activity and the use of SVA space.

These commands are valid during IPL.

- **Temporarily suspend processing**. The PAUSE command causes a partition to pause between jobs or job steps, enabling you to take action such as mounting the next tape reel or disk pack.
- **Cancel jobs**. The CANCEL command terminates the execution of a job.
- **Change I/O device assignments**.
  - The ASSGN command assigns a physical I/O device to a logical unit name;
  - The DVCDN (device down) command informs the system that an I/O device is inoperative;
  - The DVCUP (device up) command informs the system that an I/O device which 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.
- **Reset system defaults**. The STDOPT command changes the default job control options which were established at system initialization.
- **Control magnetic tape operations**. The MTC command controls magnetic tape operations, such as rewinding tapes, skipping files, writing tape marks.
- **Close 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 displays a list of current I/O device assignments in the system;
  - The VOLUME command shows which disk volumes are currently mounted;

- 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;

- The MAP command displays the allocation of storage areas in a multiprogramming system.

- **Allocate storage and control multiprogramming.**

  - The ALLOC command enables you to allocate virtual storage to the partitions;

  - The ALLOC R command enables you to allocate processor storage to partitions (for real-mode execution);

  - 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, as well as the size of the supervisor, SVA, and the amount of free space in virtual storage;

  - The PRTY (priority) command enables you to modify the priority of a partition.

Although the normal communication device for operator commands is SYSLOG (assigned to the system's console), commands may also be entered from SYSRDR, with the exception of attention commands.

Example of Operator-to-System Communication

The log example provided in Figure 2 on page 25 shows how replies to messages may be deferred.

If a message consists of a prefix only (without message number and message text), you can look up the complete message under the same prefix earlier in your console log. If you operate a display console, you can re-display the whole message by entering the D command with the respective partition identifier (see Screen Operation, later in this section).

If a previously displayed message is to be answered and this message has disappeared from the screen, then message 0D16D READY is displayed together with the message prefix.

Message replies are entered in the same way as attention commands. The only difference between a reply and a command is that the reply is preceded by the message-related reply-ID (with or without leading zeros) plus one blank.

Only one reply per active task in the system may be outstanding at a time. If a task tries to issue a message while the reply to another message (issued by the same task) is still pending, you will be notified by message

0D13D RESPOND TO REPLY-ID nnn

where nnn is the reply-ID of the message for which no response has been entered.

Operator communication support includes the REPLID command which can be used to display the prefixes of all messages for which replies are still outstanding (for details, see *VSE/Advanced Functions System Control Statements*).

If you want to enter an attention routine command (for example CANCEL), and the attention routine is not available, enter the RC (Request Communication) command, after which the messages

```
1P44I PREVIOUS 'command-name' COMMAND IGNORED
```

and

```
1I40I READY
```

are displayed. You can now enter any attention routine command.

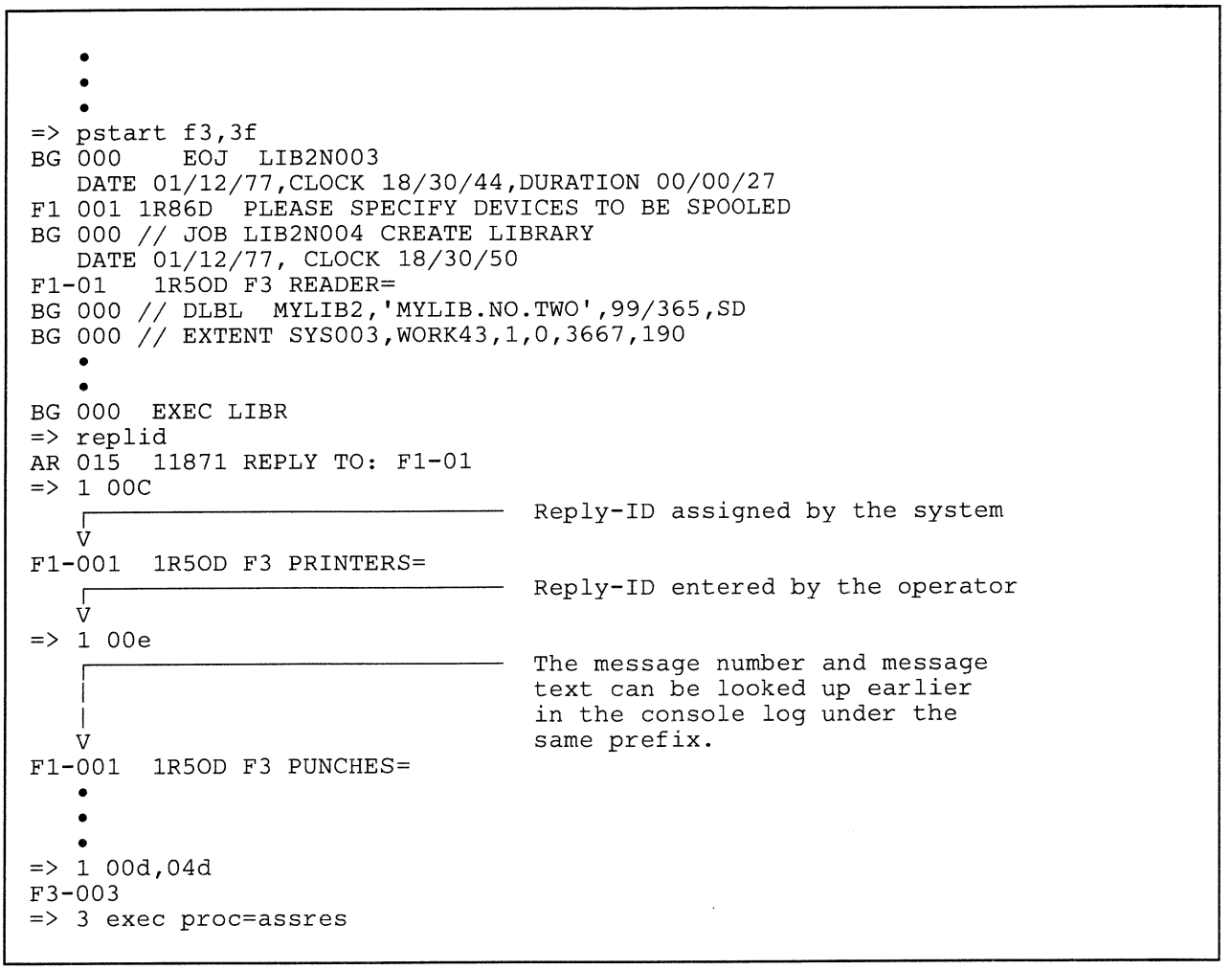


Figure 2. Log Example for Deferred Replies to Messages

Devices Used for Operator/System Communication

This chapter deals with the following types of devices that are available for operator-to-system and system-to-operator communication:

1. The IBM 3210 and 3215 Console Printer-Keyboards, which are used in conjunction with IBM System/370 Model 155-II.
2. The 3277, 3278 and 3279 Display Consoles.
3. The 3278 Model 2A Display Console
4. The 3279 Model 2C Color Display Console
5. The Console Display-Keyboard used with IBM System/370 Models 138 and 148.
6. The Operator Consoles used with the Models 158 and 3031.

IBM 3210 and 3215 Console Printer-Keyboards

The console printer-keyboard allows you to enter:

- Operator commands;
- Job control commands or statements;
- Responses to messages that are printed on its printer.

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.

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 in Figure 3 shows how these indicators and control keys are arranged and Figure 4 on page 28 and Figure 5 on page 28 summarizes their functions.

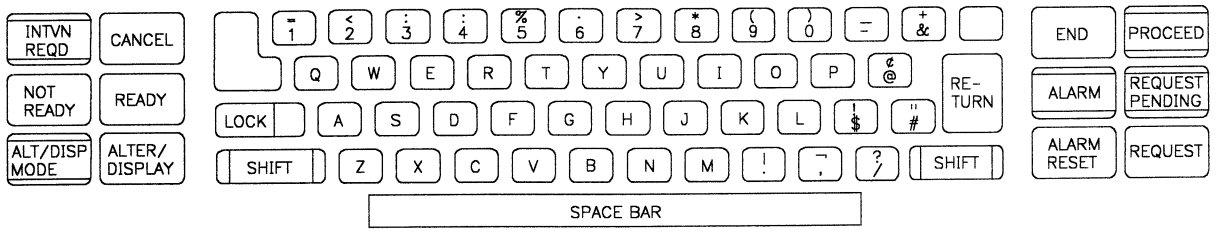


Figure 3. 3210 and 3215 Keyboard Layout

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

Figure 4. Functions of Indicators of the IBM 3210 and 3215

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 system to accept keyboard entry.

Figure 5. Functions of Control Keys of the IBM 3210 and 3215

3277, 3278 and 3279 Display Console

This section does not apply to the 3278 Model 2A and 3279 Model 2C Display Consoles. The next two sections deal with these consoles.

The 3277, 3278 and 3278 display console devices are supported as operator consoles on all System/370 models to which they are attachable.

The 3277 display console is used to enter operator commands, job control statements, and responses to messages that are displayed on its screen. The upper 20 lines of the screen are used as message area, and lines 21-24 are used as instruction line, entry area, and warning line (see Figure 6).

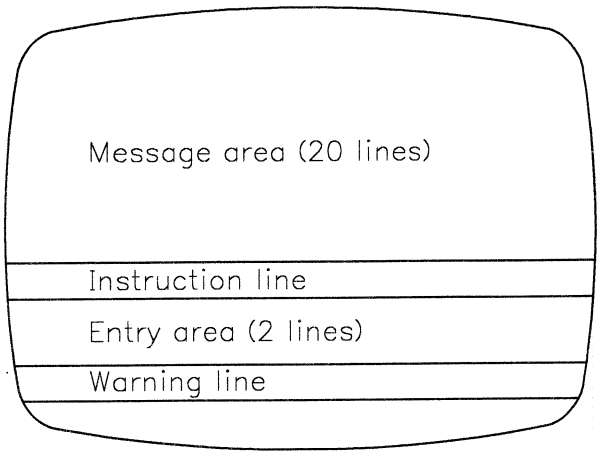


Figure 6. Layout of the 3277 Screen

Note: The solid lines and words do not appear on the 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** is for entering commands and replies.

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

Messages longer than 77 characters are continued on the next line.

The system displays action messages and outstanding reply requests in high intensity, all other messages in normal intensity.

The system resets reply requests to normal intensity when you have answered them. Action messages are not reset.

Functions of PA1 Key on 327x

In case of heavy traffic on the screen, enter PA1, after which 0D07D ENTER RESPONSE appears; then enter a command or reply, but not a K-command or a D-command.

3278 Model 2A Display Console

The 3278 Model 2A display console device is supported as operator console on the 4300 Processors. It is functionally equivalent to the 3277. However, the display unit of the 3278 has a message area of 16 lines. Therefore, once IPL has been performed on a 3277, a reassignment of SYSLOG to a 3278 is not allowed.

If the 3278 console screen has a message area of 16 lines, the last five lines display information about the status of the hardware; they are mainly used for service purposes by the IBM customer engineer (see Figure 7).

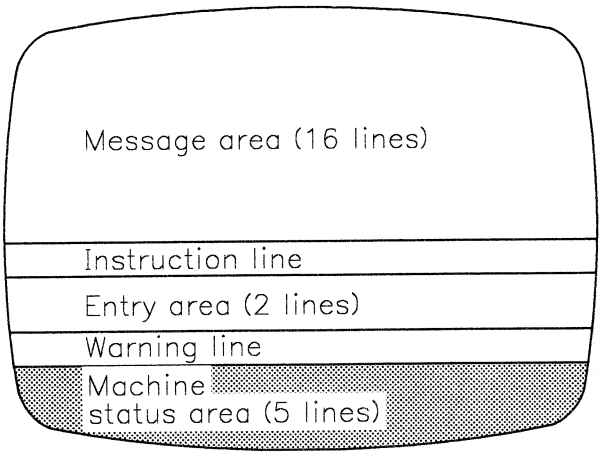


Figure 7. Layout of the 3278 Screen

Note: The solid lines and words do not appear on the 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** is for entering commands and replies.

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

The system displays actions messages and outstanding reply requests in high intensity, all other messages in normal intensity.

The system resets reply requests to normal intensity when you have answered them. Action messages are not reset.

