GC33-6108-1
File No. S370/4300-20

Program Product

Introduction to the VSE System

IBM

GC33-6108-1
File No. S370/4300-20

Program Product

Introduction to the VSE System

VSE/Advanced Functions
Release 3

Program Number 5746-XE8

IBM

Second Edition (July 1980)

This is a major revision of and obsoletes GC33-6108-1 and Technical Newsletter GN33-9277. This edition applies to Release 3 of VSE/Advanced Functions, Program Number 5746-XE8, with the required DOS/VSE SCP and to all subsequent releases until otherwise indicated in new editions or Technical Newsletters. Changes are continually made to the information herein; before using this publication in connection with the operation of IBM systems, consult the latest *IBM System/370 and 4300 Processors Bibliography,* GC20-0001, for the editions that are applicable and current.

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

Publications are not stocked at the 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 reader's comments is provided at the back of this publication. If the form has been removed, comments may be addressed either to:

International Business Machines Corporation
Department 812BP
1133 Westchester Avenue
White Plains, New York 10604

or to:

IBM Laboratory
Dept 3250, Programming Publications
Schoenaicher Strasse 220
D-7030 Boeblingen
Germany

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

Preface

This manual is a general discussion of the IBM VSE system, an operating system whose name has been derived from Virtual Storage Extended in DOS/VSE. The manual provides new users of a VSE system with an introduction to the major computing services available with the minimum operating system support: VSE/Advanced Functions and the required complementary DOS/VSE system control programming (SCP) code. This minimum support is referred to in this publication as VSE/Advanced Functions.

For users familiar with DOS/VS, the publication gives a summary of the differences between the Release 34 DOS/VS SCP and VSE/Advanced Functions. The reader is expected to have a basic knowledge of data processing.

The information in this manual is presented as follows:

- A brief discussion of what VSE is, including a summary of the major computing services available with VSE/Advanced Functions – **Chapter 1.**
- A more detailed discussion of available computing services, highlighting their usefulness – **Chapter 2.**
- An overview of the system generation process – **Chapter 3.**
- A discussion of the advantages of VSE/Advanced Functions over the Release 34 DOS/VS SCP – **Chapter 4.**
- A discussion of program compatibility between this release of VSE/Advanced Functions on one hand and earlier releases of VSE/Advanced Functions, DOS/VS, and DOS on the other – **Chapter 5.**
- An overview of additional licensed and nonlicensed programs available from IBM for use at an installation operating with VSE/Advanced Functions – **Chapter 6.**
- An overview of devices supported by VSE/Advanced Functions – **Appendix A.**
- A description of the operating environment of VSE/Advanced Functions – **Appendix B.**
- An overview of the documentation for VSE – **Appendix C.**

Supplementary information about hardware functions and machine instructions may be found in the applicable principles of operation manuals:

IBM 4300 Processors Principles of Operation, GA22-7070
IBM System/370 Principles of Operation, GA22-7000

Contents

List of Figures

Summary of Amendments

This edition contains new or changed information as follows:

On newly available facilities and/or services:

- Performance improvement – Fast CCW translation for I/O requests with IORB.
- Usability improvements:
  - Program input via job control
  - Further simplification of system generation
- Availability improvements:
  - Removal of the job information block (JIB) table
  - New I/O error handling concept
  - Improved handling of missing device interrupts
  - Enhanced Maintain System History program

On the purpose of the IBM program product **VSE/Operator Communication Control Facility.**

Information on technical details irrelevant to the intended purpose of the publication (giving an overview of the available computing services) has been deleted.

The list of logical unit names that may be used in programs or control statements and commands has been deleted. This list is contained in the following VSE/Advanced Functions publications:

System Management Guide
System Control Statements

Editorial changes to improve the publication's usability have been made throughout the manual. Changes significant in content are indicated by a vertical bar to the left of the changes.

A list of services and facilities that have become available with **Release 2 of VSE/Advanced Functions** is given below for your information and convenience.