3279 Model 2C Color Display Console

VSE/Advanced Functions Version 2 Release 1 supports the 3279 Model 2C Color Display Station as operator console. Action messages and outstanding reply requests are displayed in white, all other messages in blue. Console input is displayed in green.

The system resets requests from white to blue when you have answered them.

Action messages, on the other hand, remain in white, because the console is not notified of the responses to these messages.

The 3279 Model 2C is functionally equivalent to the 3277. However, the display unit of the 3279 has a message area of 16 lines. Therefore, once IPL has been performed on a 3277, a reassignment of SYSLOG to a 3279 is not allowed.

The last four lines display information about the status of the hardware; they are mainly used for service purposes by the IBM customer engineer (Figure 8).

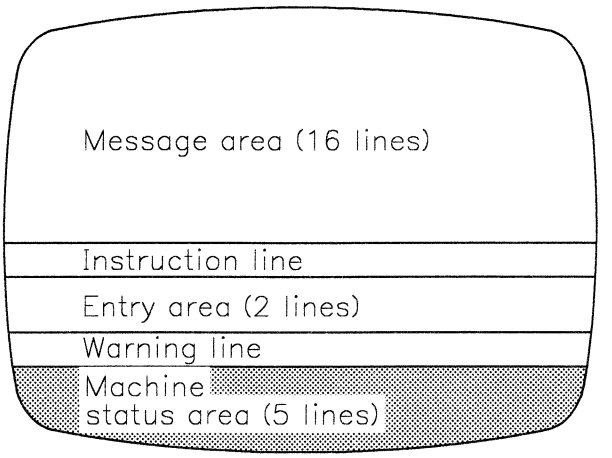


Figure 8. Layout of the 3279 Screen

Note: The solid lines and words do not appear on the 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** is for entering commands and replies.

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

Model 138/148 Console Display-Keyboard

The /370 Models 138 and 148 are operated and controlled through a system control panel on the CPU and a standard display console consisting of a display unit and an operator keyboard. Operator input is entered through the alphameric keyboard (and, at the same time, displayed on the display unit), and messages are displayed on the screen. The physical capacity of the screen is 24 lines with 80 characters each.

Console support is provided in two modes:

- Printer-keyboard mode
- 3277 display mode

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

Printer-Keyboard Mode

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

Each message requires two lines: the partition identifier occupies the first line and the message text the second line. When the output area (23 lines) is full, the top six lines are automatically deleted (by the hardware) and the remaining lines are moved to the top of the screen, thus leaving space at the bottom for fresh information. The lines that are deleted cannot be recovered, because a hard-copy file on disk is not supported in printer-keyboard mode. However, if an optional console printer is installed, all lines that appear on the screen are also printed on the console printer. This provides you with an immediate hard-copy of all messages and operator commands.

Note: Pressing the CLEAR key causes the entire display to be cleared, and positions the cursor at the left upper corner. Press the ENTER or PA1 key to redisplay the message area as it was before the CLEAR key was pressed.

3277 Display Mode

This console mode is equivalent to the 3277 support described under "3277, 3278 and 3279 Display Console" on page 29.

Model 158 and Model 3031 Operator Consoles

The operator consoles of the Models 158 and 3031 are functionally equivalent to the 3277 Display Console. The only difference is that the Model 158 and 3031 consoles do not have a PA1 key, but use the PF1 program function key instead. (Press the SEL key and then the PF1 key.)

Screen Operation

The screen operation described in this chapter applies to the 3277,3278 and 3279 Display Consoles, except where explicitly stated otherwise. Screen operation consists of deleting and redisplaying messages on the screen.

Normally, all lines that you delete from the screen are automatically stored in the the hard-copy file. This is the disk extent assigned to SYSREC. If the display unit is being used in printer-keyboard mode, however, the deleted lines are not stored.

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

0D20E HARD COPY FILE SHOULD BE PRINTED

If you wish to save the messages, you must print them on SYSLST directly after this message. How this is done is described in the procedure "23. Printing the Hard-Copy File (PRINTLOG)" on page 89.

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

0D25E 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.

If your installation uses a console printer, all lines that appear on the screen are also printed on the console printer, thus providing you with a 'hard' copy of messages, operator replies, 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 your system.

Deleting Messages

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

```
K   [ S          ]
    [ E  (options) ]
    [ D          ]
```

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.

Message Deletion Modes

The S operand of the K command is used to establish and display message deletion modes and to activate and deactivate console printer support for the 3277/3278 operator consoles. It has the following format:

```
K  S  [,DEL= Y ]  [,CON= N ]  [,ALM= Y ]  [,SEG= 6 ]
      [      N ]  [      Y ]  [      N ]  [      n ]
      [,REF     ]
```

All operands are optional; if they are omitted, the current (default or previously specified) values remain unchanged.

Automatic Deletion Mode

In automatic deletion mode, the deletable messages that appear in the first 15 lines on the 3277/3278 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. (Some VSE/POWER messages, which are not automatically deleted, must be deleted manually.)

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 (unless you specified ALM = N). 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" on page 39).

Non-Conversational Deletion Mode

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

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" on page 39).

Audible Alarm

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

Setting the Deletion Range

The SEG option allows you to specify the number of message lines to be deleted at a time by the K command. The range is 1-16 for the 3278 and 3279, and 1-20 for the 3277.

Activating the Console Printer

The PRT option allows you to activate and deactivate the console printer support for the 3277/3278/3279 operator console, if your system is equipped with such a printer. Normally, the console printer is activated at IPL time.

Displaying and Changing the Deletion Modes

You can use the REF option to display and subsequently change the deletion modes currently in effect. The command

K S,REF

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

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

The values shown in this example are the defaults set by the system. The PRT option will be displayed only if your system has a 3277/3278/3279 operator console with a printer. You can now change the default deletion modes either by the cursor or by entering a K command with the S operand. For example, to change from automatic to manual deletion mode and to specify a deletion range of four lines:

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

- enter the command K S,DEL = N,SEG = 4.

Note that the other default parameters remain unchanged: non-conversational mode (CON = N) is still in effect, and the audible alarm (ALM = Y) remains enabled.

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:

```
K  E  [ ,SEG ]
      [ ,n   ]
      [ ,n,n ]
      [ ,N   ]
```

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.

Deleting a Single Line

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

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

- one digit (or two digits) from 1 to 16 (for the 3278)

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. A-, D- and E-type messages within the specified range are not deleted. Messages of this type are deleted only when directly pointed to by a command or by the cursor.

Deleting Line Numbers of Messages

The N option of the E operand specifies that the message line numbers in character positions 2 and 3 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.

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.

If a long message is to be displayed on the screen for which there is not enough space, delete all preceding messages. If there is still not enough space, the message CONTINUE BY ENTER is displayed. If you now press ENTER, the remaining part of the message is displayed at the top of 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 line numbers in character position 2 and 3 by means of the D operand of the K command. The D operand has the following format:

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

Temporary Display

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

Permanent Display

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

Redisplaying Messages

You can redisplay messages that were previously deleted from the screen. 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 │ │ ,OCALL       ││
        │    │ ,AR  │ │ ,REPLY       ││
D       │ L  │ ,BG  │ │ ,ROUTED      ││
        │    │ ,Fn  │ │ ,SUPPRESSED  ││
        │    └      ┘ │ ,SYSCONS     ││
        └             └              ┘┘
```

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

AR specifies redisplay of attention-routine messages.

BG specifies redisplay of messages from the background partition.

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

The following operands are meaningful only if VSE/OCCF is installed in your system. They allow you to redisplay messages selectively in accordance with the action specified in the message action table of VSE/OCCF:

OCALL
specifies redisplay of all messages without selection and in accordance with the first operand specified.

REPLY
redisplay of all messages for which replies were defined in the message action table together with their replies.

Note: If you specify both one of the operands AR, BG, or FN and one of the operands SYSCONS, NCCF, SUPPRESS, or REPLY, then only those messages will be redisplayed which satisfy both conditions.

ROUTED
redisplay of all messages directed to an NCCF operator station and their replies and all commands entered from there.

SUPPRESSED
redisplay of all messages which were suppressed by VSE/OCCF.

SYSCONS
specifies all messages directed to the system console and any replies or commands entered on the system console. (This is also the default if VSE/OCCF is not active.)

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

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:

MESSAGE REDISPLAY BWD ALL=nnnn [option] [,option] = nnn

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

option
indicates the operand(s) you have specified in the D command.

nnnn is the line count for all messages and for the messages pertaining to 'option'. nnnn is displayed only as long as the above options are not changed or respecified. The display of nnnn is resumed after restarting redisplay with the D L,R command (see next section).

Note: Do not forget to press ENTER after you have typed in the D command, since otherwise the system would eventually go into the wait state without giving you any 'MESSAGE WAITING' indication.

Controlling Redisplay Operations

When you start redisplay, the screen is filled with the messages most recently stored on the hard-copy file and according to the option(s) specified in the D command.

Whenever you then press ENTER, another 20 or 16 message lines (depending on the console device type) 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 the D command in one of the following formats:

```
D  [  L  [ ,ALL  ]   [ ,OCALL    ]  [,R] ]
         [ ,AR   ]   [ ,REPLY    ]
         [ ,BG   ]   [ ,ROUTED   ]
         [ ,Fn   ]   [ ,SUPPRESS ]
                     [ ,SYSCONS  ]
```

```
D  [  L  [ ,B  [,nnn] ] ]
         [ ,F  [,nnn] ]
         [ ,R         ]
         [ ,nnn       ]
         [ ,16        ]
```

B - changes direction of redisplay to backward.

F - changes the direction of redisplay to forward.

R - causes redisplay of message to be restarted (from the point where you entered redisplay mode) according to the currently specified options.

nnn - specifies the number of messages to be spaced forward or backward; if nnn is not specified, 16 is assumed for the 3278, and 20 for the 3277.

The other operands are described above, under "Entering Redisplay Mode".

All operands are optional. If you just specify D, the default is D L,16 (for the 3278), or D L,20 (for the 3277) plus the redisplay direction and the option currently in effect. Pressing ENTER has the same effect as specifying D without any operands.

Note: If you want to change to the opposite direction of redisplay without spacing forward or backward at the same time, specify D L,F,0 or D L,B,0.

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

1. To display 16 (3278), or 20 (3277) background messages at a time, press ENTER; direction is backward.

2. To change to forward only specify: D L,F,0.

3. To space forward 20 messages, specify: D L,20.

4. To display messages for F1, starting with the currently displayed messages, specify: D L,F1.

5. To restart the redisplay of messages according to the current options, specify: D L,R. You are still in redisplay mode.

6. To display messages directed to the system console, specify:
   D L,SYSCONS.

Terminating Redisplay Mode

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

```
D E
```

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

Section 3. Procedures

This section tells you how to carry out some important tasks required of a VSE system operator. The procedures are illustrated by examples. Each procedure is numbered for quick reference.

The individual procedures are listed below.

1. Preparing the Printer for Buffer Load
2. Printer Train Cleaning
3. Automated System Initialization (ASI)
4. Starting IPL with the IBM 3210 or 3215 Printer-Keyboard
5. Starting IPL with the IBM 3277/3278/3279
6. Starting IPL with the IBM Model 158 Display Console
7. Starting IPL with the IBM Model 3031 Display Console
8. Interactive IPL at the Console
9. Creating the System Recorder File
10. Creating the Hard-Copy File
11. Loading Print Buffers
12. Setting Defaults for an IBM 3800 Printing Subsystem
13. Assigning I/O Devices
14. Allocating Storage
15. Starting the Background Partition
16. Starting a Foreground Partition
17. Interrupting or Terminating Processing
18. Using Cataloged Procedures
19. Displaying the VTOC

20. Displaying the Label Information Area
21. Displaying and Punching Members
22. Displaying the Library Directories
23. Printing the Hard-Copy File (PRINTLOG)
24. Printing Job Information (LISTLOG)
25. Executing Cataloged Programs
26. Gathering Error Information
27. Shutting down the System

1. Preparing the Printer for Buffer Load

If your computing system includes printers with Forms-Control and Universal-Character-Set buffers (FCBs and UCBs), these buffers are loaded automatically during IPL. An FCB is loaded with the standard FCB image, and a UCB is loaded with the standard buffer image.

Each printer must be prepared as follows before IPL:

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

Notes:

1. *For the 1403U printer, only the UCB is loaded.*
2. *The UCBs for the 4245 and the 4248 printers are loaded automatically by microcode.*
3. *If your system includes an IBM 3800 Printing Subsystem, the printer is initialized automatically during IPL. The printer must be readied.*

2. Printer Train Cleaning

To preserve the quality of the printed output on your system, it is necessary to clean the printer train from time to time. The IBM 3211 Printer has an integrated train cleaning mechanism, but other printers require the following cleaning procedure.

First, ensure that the printer to be cleaned is not in use as a spooling device for VSE/POWER.

Then, start the procedure by entering