- New device support:
  - PIOCS support for the IBM 3890 Document Inscriber
  - A tracing facility for the IBM 4331 integrated communication adapter

- Performance improvements:
  - Sharing of data on DASD, including libraries across computing systems
  - Extended support for multiprogramming (up to twelve partitions) and subtasking (up to 208 subtasks)
  - Execution of frequently used services from the shared virtual area rather than the logical transient area in the supervisor

- Usability improvements:
  - Device independence for files on disk
  - New initial program load functions and simplified supervisor assembly
  - Improved label processing
  - Linkage editor and assembler work files in VSE/VSAM managed data space if the VSE/VSAM Space Management for SAM feature is installed
  - More ease of use with VSE/VSAM space management – simplified job control language, dynamic device assignment, early volume mount requests
  - Installability of CICS/VS under control of the Maintain System History program

- Availability improvements:
  - Chaining (concatenating) libraries by defining search order chains
  - Avoiding initial program load after a not-ready condition for the system residence device, the page data set, or a private core image library

- Serviceability improvements:
  - Altering a core image library under control of the Maintain System History program
  - Writing job-related console communication to the system printer.

Chapter 1: The VSE System

The VSE system, an operating system whose name has been derived from Virtual Storage Extended in DOS/VSE, consists of: (1) the licensed VSE/Advanced Functions support plus the required complementary system control programming (SCP) code, and (2) any IBM supplied and user-written programs that may be required to meet an installation's data processing needs. VSE/Advanced Functions and the complementary SCP code, collectively referred to in this publication as VSE/Advanced Functions, comprise the minimum operating system support needed at a VSE controlled installation. VSE and the hardware controlled by VSE combine to form a complete, effective computing facility.

VSE/Advanced Functions provides the support needed for processing in a multi-programming environment; it includes the potential for build-up of additional computing power and for adaptation to changing data processing requirements. By including in your installation's VSE, for example, the VSE/Interactive Computing and Control Facility (VSE/ICCF), you change the installation's operating characteristics from batch to interactive. By additionally installing the VSE/Access Control – Logging and Reporting program, you can reduce considerably the risk of inadvertent or intentional misuse of data. More information about available programming support is given in "Chapter 6: Additional Licensed and Nonlicensed Programs."

VSE/Advanced Functions

Through the system generation procedure, you can tailor the IBM supplied VSE/Advanced Functions support to your installation's requirements. Final adaptation to these requirements would normally take place during initial program load, which is an almost fully automated operation.

VSE/Advanced Functions operates on central processors as follows:

- On any IBM System/370 processor, Models 115 through 158, with a processor storage of a least 160K bytes.
- On an IBM 3031.
- On IBM 4300 series central processors, which are equipped with extended control program support for VSE (ECPS:VSE).

If VSE/Advanced Functions operates in System/370 mode, the processor's storage must not exceed 8,192K bytes.

Summary of Available Computing Services

VSE/Advanced Functions relieves programmers and operators of a great deal of work. The programs and routines of this support (they are also referred to as components) are stored online on disk; they are immediately accessible whenever they are needed.

Following is a brief overview of the major computing services available with the components of VSE/Advanced Functions.

- **Automated system start-up.**
  This function initiates operation with the computing system under control of VSE/Advanced Functions; it allows for the initial program load procedure of an installation's VSE to be automated nearly 100 percent. An automated system start-up may include the allocation of storage, the loading of print control buffers, the setting of control values for later execution of programs, and the starting of partitions; it may include the start-up procedure for continuously running programs such as IBM's program products VSE/POWER and VSE/ICCF.

  As in the past, however, your operator can do an initial program load (IPL) interactively by submitting the required IPL control information from the system console, for example, and subsequently start up the system's partitions by submitting appropriate commands.

- **Controlling jobs.**
  Before an application or a system program can be executed, the supervisor loads the job control program, one of the components of VSE/Advanced Functions. This program determines which of the system resources were used by the previously executed program and are now available again. Job control also determines which system resources will be required by the program that is about to be executed next.

- **Resource management.**
  The supervisor of VSE/Advanced Functions, which is loaded into processor storage by the initial program load function, controls overall system operation. Primarily, the supervisor controls the utilization of system resources (such as processor storage and processing time) by programs, user-written or IBM supplied.

- **Library services.**
  VSE/Advanced Functions includes a number of programs collectively referred to as librarian. The librarian provides library services such as cataloging programs and procedures (sets of control statements), deleting them (if necessary), and updating them in system and private libraries as required; condensing libraries is another service function provided by the librarian.

  The linkage editor, another of the components of VSE/Advanced Functions, processes language translator output to make this output executable. A language translator output processed by the linkage editor becomes one or more phases in a program library, ready for being loaded from this library and executed.

- **Data management.**
  The input/output and file organization routines included in VSE/Advanced Functions relieve the programmer of the detailed and cumbersome programming associated with the transfer of data between auxiliary storage and programs that are executed in virtual storage.

- **Sharing of data across computing systems.**
  VSE/Advanced Functions includes the controlling capability required for the sharing of data on disk by two or more computing systems that operate under control of VSE. Components of VSE/Advanced Functions that access disk devices include the required support for the sharing of data on disk across computing systems.

- **Communication between system and operator.**
  To get jobs done, a human being – the operator – must initiate system operation. VSE/Advanced Functions provides status information, some automatically, and other on operator demand. This allows the operator to monitor system operation and to respond quickly to situations that require human intervention.

- **System utilities.**
  VSE/Advanced Functions includes a number of programs that help to improve the overall efficiency of a VSE controlled data processing installation: magnetic tape or disk volumes are to be initialized; data is to be copied for backup purposes and may have to be restored. These programs are collectively referred to as system utilities.

- **System serviceability and debugging aids.**
  Reliability, availability, and serviceability of an installation are important considerations in planning for the operation of a computing system: program debugging, for example, needs to be done whenever a new application is being implemented. VSE/Advanced Functions includes programs and programming tools that are designed to contribute in these areas of computer operation.

- **Assembling of programs.**
  VSE/Advanced Functions includes an assembler. This program translates assembler language programs into machine-readable (object) language. The assembler includes a macro processor.

The next chapter of this manual expands on the computing services summarized above, except for automated system start-up. This service is sufficiently covered in the summary above and need not be dealt with any further.

Chapter 2: VSE/Advanced Functions – Computing Services

This chapter expands on the information provided in the preceding chapter about major computing services available with VSE/Advanced Functions. The services discussed in this chapter are:

- Controlling jobs
- Resource management
- Library services
- Data management
- Sharing of data across computing systems
- Communication between system and operator
- System utilities
- System serviceability and debugging aids
- Assembling of programs.

Controlling Jobs

After the system has been successfully started by initial program load, the job control program is in control, and this program is ready to accept input for processing.

job and job stream

The unit of work you submit to the system for processing is called a job. A succession of jobs presented to a computer is a job stream. Within each job, one or more programs may be executed. These programs may be IBM programs, such as a compiler that translates a user-written source program into object code, or a user program that is already in executable format and processes data.

job control statements

A job, and the environment in which it is to run, must be defined by means of job control statements. Job control statements specify, for example, whether programs are to be compiled, link-edited, and/or executed; from which library or device system or user-written programs are to be loaded; what files (of data) they are to process and where these files reside.

When handling jobs, VSE/Advanced Functions does the following:

- It provides for automatic transition from job to job with a minimum of operator intervention.
- On the basis of job control statements supplied by you, it assigns actual I/O devices to logical device names specified in the programs that are to be executed.
- It loads executable programs from online disk libraries into storage for execution.
- It handles program termination.

user exit routines

If you are programming in assembler language, you have the option of including user-written exit routines in the program. Then, if the program is interrupted by an operator request, for example, or if an abnormal termination occurs, the supervisor passes control to the proper user-exit routine. If the program should terminate abnormally, the supervisor moreover gives information on the cause of the abnormal termination. The exit routine can examine this data and take appropriate action.

job accounting

You can write a simple program to keep track of processor usage and the usage of the various I/O devices by accessing the job accounting data accumulated by the system. This enables you to:

- Charge usage of the system to the various users.
- Check on efficient use of I/O devices.
- Maintain a record of system operation and use and thus plan for new applications and additional devices.

The necessary information (input for your program) is provided by the job accounting interface, an optional facility that you can specify for system generation. With this facility, the following information is automatically gathered by the system for each job step and stored in a table:

- Job name, date, and the partition in which the job is running.
- Start and stop times of the job, processor time used by the job, processor time used by the operating system (overhead), and processor idle time chargeable to the partition.
- Optionally, counts of I/O operations of the various devices used by the job.

At the end of each job step, your accounting program is automatically loaded to process the accounting information.

Resource Management

The main measure of an installation's efficiency is throughput, that is, the amount of work handled in a certain period of time. The major resources to be examined when discussing an installation's throughput are (1) processor time and its use, (2) virtual storage space and its exploitation by problem programs, and (3) disk library space and its employment.