```
// ASSIGN SYS000,cuu

// EXEC CLEANER
```

(For cuu, enter the unit address of the printer you want to clean.)

The CLEANER procedure prompts you for the required activities, as follows:

System prompt:

8CO1A STOP THE PRINTER (SYS000) AND PRESS END-KEY

Operator action: Press STOP on the printer and press ENTER at the console. Ignore any INTERVENTION REQUIRED condition until you have replaced the paper.

System prompt:

8C021 IN ORDER TO CLEAN THE PRINTER TRAIN REMOVE THE RIBBON AND REPLACE THE PAPER ON THE PRINTER WITH IBM CLEANING PAPER (PART NO: 451529)

Operator action: Remove the ribbon, replace the paper and ready the printer.

System action: The system prints a pattern which brings all the characters of the train in contact with the cleaning paper several times.

For the IBM 3203 Model 4 printer, the UCB image which was in use before the cleaning procedure is restored automatically. For other printers, the procedure issues the message:

System prompt:

8C06A SPECIFY NAME OF TRAIN IMAGE OR PRESS END TO GET IPL UCB LOADED

Operator action: Enter the UCB phase name you require. If you just press END/ENTER, the UCB defined at IPL is loaded.

System prompt:

8C01A STOP THE PRINTER (SYS000) AND PRESS END-KEY.

Operator action: Press STOP on the printer and ENTER at the console.

System prompt:

8C031 THE PRINTER TRAIN HAS BEEN CLEANED

REPLACE THE RIBBON AND THE ORIGINAL PAPER

Operator action: Insert the ribbon, thread the desired paper, and ready the printer.

The cleaning procedure is now complete, and the printer can be assigned for normal output.

3. Automated System Initialization (ASI)

Automated system initialization (ASI) drastically reduces the work otherwise required to bring up the system and to start subsystems (e.g. VSE/POWER) in the partitions. In fact, the only thing you have to do when ASI is installed in your system (and the TOD-clock is running) is to activate the initial microcode load and enter the address of the IPL device.

All the information necessary to bring up the system is cataloged in procedures which are executed automatically when you start IPL. You can, of course, interrupt the IPL process, make temporary changes to a procedure, or do IPL interactively.

Before ASI can be installed in your system, you must perform an interactive IPL. Then the procedures required for ASI have to be cataloged in the system sublibrary IJSYSRS.SYSLIB. Once the ASI procedure set has been installed, each subsequent IPL is done by ASI. The procedure names taken from the sublibrary are listed on SYSLOG, together with ASI IPL information messages.

ASI comprises two processing steps:

1. An automated IPL (ASI IPL).

2. An automated partition bring-up performed by job control (ASI JCL).

To execute these steps, ASI requires one procedure for IPL and one job control procedure for each partition to be started automatically. The IPL procedure contains the SYSLOG device address, the supervisor name, the paging and list options, and the IPL commands, except for the SET command.

The SET command initializes the system time and date, and is normally needed only during the first IPL after power-on. However, stop the ASI IPL procedure before processing the IPL commands which open system files. These are the DEF, DLA and DLF commands. An incorrect date could cause system files to be overwritten. If the system time and date are not already set, you must set them at this point by entering a SET command. If the time and date are already set, you can simply press END or ENTER at the console to let the ASI procedure continue. The SET command with only the **ZONE=** operand **can** be included in an ASI IPL procedure.

After ASI IPL completion, job control is automatically invoked into the background partition and calls the ASI background procedure. This procedure must contain all necessary job control statements and commands to initialize the BG partition and the system as a whole. The ASI foreground procedures are called after the foreground partitions have been started. A foreground partition can be started during execution of the ASI BG procedure or by VSE/POWER or with an AR START command. These procedures must contain job control statements to initialize the respective partition.

To perform the ASI procedure, proceed as follows:

1. Perform the power-on and load-microcode procedures as described in the appropriate hardware manuals.

2. Wait for message 0J01I which lists the names of the ASI procedures to be used for IPL and job control, as well as the name of the supervisor to be loaded.

ASI now executes the procedures, notifying you of any invalid commands contained in the ASI IPL procedure. The system writes the invalid command to SYSLOG. If SYSLOG is a display console, the invalid command is followed by message 0D07D ENTER RESPONSE. You can now restart ASI and correct the erroneous command, using the ASI Restart facility described below.

If the time-of-day clock is inoperative or not set, ASI will also prompt you for the SET command.

ASI Restart and Interactive IPL

You can interrupt ASI if you want to:

- Use non-default ASI IPL or JCL procedures. The default procedures are those listed in message 0J01I;

- Make temporary changes or corrections to the IPL commands contained in the ASI IPL procedure;

- Bypass ASI and perform an interactive IPL.

In these cases, proceed as follows:

1. Press the INTERRUPT key before message

   ```
   0J10I  IPL  RESTART  POINT  BYPASSED
   ```

   is displayed on SYSLOG. If the message **is** listed, start IPL again.

2. When the wait indicator comes on, press REQUEST or ENTER at SYSLOG. You may switch to a SYSLOG device different from that defined in the IPL procedure. Do this by pressing REQUEST or ENTER on the device you want to use as SYSLOG.

3. Wait for message

   0I03D ENTER SUPERVISOR PARAMETERS OR ASI PARAMETERS

   You can now use alternative ASI procedures, do an interactive IPL, or alter IPL commands in the existing ASI IPL procedure. Choose action a, b, or c from the following list:

   a. If you want to use alternative ASI procedures, enter the names of the desired ASI IPL and JCL procedures as follows:

```
IPL=IPL-procedurename,JCL=JCL-procedurename
```

Message 0J01I is reissued with the new ASI procedure names and the procedures are executed automatically.

b. If you want to update the IPL commands contained in the ASI IPL procedure or enter new commands, reply as follows:

```
IPL=IPL-procname,JCL=JCL-procname,STOP=(stoplist),TYPE=type
```

"JCL-procedurename" can be taken from message 0J01I. "Stoplist" specifies a single IPL command or a list of up to four different IPL commands, separated by commas and enclosed in parentheses, for example:

```
STOP=ADD
```

```
STOP=(ADD,DEF,DPD,SVA)
```

"Type" specifies how I/O devices are to be added during IPL. TYPE=SENSE causes IPL to add the devices which support device sensing automatically. TYPE=NORMAL means that IPL only checks your ADD statements for correctness.

c. If you want to perform an interactive IPL, now enter the name of the supervisor to be loaded (default is $$A$SUP1), optionally followed by the paging, list, VPOOL, VIO and VSIZE options, and proceed with step 1 of Procedure "8. Interactive IPL at the Console" on page 63.

When, during ASI IPL processing, an IPL command is encountered which is part of the stop list, ASI stops before processing this command. The command is saved and listed on SYSLOG, followed by the message:

```
0J05D  ASI STOP.  ENTER  COMMANDS,
       HIT END/ENTER TO CONTINUE
```

You can now enter one or more updated or new IPL commands from the console. To proceed with ASI, press END or ENTER. The saved command is now resubmitted to ASI for processing. It is recommended to submit the SET command in this way.

ASI then processes the remaining IPL commands in the procedure, stopping each time a command is encountered which is contained in the stop list. If the IPL procedure contains several commands of the same type, ASI will stop only before processing the **first** of these commands. For example, if you enter STOP=DEL, and the procedure contains several DEL commands, ASI stops before the first DEL command. When you press END or ENTER, the rest of the procedure is executed without stopping. The new or updated commands will change the procedure only temporarily, that is, they are valid only until the next IPL.

Assume your ASI IPL procedure looks as follows:

```
009,$$A$SUPX,N,NOLOG
ADD 009,3270
     .
     .
ADD 480,3370
DEF SYSREC=150
DEF SYSCAT=150
DPD ...
SVA ...
```

Assume you have to change the device type for 480, add device 481, change the assignment for SYSCAT, and redefine the page data set. Proceed as follows:

1. Perform the power-on and load-microcode procedures.

2. Wait for message 0J01I (it tells you the ASI procedure names).

3. Restart ASI:

   a. Press the INTERRUPT key

   b. Press REQUEST/ENTER on the console

   c. Wait for message 0I03A.

4. Reply: IPL=..., JCL=..., STOP=(DEF,DPD). (Take procedure names from message 0J01I.)

5. Before processing DEF SYSREC, ASI will stop, listing the DEF command followed by message 0J05A.

6. Now enter from SYSLOG:

   DEL 480

   ADD 480,3380

   ADD 481,3380

   Pressing ENTER results in processing DEF SYSREC.

7. ASI stops again before processing the DPD command. Now enter:

   DEF SYSCAT=151 (which overrides the DEF SYSCAT=150 specified in the ASI IPL procedure)

   DPD ... (which overrides the DPD command in the ASI IPL procedure)

The IPL program issues an error message, but continues to process the remaining commands in the procedure.

Note: The error message can be ignored. Pressing ENTER normally results in processing the command at which the STOP took effect (in this case, the DPD command). However, because a DPD command has already been accepted from the console, and only one DPD command may be submitted during each IPL, the DPD command from the ASI procedure is rejected. This causes the error message.

Starting the System Without ASI

Sometimes you may have to start the system without the help of the Automated System Initialization procedures.

The actions you must take are described in Procedures 4 to 10. They are:

1. Starting IPL;
2. Entering IPL commands;
3. Creating system files.

How you do the first two steps depends on what type of console you are using, and from what I/O device you want to enter the commands. The third and fourth steps may or may not be required (ask your system programmer). They are carried out at the console.

Choose the procedures you need from this list:

1. Starting IPL (choose a, b, c or d):

   a. At the 3210 or 3215 Printer Keyboard: "4. Starting IPL with the IBM 3210 or 3215 Printer-Keyboard" on page 58.

   b. At the 3277/3278/3279 Display Console: "5. Starting IPL with the IBM 3277/3278/3279" on page 59.

   c. At the Model 158 Display Console: "6. Starting IPL with the IBM Model 158 Display Console" on page 60.

   d. At the 303x Display Console: "7. Starting IPL with the IBM Model 3031 Display Console" on page 61.

2. Entering IPL commands manually: "8. Interactive IPL at the Console" on page 63.

3. Creating system files (do a or b or both, if needed):

   a. Creating the System Recorder File: "9. Creating the System Recorder File" on page 66.

   b. Creating the Hard-Copy File: "10. Creating the Hard-Copy File" on page 67.

4. Starting IPL with the IBM 3210 or 3215 Printer-Keyboard

1. Switch the power on as described in the appropriate hardware manual.
2. Ready the devices containing SYSRES, the page data set, the label information area and, if required, the lock file.
3. Set the load-unit switches on the system control panel to the physical device address of the disk drive that holds the SYSRES pack.
4. Press LOAD on the system control panel.
5. The WAIT indicator comes on.

Turn to Procedure "8. Interactive IPL at the Console" on page 63.

5. Starting IPL with the IBM 3277/3278/3279

1. Perform the power-on and load-microprogram procedures as described in the appropriate hardware manuals. On the 3278 and 3279, the program load display appears on the screen. If this is not the first IPL since power-on, press the MODE SELECT key. When the mode select display appears, type in L and press ENTER. The program load display appears on the screen, and you can proceed as follows:

2. Ready the device containing SYSRES and the device that contains the page data set. If the devices have removable disks, you must first mount the correct packs.

3. **For the 3277**:

   a. set the load-unit switches on the system control panel to the physical device address of the disk drive that holds the SYSRES pack, and

   b. press LOAD on the system control panel.

   On the 3278 and 3279: type in

   a. the address of the SYSRES device,

   b. the character C in order to clear storage,

   c. other required specifications in the appropriate fields of the program load display, and

   d. press ENTER.

4. **(All types)** The WAIT indicator comes on.

Turn to Procedure "8. Interactive IPL at the Console" on page 63.

6. Starting IPL with the IBM Model 158 Display Console

1. Insert the operations console file in the console's diskette reader.
2. Press POWER ON.
3. Press MODE SEL. This causes the manual frame to be displayed.
4. Type in L. This causes L-LOAD UA to be activated.
5. Type in the load unit address using the keyboard.
6. Press ENTER.
7. Type in O, 7, and X, causing an IPL from the selected unit.
8. Press ENTER.
9. The WAIT indicator comes on.

Turn to Procedure "8. Interactive IPL at the Console" on page 63.

7. Starting IPL with the IBM Model 3031 Display Console

1. Press the Power-On switch.
2. Turn the security keys on the sides of both display units to the enable position (key face horizontal).
3. Set the control panel switches as follows:
   - I/O Interface Console A and B to Enable.
   - Diagnostic on IMPL to Disable for both A and B.
   - Operator Console on IMPL to A.
   - Power Select to System.
4. Press the IPC Reset button on the control panel.
5. Press the Power On button on the control panel. Wait for the display of the C1 (Configuration) frame.
6. On Console A:
   - Type in A1X.
   - Press the SEL FRAME key and type in PR
7. On Console B:
   - Type in A2X.
   - Press the SEL FRAME key and type in OP. (Ready the devices containing SYSRES and the page data set and make sure that the correct volumes are mounted.)
   - Type in L and enter the IPL unit address (three digits).
   - Type in O7X (or O4X if the CLEAR function is not desired; for example, for an IPL of a stand-alone dump).
8. On Console A, press ENTER.

9. The WAIT indicator comes on.

Note: *The command:*

```
ADD cuu,7443
```

has to be given during interactive IPL (in step 3 of Procedure "8. Interactive IPL at the Console" on page 63) to define the service record file associated with the service support console.

Turn to Procedure "8. Interactive IPL at the Console" on page 63.

8. Interactive IPL at the Console

Interrupt the wait state by pressing REQUEST or ENTER at the console. The system responds with the following information message identifying the SYSRES file and CPU:

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

Then the following message is issued:

```
0I03A  ENTER  SUPERVISOR  PARAMETERS

[OR ASI  PARAMETERS]
```

If you wish to use the default supervisor ($$A$SUP1) with the paging and list option on, and the system default virtual address area (370 mode only), press ENTER; otherwise, enter the name of the required supervisor, the size of the virtual address area (VSIZE = nK - 370-mode only), the size of the virtual I/O work area (VIO = [K|M]) and the size of the V-pool (VPOOL = n[K|M]), the paging and the list option, and press ENTER.

If you want to proceed with ASI, enter the name of the desired ASI IPL (and JCL) procedure, as described in step 3a of the ASI Restart procedure (Procedure 3).

To do an interactive IPL, proceed with the following steps:

1. When the WAIT indicator comes on again, press REQUEST on the printer-keyboard, or ENTER on the display console.

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

   a. 0I30I DATE = date, CLOCK = time, ZONE = difference

      THE DATE VALUE FORMAT IS MM/DD/YY

      0I10D GIVE IPL CONTROL COMMAND

   b. 0I31I DATE REQUIRED, CLOCK REQUIRED,

      ZONE = difference

      THE DATE VALUE FORMAT IS MM/DD/YY

      0I10D GIVE IPL CONTROL COMMAND

c. 0I32I TOD CLOCK INOPERATIVE; NO TOD SUPPORT,

   NO IT SUPPORT

   0I31I DATE REQUIRED, CLOCK REQUIRED,

   ZONE = difference

   THE DATE VALUE FORMAT IS MM/DD/YY

   0I10D GIVE IPL CONTROL COMMAND

2. Depending on the messages that were issued on SYSLOG (see step 6), take the following action:

   a. If all values are satisfactory, enter the next IPL command.

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

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

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

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

3. Enter ADD commands (and if necessary, DEL commands).

4. Enter the DLF command, if required, to define the system lock file.

5. Enter the DEF command, to indicate on which physical devices the disk packs containing SYSREC and SYSCAT are mounted.

6. Enter the DPD command, to define the page data set.

7. Enter the DLA command to define an additional label information area.

8. Enter the SYS command to specify

   - System action when printers are not READY for automatic buffer load;
   - The number of supervisor I/O buffers;
   - The number of channel queue entries;
   - Whether DASD file protection is to be active;

- Whether access control is to be active;
- How many sublibraries may be defined in the system;
- Whether job accounting should be active;
- The size of the real storage area for monitoring functions.

9. Enter the SVA command to allocate space within the shared virtual area into which user phases can later be loaded. Even if no such space is required, the mandatory SVA command must be entered (without operands) to signal the end of the IPL commands.

10. The system then issues the messages

    0I20I IPL COMPLETE ...

    1I00D READY FOR COMMUNICATIONS

    to the console.

You can now start the partitions of the system.
How this is done is explained in "15. Starting the Background Partition" on page 77 and "16. Starting a Foreground Partition" on page 78.

9. Creating the System Recorder File

The system requires a disk extent on which to record statistical information on machine errors and environmental information. 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 only (**not** after each IPL). If, however, the SYSREC file is damaged, you must re-IPL and recreate the system recorder file.

If your system is running on a 3031, the SYSREC file must be listed (by running the Environmental Recording Editing and Printing (EREP) program) and recreated each time a hardware (microcode) change is installed which affects the frames on the 3031's Service Record File (SRF).

The following commands and job control statements create the system recorder file. The commands and statements outside the box are included to show the proper placement of the statements and commands that create the recorder file.

```
0I10A  GIVE IPL CONTROL COMMAND
       ADD cuu,7443     SRF device of 3031,
                        usually on address 01D
                        (always use address asso-
                        ciated with Service Support
                        Console)

      SET
      DEF   SYSREC=cuu
      DPD
      SVA

0I20I IPL COMPLETE...
1I00D READY FOR COMMUNICATIONS
```

```
 SET RF=CREATE
//  OPTION STDLABEL
//  DLBL IJSYSRC,'DOS.SYSTEM.RMSR.FILE'
//  EXTENT SYSREC,nnnnnn,,,nnnnn,nnnnn
```

```
•
•    (other standard label information)
•
//   JOB FIRST
```

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

Turn to Procedure "10. Creating the Hard-Copy File" on page 67.

10. Creating the Hard-Copy File

If a display console is operated without a console printer, a disk extent is required which is used for recording all information on the screen, both from the system and from the operator. This disk extent is called the hard-copy file and resides on the device assigned to SYSREC. The hard-copy file can be used to obtain hard (printed) copies of the file whenever required.

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

The following sequence of commands and job control statements is an example of how to create the hard-copy file. The commands and statements outside the box are included to show the proper placement of the statements and commands that create the hard-copy file.

```
0I10A  GIVE IPL CONTROL COMMAND
       ADD
       SET
       DEF    SYSREC=cuu
       DPD
       SVA

0I20I  IPL COMPLETE...
1I00A  READY FOR COMMUNICATIONS
```

```
SET HC=CREATE
//  OPTION STDLABEL
//  DLBL IJSYSCN,'DOS/VS.HARD—COPY.FILE'
//  EXTENT SYSREC,nnnnnn,,,nnnnn,nnnnn
```

```
•
• (other standard label information)
•
     // JOB FIRST
```

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

Issue the ROD command before system shut-down time, to record all messages, including the latest.

Printing the hard-copy file is described in Procedure 23.

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

11. Loading Print Buffers

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

There are two ways of loading these buffers: (1) by having the system execute the SYSBUFLD program and (2) by using attention commands. In addition, the FCB can be loaded dynamically under the control of a problem program that is being executed. In this case, you merely have to service the particular printer in response to appropriate system messages. Note that the UCS job control command can also be used for loading the UCB of a 1403 UCS printer.

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

For information on how to load the FCB/UCB under VSE/POWER, see the VSE/POWER manuals listed in the Bibliography.

Loading the FCB by Using the SYSBUFLD Program

1. Enter the following control statements on SYSIN:

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

   The operands of the FCB statement are the same as the corresponding operands of the LFCB command. For a discussion of these operands, see *VSE/Advanced Functions System Control Statements.*

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

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

3. If the subsequent print job requires a different setting of the carriage clutch, change the setting now (does not apply to a PRT1 printer).

4. Stop the printer.

5. Press END/ENTER.

6. To align the new forms, place the upper margin of the first new form exactly over that fold of the currently used forms which has just passed the print train.

7. Ready the printer.

Loading the FCB by Using the LFCB Command

1. Enter the LFCB attention routine command (for details about this command, see *VSE/Advanced Functions System Control Statements*. The format of the command is:

   ```
   LFCB cuu,phasename[,FORMS=xxxx][,NULMSG]
   ```

   Note: If FORMS=xxxx is not specified, the system does not return a message and no further operator action is required.

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

Respond to message 1B13A as follows:

a. Stop the printer (only if forms must be changed).

b. Change the setting of the carriage clutch (if necessary).

c. Press END/ENTER (omit steps d and e if forms need not be changed).

d. Place the new forms on the printer and align line 1 of the first new form with the print line.

e. Ready the printer.

Respond to message 1B14A as follows:

a. Stop the printer.

b. Press END/ENTER.

c. To align the new forms, place the upper margin of the first new form exactly over that fold of the currently used forms which has just passed the print train.

d. Ready the printer.

Loading the UCB by Using the SYSBUFLD Program

1. Mount the new print train on the printer and ready the device.

2. Enter the following control statements:

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

   The operands of the UCB statement have the same purpose as the corresponding operands of the LUCB command. For details refer to *VSE/Advanced Functions System Control Statements*.

Loading the UCB by Using the LUCB Command

1. Enter the LUCB command (for details about this command, see *VSE/Advanced Functions System Control Statements*).

   The format of the command is:

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

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

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

2. Stop the printer.

3. Press END/ENTER.

4. Mount the new print train (or chain) and ready the printer again.

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

Loading the 1403U Buffer by Using the UCS Job Control Command

1. Mount the new print chain on the 1403U.

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

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

   The operands of the UCS command have the same purpose as the corresponding operands (BLOCK=NOCHK) of the LUCB command. For details refer to *VSE/Advanced Functions System Control Statements*.

Print Band Handling on 4248 Printers

If you have an IBM 4248 printer in native mode, you may have to change the print band, or check that the correct band is mounted. (A 4248 is in native mode if it was ADDed at IPL with the device type 4248 and is switched to 4248 operation mode. It is also possible to ADD an IBM 4248 with the device type PRT1 and set the switch to 3211 operation mode.)

The attention routine (AR) command BANDID gives you the information you need.

To check which band is mounted, enter the command:

```
BANDID cuu
```

where **cuu** is the physical unit address of the 4248 printer (for example, 00E). You then get a console message giving you the band identifier of the print band at present on the printer. If a different band is required for the next job, change the band and check that you have mounted the right one by entering the BANDID command again.

Loading the Buffers in a 3800 Printing Subsystem

Loading of the buffers in an IBM 3800 Printing Subsystem is controlled by the SETPRT command or statement. This is normally provided in the form of a job control statement by the programmer. For further details, see *VSE/Advanced Functions System Control Statements.*

12. Setting Defaults for an IBM 3800 Printing Subsystem

If your computing system includes an IBM 3800 Printing Subsystem, you can set and/or reset the 3800 system default parameters by using the SETDF operator command. Issuing the SETDF command does not change the settings of the 3800. Instead, the parameters are saved such that when the user specifies DFLT = Y or keyword = * in a SETPRT statement, the SETPRT routine sets the 3800 with the predefined defaults.

The following examples illustrate uses of the SETDF command:

Example 1: To set selected default parameters for a 3800 and reset the others, enter the command:

SETDF 018,BURST = Y,FLASH = AC01,FORMS = 8X11,RESET

This command sets the 3800 printer whose address is 018 with the default of bursting = yes, forms overlay name = AC01, and forms name = 8X11. It resets the values for CHARS, FCB, and MODIFY. If all 3800 printers at your location are to be set with these defaults, specify 3800 instead of a single printer address.

Example 2: To set or reset selected defaults, and list the defaults, enter the command:

```
SETDF 3800,BURST=,FCB=AC02,FORMS=,MODIFY=AC03,LIST
```

This command resets the system default values for the BURST and FORMS parameters. The FCB default is specified as AC02 and the default copy modification is AC03. The LIST parameter then causes a display of the defaults on the operator console.

For detailed information, see the description of the SETDF command in *VSE/Advanced Functions System Control Statements.*

13. Assigning I/O Devices

After you have started the system, you have to make the necessary I/O device assignments. 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 /& or // 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,00E
```

or

```
ASSGN SYSLST,00E,TEMP
```

Permanent assignment:

```
ASSGN SYSLST,00E
```

or

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

The device address (00E in the example) may be replaced by a generic name or device type, for example PRINTER or 1403. You will find more information on how to use ASSGN in *VSE/Advanced Functions System Control Statements.*

If you are not certain about the current I/O device assignments at your location, issue the LISTIO **Command**, which prints all current assignments on SYSLOG. The // LISTIO job control **statement** prints all current assignments on SYSLST.