VSE/Advanced Functions provides support that improves system throughput by allowing for:

- Efficient use of processor time through multiprogramming or multitasking. Both multiprogramming and multitasking are discussed later in this section.
- Optimum use of the available storage through multiprogramming and virtual storage support. Virtual storage support is dealt with separately later in this section.
- Optimum use of disk space for data by utilizing the available resource share control support. More information about this support is given under "Sharing of Data Across Computing Systems" later in this chapter.
- Optimum use of disk space for libraries through the available library services and the relocating loader facility. The library services,

including the loader facility, are discussed under "Library Services" later in this chapter.

System throughput may be further improved by utilizing the services of VSE/POWER, a licensed IBM program product which is designed primarily for more efficient use of processor time and unit record I/O devices. For a summary on VSE/POWER, refer to Chapter 6.

In order to examine the various facilities and services in more detail, it is first necessary to look briefly at the storage organization of a system.

Storage Organization

VSE/Advanced Functions can be generated and set up to organize the available storage for optimum operating conditions at the particular installation. VSE/Advanced Functions always reserves a certain amount of the available storage for its own use leaving the remainder to application program use.

partitions

The storage available for the execution of programs is divided into areas called partitions. You can generate your VSE/Advanced Functions to support concurrent execution of programs – multiprogramming – in two to twelve partitions. Normally, one program is contained and executed in each of these partitions; therefore, if you generate your operating system to support multiprogramming in eight partitions, for example, you can request the system to load up to eight programs for concurrent execution.

At the time of system generation, you define the number of partitions to be supported; partition sizes, however, you define in a system start-up procedure for IPL or between jobs, whichever is appropriate.

Figure 2-1 shows the storage organization of a system set up for 8-partition operation. In addition, it illustrates the IBM-set processing priorities for the various partitions; processing priorities are discussed under "Multiprogramming" below.

Multiprogramming

Multiprogramming provides for efficient utilization of processor time. Without multiprogramming, the central processor of an installation would be idle during most of its power-on time.

Figure 2-2 illustrates how processor time utilization can be improved by multiprogramming. As the illustration shows, the processor is idle for a considerable amount of time even in a three-partition system. The number of partitions that you should set up for most efficient operation depends on a number of installation specific factors such as the nature of the applications to be processed, the total amount of available processor storage, and the mix of the programs that are to be executed concurrently.

processing priorities

With programs taking turns executing in a multiprogramming environment, processing must obviously proceed according to a set of priorities. The priority of a program for receiving processor resources is dependent upon the processing priority of the partition in which the program resides. By assigning a job to a certain partition, a programmer or an operator, therefore, assigns the priority of that partition to the job as well. The supervisor, of course, always has the highest priority.

The default priorities for the partitions are as shown in Figure 2-1. However, these default values can be changed during automatic system start-up or by means of an operator command.