The output of a // LISTIO F1 statement, produced on SYSLST, is shown in Figure 9.

```
                              *** FOREGROUND 1***

I/O UNIT ASSGMNT CHNL UNIT MODE

 SYSRDR  PER        0   0C
 SYSIPT  PER        0   0C
 SYSPCH             * UA *
 SYSLST  PER        0   0E
 SYSLOG  PER        0   1F
 SYSLNK  PER        1   31
 SYSRES  PER        1   30
 SYSUSE             * UA *
 SYSREC  PER        1   30
 SYSDMP             * UA *
 SYSCAT             * UA *
 SYSTEM FILES
 SYS000  PER        2   90   C0
 SYS001  PER        2   80
 SYS002  TEM        2   80
 SYS002  PER        2   81
 SYS003  PER        1   30
 SYS004  PER        1   30
 SYS005  TEM        2   30
 SYS005  PER        * UA *
 SYS006             * UA *
 SYS007  TEM        2   80   D0
 SYS007  PER        2   60
 SYS007  TEM ALT    2   81   D0
 SYS008             * UA *
 UP TO
 SYS099             * UA *
```

Figure 9. Sample Output of the LISTIO Statement for F1

Logical units are listed grouped by partition (BG, Fn). The system logical units are listed first. Note that the logical units SYSUSE and SYSDMP may be shown as assigned, although you have entered no ASSGN command for them. The system assigns these units automatically when it needs them. The line SYSTEM FILES refers to the page data set and the lock file. This line contains no further information.

The programmer logical units follow in ascending order starting with SYS000. The number of programmer logical units shown corresponds to the value specified in the NPGR command. In the example in Figure 9, NPGR F1 = 100 has been specified. If a series of programmer logical units is unassigned, this is shown by the line UP TO inserted between the first and the last of the series (SYS008 and SYS099 in the example).

The information for each logical unit consists of the type of assignment, the physical address of the device used, and, for assignments to tape devices, the mode specified.

Column	Contents
I/O UNIT	The logical unit name to which the information in this line refers.
ASSGMT	The type of assignment. This may be PER for a permanent assignment, or TEM for temporary. In addition, if the unit is assigned to a tape device, the entry ALT may appear. This means that the device at the address shown is an alternate assignment. The entries PER ALT and TEM ALT are both possible. If a permanent and a temporary assignment exist for a logical unit, each is shown in a line by itself. A unit assigned with an alternate tape device has an additional line showing the alternate assignment. See SYS007 in the example in Figure 9 on page 74. If the logical unit is unassigned (* UA * in the CHNL UNIT column), this column is left blank. The exception to this rule is when a logical unit has a temporary but no permanent assignment. In this case, PER appears on the same line as * UA *, and TEM with the device address on the line above. See SYS005 in the example in Figure 9 on page 74.
CHNL UNIT	The channel and unit address to which the logical unit is assigned. If the unit is unassigned, this column contains * UA *.
MODE	For assignments to tape only, the specified tape mode.

You can use the RESET command to reset temporary I/O device assignments. This command resets temporary device assignments to the last permanent assignment, if any (made by an ASSGN command without the TEMP option). If no 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.

You can use the VOLUME command to obtain a short summary of the volumes mounted on disk devices. The output of the command includes the volume identifier of the mounted pack, together with an indication of whether the volume is in use, whether it is shared by another system, and whether it is reserved for VSAM space management service. For details, see the description of the VOLUME command in *VSE/Advanced Functions System Control Statements*.

14. Allocating Storage

Normally, storage allocation is done by the ASI JCL procedures during IPL. However, the values may have to be changed while the system is running. If you have started the system with an interactive IPL, you also have to allocate the storage to the partitions.

There are two commands - ALLOC and SIZE - that enable you to allocate or reallocate storage.

ALLOC n or ALLOC S allocates virtual storage to a partition;

ALLOC R allocates processor storage to a partition (for real-mode execution);

SIZE changes the amount of storage in a partition reserved for program execution.

There are a number of rules and restrictions that must be observed when using these 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.

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 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 or SIZE commands and to keep a record of the new system status.

For a description of the MAP command and examples of the MAP output, see *VSE/Advanced Functions System Control Statements.*

15. Starting the Background Partition

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

Normally, partition start-up is covered by the ASI procedures for your system. After an interactive IPL, however, you must start the partitions interactively. When it has processed the last IPL command, the system displays the message 0I20I IPL COMPLETE ..., followed by the partition indicator BG, to signal that the background partition is ready for processing. You must then prepare the I/O devices that are needed by the first (or only) job and ready the device that is to be used as SYSRDR.

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

```
BG-000 1C10D PLEASE ASSIGN SYSRDR
```

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

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

16. Starting a Foreground Partition

Assume that the background partition is processing a job stream, and you now want to start a foreground partition. Foreground partitions must be allocated after IPL with ALLOC commands before they can be started. This is usually done by the ASI JCL procedures.

To start a foreground partition from the console, for example F1, enter the attention command (on a printer-keyboard console, press the REQUEST key first):

```
START F1
```

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

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

You can obtain a list of I/O device assignments by entering the job control command or statement LISTIO. If you enter // LISTIO ALL you will get, on SYSLOG, a list of all current assignments for all partitions. To obtain BG and F1 assignments only, you must enter LISTIO BG, wait until the list of I/O assignments has been produced and the partition identifier F1 appears again on SYSLOG. You then enter LISTIO F1. When the list of assignments for F1 has been produced, the partition identifier F1 will again appear on SYSLOG. Make sure that the assignments are correct; if necessary, make new assignments for F1, using the ASSGN command.

You may now enter job control statements or commands from SYSLOG. If you do not want to do this, ready SYSRDR and press END or ENTER. The job control 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.

The steps described above can, of course, be integrated in the system's ASI procedures. The JCL procedure for the BG partition must contain the START Fn commands, and the job control statements to be used for bringing up the foreground partition must be cataloged in the sublibrary IJSYSRS.SYSLIB under a member name with the form $nxxxxx, where n is the number of the foreground partition in question, and xxxxx is the same as the third to last characters of the JCL procedure name given in the ASI start command.

17. 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:

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.

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 either case, you continue processing by pressing END or ENTER.

Sometimes 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 device assignments and active library definitions (LIBDEFs) 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.

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.

18. Using Cataloged Procedures

VSE/Advanced Functions allows you to catalog procedures, that is, sets of job control and linkage editor control statements, with or without SYSIPT data, to a sublibrary. You can access this sublibrary whenever you need a procedure to complete or set up a job stream.

This section describes how to

- retrieve a procedure from a sublibrary;
- modify control statements of a cataloged procedure for the duration of one job;
- catalog a procedure into a sublibrary.

Calling JCL Procedures

To use a procedure, simply enter the following statements at your console:

```
// JOB jobname
// LIBDEF PROC,SEARCH=library.sublibrary
// EXEC PROC=procedurename
/&
```

For "jobname", type in any name you like, up to eight characters long. For "library.sublibrary", type in the name of the sublibrary in which the procedure is stored. For "procedurename", type in the name of the procedure you want to use.

No matter how long or complex the procedure is, this is all you have to do to run it.

Let us take a simple example of a procedure, and assume that a LIBDEF statement for the appropriate sublibrary has been entered. Suppose you often list the volume table of contents (VTOC) of the SYSRES disk on the printer at address 00E. "19. Displaying the VTOC" on page 83 shows you what to do. You have to assign SYS004 to the disk in question, and SYS005 to the printer to be used. Then you must call the utility program LVTOC. That is, each time you want a listing, you must enter:

```
ASSGN SYS004,SYSRES,TEMP
ASSGN SYS005,00E,TEMP
EXEC LVTOC
```

To speed up operation, you might have the following procedure stored with the name LVTOCP (for List-VTOC-Procedure):

```
// PROC
// ASSGN SYS004,SYSRES,TEMP
// ASSGN SYS005,00E,TEMP
// EXEC LVTOC
/+
```

To list the VTOC of the SYSRES disk, you would now have to enter:

```
// JOB LV
// EXEC PROC=LVTOCP
/&
```

Or, at the console, simply:

```
EXEC PROC=LVTOCP
```

Even in this simple example, you have saved quite a lot of typing. If you have procedures for all frequently-run programs, the saving is considerable.

Modifying JCL Procedures

The example above lists only the VTOC of the SYSRES disk. However, there is no need to write a separate procedure for each disk address on your system. The procedure LVTOCP will do for all disk addresses, if it is written as follows:

```
// PROC DISK=SYSRES
// ASSGN SYS004,&DISK
// ASSGN SYS005,00E
// EXEC LVTOC
/+
```

The logical unit SYS004 is not assigned to a specific physical unit. The characters "&DISK" are what is called a **"symbolic parameter"**. When the operating system reads the procedure, it substitutes a physical unit address. Which one? Look at the **PROC statement**. Here, DISK=SYSRES is specified. So the procedure is executed as if it contained the statement:

```
// ASSGN SYS004,SYSRES
```

If you call this procedure with the statement:

```
// EXEC PROC=LVTOCP
```

you will get a listing of the VTOC of the SYSRES disk.

If, however, you want a VTOC listing of the disk on unit address 151, you can still use the LVTOCP procedure. You simply have to use the EXEC statement:

```
// EXEC PROC=LVTOCP,DISK=151
```

Now the procedure will be executed as if it had the statement:

```
// ASSGN SYS004,151
```

This is called **"passing a parameter** to a procedure". If you do not pass a parameter, the value specified in the PROC statement of the procedure is used. This is called the **"default value"**.

Symbolic parameters can be used anywhere in the operands of job control statements. Many of the procedures you use may have symbolic parameters, and you can pass the values you need each time you use them.

The **internal documentation** at your location should tell you what parameters are needed for what procedures, and what their default values are.

Cataloging JCL Procedures

To catalog a procedure into a sublibrary, use the librarian program with a CATALOG control statement. Enter the statements at the console:

```
// JOB jobname
// EXEC  LIBR
ACCESS SUBLIB = lib.sublib
CATALOG membername.PROC

// PROC       |
// ...        |   job control
// ...        |   statements to
// ...        |   be cataloged

/+
END
/&
```

The ACCESS command specifies the name of the library/sublibrary into which the procedure is to be cataloged, and the membername operand of the CATALOG command specifies the name of the procedure. The end of the procedure is indicated by the /+ statement. You may catalog more than one procedure in one job. In this case, the next ACCESS and CATALOG commands follow the /+ statement.

A number of restrictions apply to cataloging procedures. For details, refer to *VSE/Advanced Functions System Management Guide.*

19. Displaying the VTOC

The operator action for a number of system 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.

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

1. Using SYSRDR.

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

   ```
   // JOB NAME
   // ASSGN SYS004,cuu
   // ASSGN SYS005,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.

2. Using the console (SYSLOG).

   To display the VTOC from the printer-keyboard or display console, enter the following statements:

   ```
   // ASSGN SYS004,cuu
   // ASSGN SYS005,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, unless a message response (CANCELV, for example) is available for causing both the program to be cancelled and the particular VTOC to be printed on SYSLST.

20. Displaying the Label Information Area

The "operator action" for a number of system messages includes listing the label information area. This listing helps your programmer to determine and correct the error that caused the message.

To display the contents of the label information area, use the LSERV program, which prints the contents of the label information area on the device assigned to SYSLST. You can do this in one of two ways:

1. Using SYSRDR

   The job control statements required to execute the LSERV program from SYSRDR are:

   ```
   // JOB NAME
   // EXEC LSERV
   /*
   /&
   ```

2. Using the system console (SYSLOG).

   To execute the LSERV program from the console, enter

   ```
   // EXEC LSERV
   ```

If you want to display the label information area in response to an error message, cancel the program that caused the error before executing LSERV.