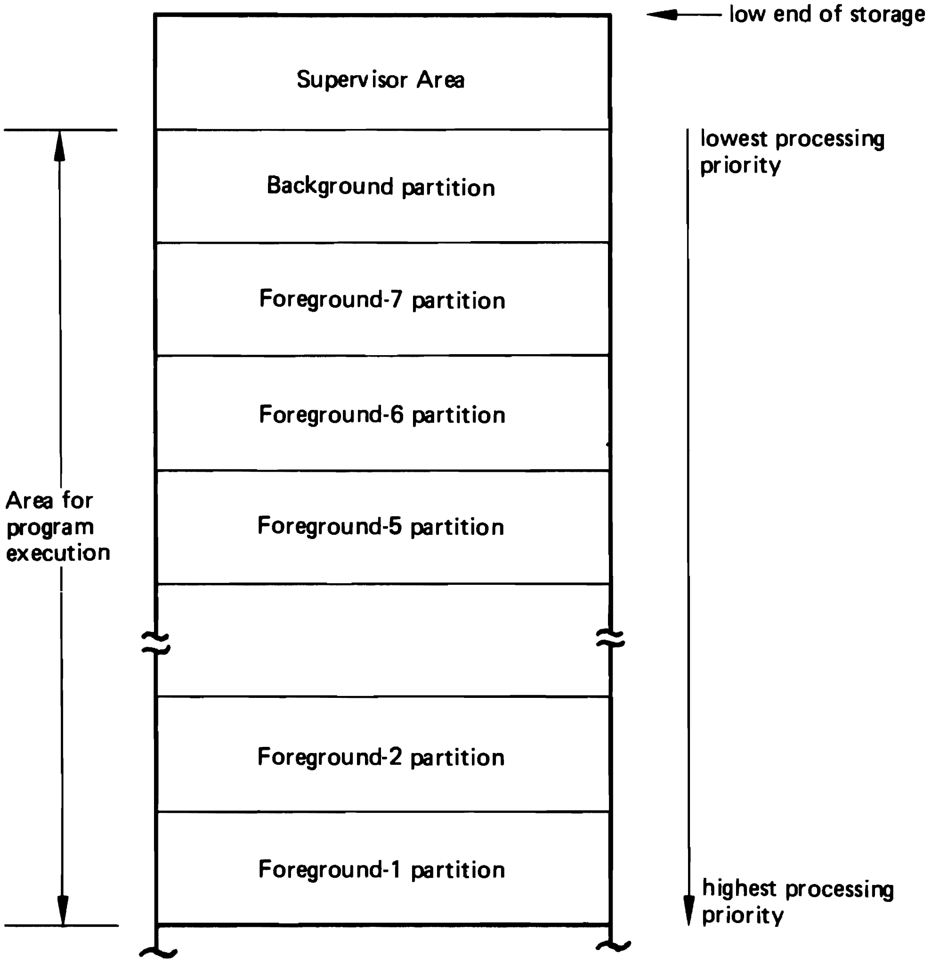


Figure 2-1. Storage organization and default processing priorities for an 8-partition system

partition balancing

In addition you can, by means of the same operator command, define two or more partitions to participate in *partition balancing.* This means, that those partitions have their processing priorities changed dynamically to privilege that partition which has the lowest record of central processor usage for a specific time interval.

cross-partition event control

For the implementation of complex applications, it may be necessary to synchronize program execution in two or more partitions. VSE/Advanced Functions includes macros that facilitate the implementation of this type of applications.

Multitasking

Multitasking, a special form of multiprogramming, provides for making still more efficient use of processor time.

main and subtasks

A program that has been loaded into a partition for execution is, for the system's supervisor, the main (or only) task running in that partition. The available multitasking facility, however, allows you to define one or more specific sections of the program's code as subtasks.

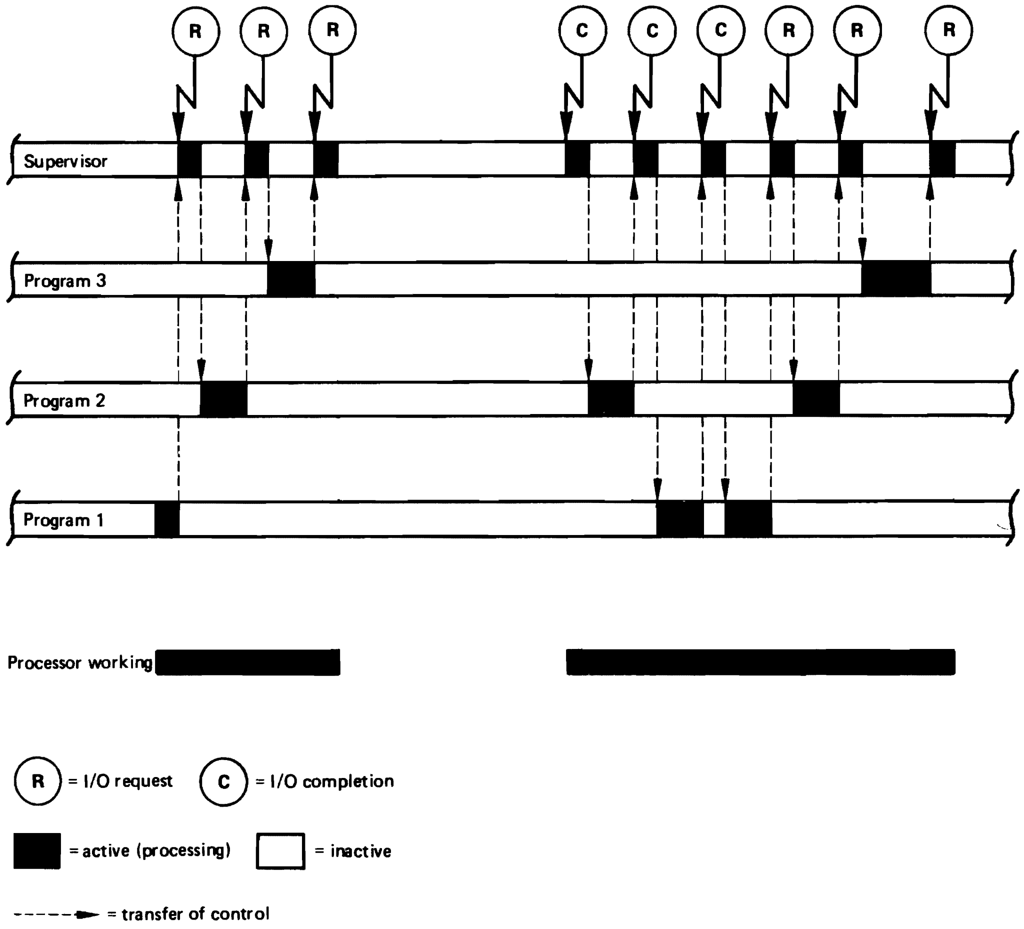
Assume that a program with subtasks is being executed and that it has activated its subtasks. Whenever the execution of this program is interrupted because it has to wait for the completion of an event (for example, an I/O operation), the supervisor gives control over the processor to one of those subtasks rather than to a program in another partition. In other words, sections of a program defined as subtasks compete for the allocation of processor time as if they were independent programs.

The supervisor can give control to a subtask of a program only as long as no program (or subtask of a program) in a partition of higher processing priority is waiting to receive control.

Multitasking allows you to activate (attach) up to 31 subtasks within a partition, and you may have up to 208 subtasks active within your system at any point in time.

Virtual Storage Support

A computing system without virtual storage support restricts its user to an address space limited by the storage physically contained in the system's central processor.



This diagram assumes that the processing priorities from high to low are as follows: Program 1, Program 2, Program 3.

Figure 2-2. Processor usage in a 3-partition system

A user of DOS on a System/360 processor, for example, could not run a program that exceeded the limits of the largest of his system's partitions. When a user of DOS was running programs that were smaller than the partitions of the system, the storage fragments not used by those programs could not be used for the execution of additional programs. The system had no means of dynamically reallocating these fragments to other programs.

VSE/Advanced Functions removes these limitations without any extra programming effort on your part.

Program Execution in a Virtual Storage Environment

The instructions of a program, and also related data, must be in processor storage in order to be executed. However, only one instruction of one program can be executed by the central processor at any point in time; therefore, all other instructions of the same or any other program need not be in processor storage at the same point in time. The design and implementation of the virtual storage concept makes use of this operating characteristic of a central processor.

program pages

page data set

Programs to be executed in a virtual storage environment are sectioned by VSE/Advanced Functions into blocks of 2K bytes called program pages. If not all of the pages of a program fit into processor storage, the system writes pages currently not needed for program execution to a disk file called page data set and continues executing instructions contained in program pages that are still in processor storage. Figure 2-3 shows the relationship of the page data set and processor storage; it illustrates the execution of four programs in a virtual storage environment; it may help you to understand the subsequent discussion.