21. Displaying and Punching Members

The librarian allows you to display and/or punch the contents of all or selected members of sublibraries. For example, your programmer may request you to have a phase punched and then re-linkedited with one or more other phases to form a new program.

Two librarian commands are available to print and punch the contents of members:

LIST and PUNCH

Output generated by the PUNCH command is written to the device assigned to SYSPCH. By default, output generated by the LIST command is written to:

- SYSLOG, if the command is entered from SYSLOG;
- SYSLST, if the command is entered from any other source.

You can override the default selection by using the UNIT operand of the LIST command.

Note that the LIST and PUNCH commands must be preceded by an ACCESS command, to specify from which sublibrary the specified members are to be taken.

When punching members of type PHASE, the Librarian program prepares the members for recataloging. It generates a CATALOG command at the beginning and an EOD (end-of-data) command at the end of the output. If several phases are punched with one PUNCH command, the CATALOG command is in front of the first phase, and the EOD after the last one.

You can suppress the CATALOG and EOD commands by specifying the operand FORMAT=NOHEADER in the PUNCH command.

The examples in Figure 10 on page 86 contain the control commands necessary to display and punch members. When preparing the jobs, be sure to replace the words lib, sublib, member name and member type with the actual names required.

Function	Librarian Commands and JCS Required

Punch one or more phases	// JOB jobname
	// EXEC LIBR
	ACCESS SUBLIB = lib.sublib
	PUNCH membername.PHASE ad.,...bd.
	/*
	/&

Punch all phases in a sublibrary	// JOB jobname
	// EXEC LIBR
	ACCESS SUBLIB = lib.sublib
	PUNCH *.PHASE
	/*
	/&

Display one or more members of any type	// JOB jobname
	// EXEC LIBR
	ACCESS SUBLIB = lib.sublib
	LIST membername.membertype ad.,...bd.
	/*
	/&

Display all members in a sublibrary	// JOB jobname
	// EXEC LIBR
	ACCESS SUBLIB = lib.sublib
	LIST *.*
	/*
	/&

Figure 10. Librarian Print and Punch Examples

22. 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 was not found, you can display the directory of the appropriate sublibrary to determine whether the procedure is missing from the sublibrary or whether you have merely misspelled the name of the procedure.

The directories are displayed by the librarian command LISTDIR. The output is written on SYSLOG, if the command is entered from there, otherwise on SYSLST. You can direct the output to the non-default device explicitly, using the UNIT operand of the LISTD command.

The output depends on the values specified in the operands of the LISTDIR command. It is possible to display:

- the directory contents for all sublibraries in a specified library;
- the directory contents for one or more specified sublibraries;
- the part of a sublib directory relevant to one or more specified members.
- the system directory list (SDL)

For full details on the LISTDIR command, see *VSE/Advanced Functions System Control Statements*.

The following are examples of the job streams necessary to produce listings of the various directories.

```
// JOB jobname
// EXEC LIBR
  LISTD LIB=LIBA UNIT=SYSLOG
/*
/&
```

This will cause the system to produce a listing on SYSLOG of the directory information for all sublibraries in the library with the name LIBA.

```
// JOB jobname
// EXEC LIBR
  LISTD SUBLIB=LIBA.PROCS,LIBB.PGMS
/*
/&
```

This will cause the system to produce, on SYSLST, a listing of the directory information for sublibrary named PROCS in the library named LIBA and for the sublibrary named PGMS in the library named LIBB.

```
// JOB jobname
// EXEC LIBR
  ACCESS SUBLIB=LIBC.PROD
  LISTD PGM1.PHASE DATA2.DATA
/*
/&
```

This will cause the system to produce, on SYSLST, the information from the directory for sublibrary PROD in the library named LIBC which is relevant to the member named PGM1 of the type PHASE and the member named DATA2 of the user type DATA.

```
// JOB jobname
// EXEC LIBR
  LISTD SDL UNIT=SYSLOG
/*
/&
```

This will cause the system to produce, on SYSLOG, a display of the system directory list (SDL).

23. Printing the Hard-Copy File (PRINTLOG)

Each line that appears on the screen of the display console 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 // EXEC PRINTLOG and press ENTER. The following message appears on the screen:

   ENTER ONE OR MORE OF THE FOLLOWING OPTIONS:

   ALL OR NEW, AR OR BG OR F1 OR F2 OR F3 ... ETC, ROUTED OR REPLY OR SYSCONS OR SUPPRESSED,
   A,D,I,E,U,JOBNAME=NAME,MM/DD/YY

2. Select the desired options and enter them. The options must be separated by commas; intervening blanks are not permitted.

   You may enter a null line (just press ENTER). The system then uses the option **ALL** by default.

ALL specifies printing of all messages that are on the hard-copy file. (See Note 1).

NEW specifies printing of only those messages that have accumulated since PRINTLOG was last run with only the NEW or ALL option specified, or since the last IPL with SET HC=CREATE. (See Note 1).

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.

AR specifies printing of messages issued by the attention routine.

BG - Fn specifies printing of messages issued by a particular partition; only *one* partition identifier or AR may be specified. n can be from 1 up to the number of foreground partitions supported in your system.

ROUTED specifies printing of all messages directed to an NCCF operator station, plus any replies or commands entered from there.

REPLY specifies printing of all messages for which replies were defined in the message action table, plus their replies.

SYSCONS specifies printing of all messages directed to the system console, plus any replies or commands entered from there.

SUPPRESSED specifies printing of all messages which were suppressed by VSE/OCCF.

A specifies printing of action messages. (See Note 2).

D specifies printing of decision messages. (See Note 2).

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 (if this was specified in the STDOPT command). Enter only the last two digits of the year. Leading zeros may be omitted.

Notes:

1. *If PRINTLOG with the option ALL or NEW is cancelled, the following PRINTLOG with the option NEW starts at the position where the PRINTLOG ALL/NEW was interrupted.*

2. *For asynchronous processing, important reply messages, which are flagged with a plus or minus sign after the partition identifier, are internally marked as action and decision messages. Therefore, those messages are always printed if you specify either A or D.*

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, 1986:

```
9/30/86
```

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.

24. Printing Job Information (LISTLOG)

To gather information about how a particular job has run on the system, use the utility program LISTLOG. This will provide a listing on SYSLST of the following items:

- All job control statements in the job stream;
- All console messages for the job;
- All operator responses for the job;
- Any attention routine messages and commands issued while the job was running.

The LISTLOG utility program can be invoked in two ways:

- Explicitly, by including the job control statement

  ```
  // EXEC LISTLOG
  ```

  in the job stream. The items listed above will be printed for this job.

- Automatically, when a job is cancelled by the operator or by program request. In this case, job control generates a // EXEC LISTLOG statement, which will not, however itself appear on the console or be written to the hard copy file.

LISTLOG derives the information to be printed from the hard copy file. Starting with the beginning of the job in or for which the utility was invoked, it prints all messages and commands relevant to the partition in which the job ran.

Printing stops when the current hard copy file address is reached, not necessarily at end-of-job for the calling job. This may mean that the output contains information relevant to a job which followed the failing job in the same partition.

If a second job cancels due to program request while LISTLOG is running, it will not be invoked again, but will continue printing the information for the relevant partition.

25. Executing Cataloged Programs

Programs that are used more than once are cataloged permanently in a sublibrary after they have been assembled and link-edited.

To execute a cataloged program, use a LIBDEF statement to identify the sublibrary in which the program is cataloged, and 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 sublibrary LIB1.SUBP under the name PROGA; data lines are entered on SYSIPT.

```
Example:
// JOB jobname
      •
      •
      assignment, label and
      LIBDEF statements,
      as required
      •
      •
// LIBDEF PHASE,SEARCH=LIB1.SUBP
// EXEC PROGA
      •
      •
      input data
      •
/*
/&
```

26. Gathering Error Information

When normal system operation is interrupted due to a software or hardware failure, your system programmer or the IBM service representative need information about the error.

Some of the means that are available to analyze error situations are:

- System debugging and problem determination aids;
- Displaying virtual storage;
- Dumps;
- The EREP (environmental recording, editing, and printing) program.

These aids are normally used in response to system messages on SYSLOG. These messages are contained in *VSE/System Package Messages and Codes*, and tell you which debugging aids you should use in a particular situation.

Some procedures for information gathering are described in this manual. Refer to:

- Procedure "19. Displaying the VTOC" on page 83;
- Procedure "20. Displaying the Label Information Area" on page 84;
- Procedure "22. Displaying the Library Directories" on page 87;
- Procedure "23. Printing the Hard-Copy File (PRINTLOG)" on page 89;
- Procedure "24. Printing Job Information (LISTLOG)" on page 92.

If a console message tells you to enter a certain command, you can find details in *VSE/Advanced Functions System Control Statements.*

27. Shutting down the System

There are a number of things you should do **before** you switch off the power at the central processor:

- Issue the ROD command if you want statistical information on the system, which has been accumulated since the previous IPL procedure. The information will be written to the system recorder file (SYSREC).

- If your operator console is a display unit and your hard-copy file resides on an FBA device, issue the ROD command in order to force writing of the last messages on the hard-copy file.

- Unload and remove all tape reels to prevent the tape heads from damaging the tape when the power is switched off.

- Switch off all disk drives and remove the disk packs.

Depending on the complexity of your installation, you may have to do other things in addition to those described above. If you operate a system with a display console, for example, you may want to make a printout of the hard-copy file for easy reference. Your installation manager will give you detailed instructions on how to shut down the system.

Section 4. Control Statements

In your job you will be concerned with entering statements and commands, all of which are contained in *VSE/Advanced Functions System Control Statements*. If you are not yet an experienced operator, you will have to refer to this manual quite frequently.

Syntax Rules

Statements and commands are written according to a few simple rules, which are best discussed in an example. The ASSGN statement/command (with some of its operands only) is used to illustrate these rules (see Figure 11).

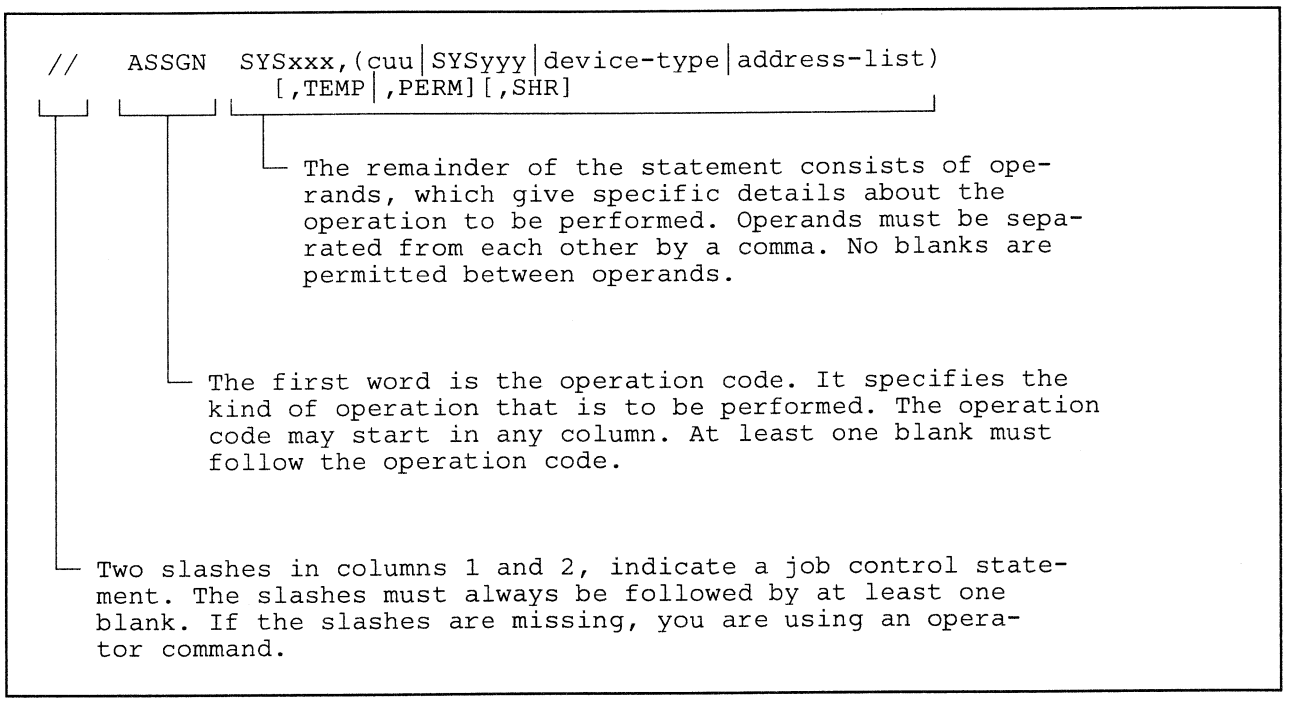


Figure 11. How to Read and Write Statements and Commands

If several operands are shown in braces and separated by a | sign, for example:

```
{cuu|SYSyyy|device-class|device-type}
```

you **must** select one (and only one) of these operands.

If several operands are shown between square brackets and separated by a | sign, for example:

```
[,TEMP|,PERM]
```

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 the braces and square brackets do not appear in the actual statement or command.

Commas, digits, and uppercase letters must be specified exactly as shown in the following examples:

```
// ASSGN SYSLST,00E
// ASSGN SYS001,DISK,SHR
```

Lowercase letters stand for information that you or the programmer must supply; for instance, **cuu** may be specified as 00E.

The values you have to specify may be *alphanumeric* or *numeric*. The lower-case letters used in the command description show which form of specification is required:

x and **y** indicate alphanumeric (or alphameric) characters. These are the letters A - Z, the digits 0 - 9 and the special characters @, $ and #.

n and **m** indicate numeric characters.

These may be:

- *decimal*, that is the numbers 0 - 9;
- *hexadecimal*, that is, the numbers 0 - 9 or the letters A-F.

The description of a command always states explicitly if a numeric value must be hexadecimal.

Operands or operand values which are underlined are default values. That is, they are assumed if no operand is supplied.

When entered through SYSLOG, all job control commands and statements (except the overriding statements following an EXEC statement with the OV option) and all attention routine commands may be continued on subsequent lines. The existence of a continuation line is indicated by a minus sign (and, optionally, one or more blanks) immediately following the last delimiting comma on the current line. Continuation starts at the first position of the next line.

Figure 12 on page 102 shows a list of all operator commands. A detailed description of each attention routine and job control command, see *VSE/Advanced Functions System Control Statements.*

Operator Commands

This section gives you an overview of the following types of operator commands:

- IPL commands;
- Job control commands;
- Attention routine commands.

The main difference between these types of commands lies in:

The time at which they can be entered, and

The way in which they are entered.

IPL commands are accepted only during the IPL procedure. They provide information the system needs before processing can start. With ASI (Automated System Initialization) all IPL commands and all job control statements needed to start the partitions are cataloged in procedures which are executed automatically after IPL is started.

IPL commands can be entered from the console printer-keyboard, the display console, a card reader, or from the diskette.

When entering IPL commands from the console printer-keyboard, press END after every command.

When entering IPL commands from the display console, press ENTER after every command.

When IPL is complete job control tries to read job control statements/commands either from the console (if the IPL commands were entered from console/diskette) or from the device to which SYSRDR was assigned (if the IPL commands were entered from this device).

Job control commands are accepted immediately after IPL and between jobs and job steps. They can be entered either from the console printer-keyboard, from the display console, or from the device to which SYSRDR is assigned.

When entering job control commands from the console, press END or ENTER after typing in a command.

Job control commands to be read from SYSRDR are simply placed in the job stream, that is, they are treated in the same way as job control statements. (In certain cases, however, their effect is different.)

Attention routine commands can be entered after IPL during system operation, when the attention routine identifier **AR** is displayed on the console. They can be entered only from the console. Simply type in the

command and press END or ENTER. (At a printer-keyboard console, press the REQUEST key first.)

If the attention routine is not available (the identifier **AR** is not displayed at the console), enter the command **RC** (Request Communication). You can then enter any attention routine command.

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.

Processor storage allocated to an active partition for program execution in real mode can be reduced only by the ALLOC R **job control** command. When a partition is inactive, you may use either the ALLOC R job control command or the ALLOC R 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 *VSE/Advanced Functions System Control Statements*.

Figure 12 on page 102 shows to which type(s) the commands belong, and summarizes their functions.

Command	Meaning	Type of Command		
		IPL	Job Control	Attention Routine
ADD	Add a device to the PUB table	X		
ALLOC	Allocate virtual or real storage to partitions		X	X
ALTER	Alter 1 to 16 bytes in virtual storage			X
ASSGN	Assign a logical unit		X	
BANDID	Display print band identifier on 4248 printer			X
BATCH	Initiate or resume processing in a partition			X
CANCEL	Cancel execution of current job		X	X
CLOSE	Close a logical unit		X	
DEF	Assign SYSCAT, and SYSREC	X		
DEL	Delete a device from the PUB table	X		
DLA	Define separate label area	X		
DLF	Define lock file	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 unavailable		X	
DVCUP	Make a device available		X	
EXEC	Execute a program or procedure		X	
FREE	Free device from RESERVED status			X
HOLD	Hold unit assignments after subsequent UNBATCH until end of next job		X	
ID	Provide user identification		X	
IGNORE	Ignore command just entered		X	X
LFCB	Load forms control buffer			X
LIBDEF	Define library chains		X	
LIBDROP	Drop library chain definitions		X	
LIBLIST	List library chain definitions		X	
LISTIO	List current I/O assignments		X	
LOG	Log (print) job control statements and commands		X	X
LUCB	Load universal character-set-buffer			X
MAP	List virtual storage allocations		X	X
MODE	Set RMS recording mode or make status inquiry			X

Figure 12 (Part 1 of 2). Alphabetical List of Operator Commands

Command	Meaning	Type of Command		
		IPL	Job Control	Attention Routine
MSECS	Change or display time-slice for partition balancing		X	
MSG	Communicate with a program			X
MTC	Perform magnetic tape control operations		X	
NEWVOL	Resume processing, required volume is mounted			X
NOLOG	Suppress logging of job control		X	X
ONLINE	Simulate 'device and ready' status			X
OVEND	Indicate end of override statements for cataloged procedure		X	
PAUSE	Interrupt processing at end of job or job step		X	X
PRTY	Modify or display partition		X	X
PWR	Passes commands to POWER. (Do not enter at the console.)		X	
RC	Force acceptance of attention routine commands			X
REPLID	Display prefixes of messages with outstanding responses			X
RESERV	Reserve device for VSAM space management usage			X
RESET	Reset temporary I/O device assignments		X	
ROD	Record on demand and end-of-day message		X	
SET	Set date, clock and time zone	X	X	
SETDF	Set and/or reset default values for the 3800 printer			X
SETMOD	Set mode for 8809 tape			X
SETPARM	Define or set symbolic parameter		X	
SETPRT	Set the 3800 printer		X	
SIZE	Change storage allocation		X	X
START	Initiate or resume processing in a partition		X	X
STDOPT	Set standard options		X	
STOP	Stop processing in a partition		X	
SVA	Reserve space in the shared virtual area	X		
SYS	Set system values	X		
TPBAL	Display or modify TP balancing			X
UCS	Load Universal Character Set buffer		X	
UNBATCH	Deactivate a partition		X	
UNLOCK	Release locks set by another system			X
VOLUME	Display currently-mounted disk volumes			X

Figure 12 (Part 2 of 2). Alphabetical List of Operator Commands

Glossary

This glossary includes definitions developed by the American National Standards Institute (ANSI) and the International Organization for Standardization (ISO). This material is reproduced from the American National Dictionary for Information Processing, copyright 1977, by the Computer and Business Equipment Manufacturers Association, copies of which may be purchased from the American National Standards Institute, 1430 Broadway, New York, New York 10018.

ANSI and ISO definitions are marked by an asterisk (*).

access control. A facility that provides for programmed control of a user's authorization to access protected data.

access method. A technique for moving data between virtual storage and input/output devices.

ACF/VTAM. An enhanced program version of the IBM Virtual Telecommunications Access Method (VTAM).

*** address.** (1) An identification, as represented by a name, label, or number, for a register, location in storage, or any other data source or destination such as the location of a station in a communication network. (2) Loosely, any part of an instruction that specifies the location of an operand for the instruction.

address translation. The process of changing the address of an item of data or an instruction from its virtual address to its real storage address. See also dynamic address translation.

alphameric characters. The alphabetic characters A through Z, the digits 0 through 9, and the characters #, $, and @.

alternate track. One of a number of tracks set aside on a disk pack for use as alternatives to any defective tracks found elsewhere on the disk pack.

application program. A program written for or 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.

asynchronous operator communication. A facility which allows the operator to defer the reply to a system message that requires a response.

auxiliary storage. Data storage other than real storage; for example storage on magnetic tape or disk. Synonymous with external storage, secondary storage.

batch partition. Partition in which batch processing takes place.

batch processing. Sequential processing of programs submitted to the computer as a collection (batch) of jobs that are separated from one another by job control statements.

block. A set of logical records, usually the unit of data for a transfer of data from processor storage to an I/O device and vice versa.

blocking. Combining two or more logical records into one block.

book. A group of source statements written in any of the languages supported by VSE and stored in a sublibrary.

buffer. An area of storage that is temporarily reserved for use in performing an input/output operation, into which data is read or from which data is written. Synonymous with I/O area.

byte. A sequence of eight adjacent binary digits that are operated upon as a unit and that constitute the smallest addressable unit of the system.

card punch. A device to record information in cards by punching holes in the cards to represent letters, digits, and special characters.

card reader. A device which senses and translates into machine code the holes in punched cards.

cardless system. A computing system configured without a card reader or card punch, but with an IBM diskette input/output unit.

catalog. To enter a phase, module, book, or procedure into one of the system or private libraries.

*** central processing unit.** A unit of a computer that includes the circuits for controlling the interpretation and execution of instructions. Abbreviated CPU.

central processor. Synonym for central processing unit.

channel. (1)* A path along which signals can be sent, for example, data channel, output channel. (2) A hardware device that connects the central processor and its associated storage with the I/O control units.

CKD (count-key-data) device. A disk storage device storing data in the format: count field, normally followed by a key field, followed by the actual data record. The count field contains, among others, the address of the record in the format CCHHR (CC = cylinder number, HH = head or track number, R record number) and the length of the data; the key field contains the record's key (search argument). See also FBA (fixed-block-architecture) device.

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.

component. A functional part of VSE (for example: job control program, VSE/POWER).

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. See also system control programming (SCP).

control unit. A device that controls the reading, writing, or display of data at one or more input/output devices.

count-key-data device. See CKD device.

CPU. See central processing unit.

data file. See file.

data integrity. See integrity.

data management. A major function of the operating system; the function involves organizing, storing, locating, retrieving, and updating data.

data security. See security.

deblocking. The action of making the first and each subsequent logical record of a block available for processing one record at a time.

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 detection and isolation of malfunctions or mistakes.

direct access. Retrieval or storage of data by a reference to its location on a volume, other than relative to the previously retrieved or stored data.

directory. An index that is used by the system control program to locate one or more sequential blocks of program information that is stored on direct access storage.

disk pack. A direct access storage volume containing magnetic disks on which data is stored. Disk packs are mounted on a disk storage drive, such as the IBM 3330 Disk Storage Drive.

diskette. A flexible magnetic oxide coated disk suitable for data storage and retrieval.

dump. (1) To copy the contents of all or part of virtual storage. (2) The data resulting from the process as in (1).

dynamic address translation (DAT). (1) The change of a virtual storage address to an address in processor storage during execution of an instruction. (2) A hardware function that performs the translation.

dynamic partition balancing. See partition balancing.

ECPS/VSE (extended control program support:VSE). An implementation of the virtual storage concept which does not require software participation in the translation of virtual addresses into real addresses.

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.

external storage. See auxiliary storage.

FBA (fixed-block-architecture) device. A disk storage device storing data in blocks of fixed size; these blocks are addressed by block number relative to the beginning of the file.

FBA block. A unit of data of fixed length on which the FBA architecture is based.

*** 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.

fixed-block-architecture. See FBA device.

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. In a hard wait state, the system does not accept any commands. Operations can be restarted only by a new initial program load (IPL).

*** 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.

index. In data management, a table used to locate the records of a file.

initial program load (IPL). The initialization procedure that causes the operating system to be loaded into virtual storage and to commence operation.

integrity. Preservation of data or programs for their intended purpose.

*** I/O.** Abbreviation for input/output.

IPL. See initial program load.

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 such as job step start and stop times and counts of I/O operations.

job control. A program that is called into storage 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.

K. When referring to storage capacity, 1024 bytes.

*** key.** One or more characters associated with an item of data; these characters are used to identify it or control its use.

label. An identification record for a volume of data on magnetic tape, disk, or diskette; an identification record of an individual file on such a volume of data.

label information area. An area on a DASD device that stores label information read from job control statements or commands. Synonymous with label area.

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 program that maintains, services, and organizes the system and private libraries.

library. A collection of sublibraries which may, but need not, be related by some common characteristics. For example, all programs and procedures connected with the payroll could be in one library, all those connected with stock-keeping on another.

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 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 VSE, to bring a program phase from a sublibrary into virtual storage for execution.

M. When referring to storage capacity, 1,048,576 bytes (equal to 1024K bytes).

message. See error message, operator message.

*** multiprogramming.** A mode of operation that provides for the interleaved execution of two or more computer programs by a single central processor.

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 (program).** A module (program) that is the output of an assembler or compiler and is input to a linkage editor. 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 from 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 action, such as mounting a tape reel, or informing him of specific conditions within the system, such as an error condition.

page. (1) A 2K or 4K block of instructions, of data, or of both which can be transferred between processor storage and the page data set. (2) To transfer instructions, data, or both between processor storage and the page data set.

page data set. One or more extents in auxiliary storage in which pages are stored.

page frame. 2K block of processor storage that can contain a page.

page in. The process of transferring a page from the page data set to processor storage.

page out. The process of transferring a page from processor storage to the page data set.

page pool. The set of all page frames available for paging virtual-mode programs.

paging. The process of transferring pages between processor storage and the page data set.

*** parameter.** A variable that is given a constant value for a specific purpose or process.

partition. A contiguous area of virtual storage available for the execution of programs.

partition balancing. A facility which allows the user to specify two or more or all partitions of the system to have their processing priorities changed dynamically such that each of these partitions receives approximately the same amount of CPU processing time.

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 central processor.

phase. The smallest complete unit that can be referred to in a sublibrary containing executable programs.

printer. A device that writes output data from a system on paper.

priority. A rank assigned to a partition that determines its precedence in receiving processing time.

private library. A user-owned library that is separate and distinct from the system library.

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.

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 line or list formed by items in a system waiting for service; for example, tasks to be performed or messages to be transmitted in a message switching system. (2) To arrange in, or form, a queue.

real address. The address of a location in real storage.

real address area. The area of virtual storage where virtual addresses are equal to real addresses.

real mode. The execution mode of a program that may not be paged.

real storage. The storage of a 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.

restore. To write data previously copied from disk (onto magnetic tape, for example) back onto disk.

*** routine.** An ordered set of instructions that may have some general or frequent use.

SCP. See system control program.

SDL. See system directory list.

secondary storage:. See auxiliary storage.

security. Prevention of access to, or use of, data or programs without authorization (see also access control).

sequential file. A file whose records are organized on the basis of their successive 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 (SVA). An area located in the highest address range of virtual storage. It can contain a system directory list (SDL) of highly used phases, resident programs that can be shared between partitions, and an area for dynamic allocation to components of VSE.

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.

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.

stand-alone dump. A program that displays the contents of the registers and of virtual storage and that runs independently of (is not controlled by) the system.

standard label. A fixed-format identification record for a tape or disk file. Standard labels can be written and processed by the system.

storage protection. An arrangement for preventing unauthorized access to storage.

sublibrary. A collection of data readily available to the system. A sublibrary is a subdivision of a library, and the data are stored as members in the sublibrary.

supervisor. A component of the control program. Coordinates the use of resources and controls the flow of operations in a data processing system.

SVA. See shared virtual area.

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 control programming (SCP). IBM-supplied programming support that is fundamental to the operation and service of the system.

system directory list (SDL). A list containing directory entries of frequently 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 volume is mounted.

system residence volume. The volume on which the basic operating system and all related supervisor code is located.

telecommunication. Data transmission between a computing system and remotely located devices via a unit that performs the necessary format conversion and controls the rate of transmission.

teleprocessing balancing. A supervisor function that allows telecommunication users who have concurrent batch processing in a paging environment to obtain better telecommunication response times at the expense of slower job execution in the batch partition(s).

terminal. (1) * A point in a system or communication network at which data can either enter or leave. (2) Any device capable of sending and receiving information over a communication channel.

throughput. The total volume of work performed by a computing system over a given period of time.

*** track.** The portion of a moving storage medium, such as a magnetic tape, or disk, that is accessible to a given reading head position.

transient area. An area within the control program and fixed in processor storage, used for temporary storage of executable high-priority code (transient routines).

unit record. A card containing one complete record; a punched card. Also a line-printer output record.

universal character set (UCS). A printer feature that permits the use of a variety of character arrays.

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 370-mode, the area of virtual storage whose addresses are greater than the highest address of the real address area.

virtual mode. The operating mode of a program which may be paged.

virtual storage. Addressable space that appears to the user as processor storage from which instructions and data are mapped into processor storage locations.

Virtual Storage Access Method (VSAM). An access method (available as the licensed program product VSE/VSAM) for direct or sequential processing of fixed and variable-length records on direct access devices; designed for use in a virtual storage environment.

Virtual Telecommunications Access Method (VTAM). An IBM program (available as a licensed program product) that control communication between application programs and terminals in a telecommunications network.

volume. That portion of a single unit of storage media which is accessible to a single read/write mechanism, for example, a disk pack or a reel of magnetic tape. (2) A recording medium that is mounted and dismounted as a unit.

volume table of contents. A table on a direct access volume that describes each file on the volume. Abbreviated VTOC.

VSAM. See Virtual Storage Access Method.

VSAM catalog. A file, with an index, containing extensive file and volume information that VSE/VSAM requires to locate files, to allocate and de-allocate storage space, to verify the authorization of a program or operator to gain access to a file, and to accumulate usage statistics for files.

VTAM. See Virtual Telecommunications Access Method.

VTOC. See volume table of contents.

Bibliography

The VSE/Advanced Functions Planning and Installation, GC33-6193

VSE/Advanced Functions System Management Guide, SC33-6191

VSE/Advanced Functions System Control Statements, SC33-6198

VSE/System Package Messages and Codes, SC33-6181

VSE/Advanced Functions Service Aids, SC33-6195

VSE/POWER Installation and Operations Guide, SH12-5329

VSE/POWER Remote Job Entry User's Guide, SH12-5328

System/370 Model 138 Procedures, GC38-0016

System/370 Model 148 Procedures, GC38-0017

System/370 Model 155 Procedures, GA22-6966

System/370 Model 158 Procedures, GC38-0025

IBM 3031 Processor Complex Operator's Guide, GC38-0033

IBM 3033 Operator's Guide, GC38-0031

IBM 4331 Processor, Operating Procedures and Problem Determination Guide, GA33-1525

IBM 3340 Fixed Head Feature User's Guide, GA26-1632

DOS/VSE IBM 3800 Printing Subsystem Programmer's Guide, GC26-3900

Index

Special Characters

A

B

C

D

E

F

V

W

Numerics

IBM Virtual Storage Extended/
Advanced Functions
Operation
Order No. SC33-6194-02

READER'S COMMENT FORM

This form may be used to communicate your views about this publication. They will be sent to the author's department for whatever review and action, if any, is deemed appropriate. Comments may be written in your own language; use of English is not required.
IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

Note: *Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.*

Possible topics for comments are:

Clarity Accuracy Completeness Organization Coding Retrieval Legibility

If you wish a reply, give your name and mailing address: _____

_____

_____

_____

What is your occupation? _____

Number of latest Newsletter associated with this publication: _____

Thank you for your cooperation. No postage stamp is necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page).

Note: Staples can cause problems with automated mail sorting equipment. Please use pressure sensitive or other gummed tape to seal this form.

Cut or Fold Along Line

SC33-6194-02

Reader's Comment Form

Fold and Tape | **Please Do Not Staple** | Fold and Tape

No Postage necessary if mailed in the United States

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

Postage will be paid by addressee:

International Business Machines Corp.
Department 6R1BP
180 Kost Road
Mechanicsburg, PA 17055

Fold and Tape | **Please Do Not Staple** | Fold and Tape

IBM®

IBM Virtual Storage Extended/
Advanced Functions
Operation
Order No. SC33-6194-02

READER'S COMMENT FORM

This form may be used to communicate your views about this publication. They will be sent to the author's department for whatever review and action, if any, is deemed appropriate. Comments may be written in your own language; use of English is not required.
IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

Note: *Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.*

Possible topics for comments are:

Clarity Accuracy Completeness Organization Coding Retrieval Legibility

If you wish a reply, give your name and mailing address: _____

_____

_____

_____

What is your occupation? _____

Number of latest Newsletter associated with this publication: _____

Thank you for your cooperation. No postage stamp is necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page).

Note: Staples can cause problems with automated mail sorting equipment. Please use pressure sensitive or other gummed tape to seal this form.

Cut or Fold Along Line

SC33-6194-02

Reader's Comment Form

Fold and Tape **Please Do Not Staple** Fold and Tape

No Postage necessary if mailed in the United States

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

Postage will be paid by addressee:

International Business Machines Corp.
Department 6R1BP
180 Kost Road
Mechanicsburg, PA 17055

Fold and Tape **Please Do Not Staple** Fold and Tape

IBM®

IBM Virtual Storage Extended/
Advanced Functions
Operation
Order No. SC33-6194-02

READER'S COMMENT FORM

This form may be used to communicate your views about this publication. They will be sent to the author's department for whatever review and action, if any, is deemed appropriate. Comments may be written in your own language; use of English is not required.
IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

Note: *Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.*

Possible topics for comments are:

Clarity Accuracy Completeness Organization Coding Retrieval Legibility

If you wish a reply, give your name and mailing address: _____

_____

_____

_____

What is your occupation? _____

Number of latest Newsletter associated with this publication: _____

Thank you for your cooperation. No postage stamp is necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page).

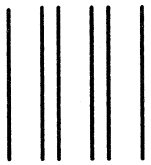
Note: Staples can cause problems with automated mail sorting equipment. Please use pressure sensitive or other gummed tape to seal this form.

Cut or Fold Along Line

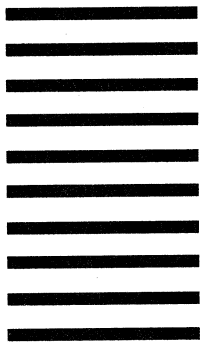
SC33-6194-02

Reader's Comment Form

Fold and Tape **Please Do Not Staple** Fold and Tape



No Postage necessary if mailed in the United States



BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

Postage will be paid by addressee:

International Business Machines Corp.
Department 6R1BP
180 Kost Road
Mechanicsburg, PA 17055

Fold and Tape **Please Do Not Staple** Fold and Tape

IBM®

IBM Virtual Storage Extended/
Advanced Functions
Operation
Order No. SC33-6194-02

READER'S COMMENT FORM

This form may be used to communicate your views about this publication. They will be sent to the author's department for whatever review and action, if any, is deemed appropriate. Comments may be written in your own language; use of English is not required.
IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

Note: *Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.*

Possible topics for comments are:

Clarity Accuracy Completeness Organization Coding Retrieval Legibility

If you wish a reply, give your name and mailing address: _____

_____

_____

_____

What is your occupation? _____

Number of latest Newsletter associated with this publication: _____

Thank you for your cooperation. No postage stamp is necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page).

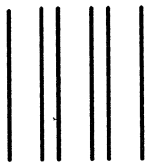
Note: Staples can cause problems with automated mail sorting equipment. Please use pressure sensitive or other gummed tape to seal this form.

Cut or Fold Along Line

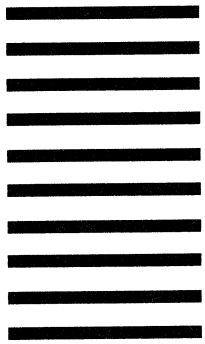
SC33-6194-02

Reader's Comment Form

Fold and Tape **Please Do Not Staple** Fold and Tape



No Postage necessary if mailed in the United States



BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

Postage will be paid by addressee:

International Business Machines Corp.
Department 6R1BP
180 Kost Road
Mechanicsburg, PA 17055

Fold and Tape **Please Do Not Staple** Fold and Tape

IBM®

SC33-6194-02

IBM Virtual Storage Extended/
Advanced Functions
(File No. S370/4300-40)
Printed in U.S.A.