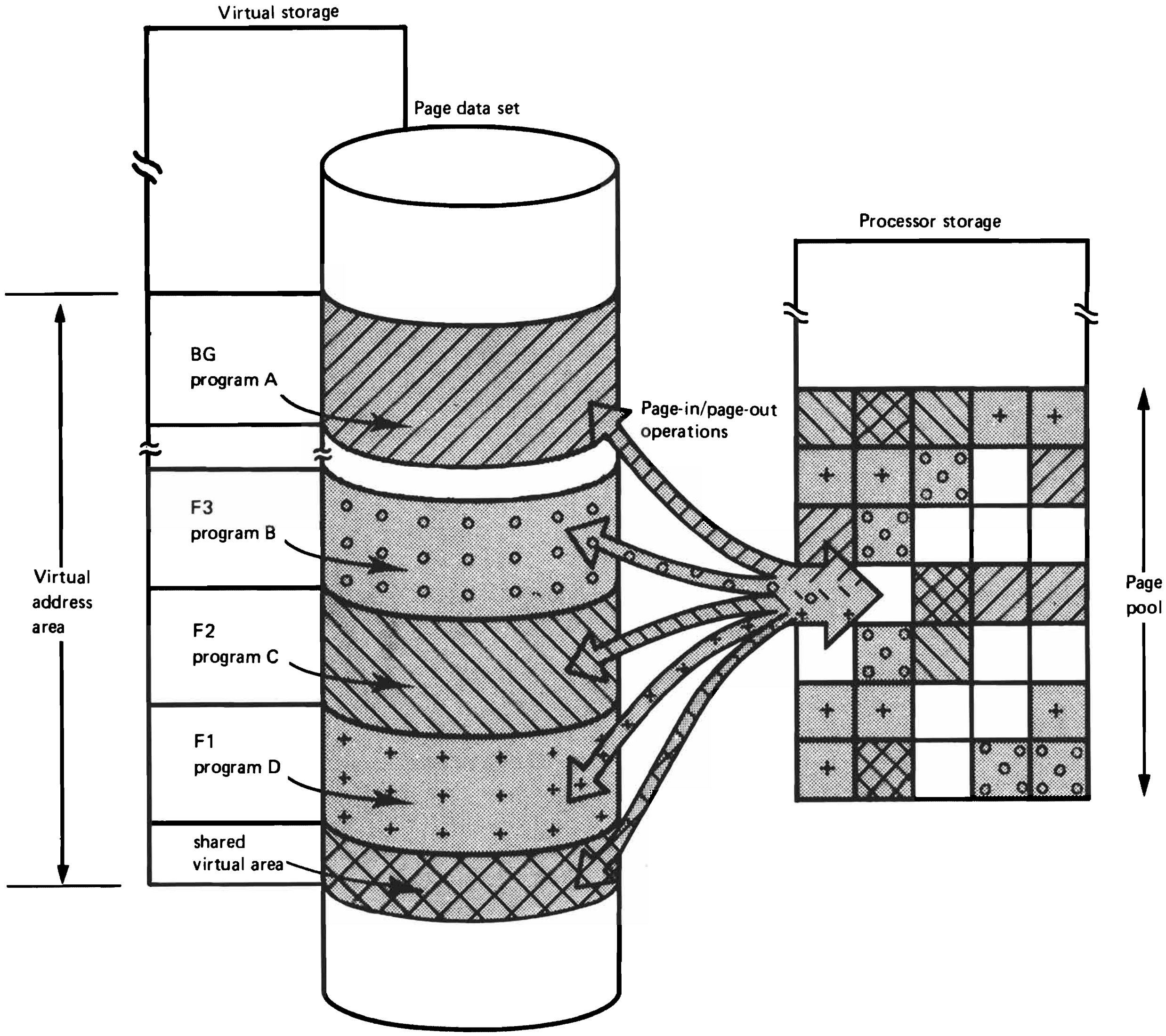


Figure 2-3. Four programs executing in a virtual environment

page frame
page-in

If, during program execution, an instruction refers to a program page which is not in processor storage or control is transferred to such a page, the system automatically suspends execution of the program, retrieves that page from the page data set and places it into a 2K-byte section (page frame) of processor storage. Once this operation (which is referred to as page-in) is complete, execution of the interrupted program continues.

Chances are that no page frame is available in processor storage to accommodate a program page needed for continued execution of a program. In this case, the system automatically frees a page frame; it selects a program page not recently referenced and writes

page-out

it to the page data set if necessary. This operation (which is called page-out) is necessary if the page was altered while in processor storage or if it had not been written to the page data set previously.

'page pool

Page frames that are not used by the supervisor and are not needed for the execution of programs in real mode – this is discussed below – comprise the system's page pool.

shared virtual area

As Figure 2-3 shows, there is a shared virtual area at the high end of virtual storage. The area is primarily used to accommodate phases which can be used concurrently by programs that are being executed in the system's partitions. Execution of phases from the SVA is faster than loading the same phases from a core image library into a partition.

execution in real mode

Some programs cannot tolerate page-in or page-out operations. For these programs, execution in real mode is indicated. Examples of programs that should run in real mode are time or performance critical applications (for example, programs that include MICR stacker-select routines or real-time operations).

The execution of a program in real mode requires a sufficiently large area of processor storage to be allocated to the partition in which the program is to run. The page frames used by that program are taken away from the page pool; they are therefore no longer available for the execution of programs which run in other partitions in virtual (not real) mode.

Allocating Storage to Partitions

A generated VSE system has an address space, called virtual storage. This address space, one or more extents on disk, starts at zero and can extend up to 16,777,216 (or 16,384K) bytes. How much of this address space will be used in a particular system depends upon a number of factors, for example: the size of the installation's central processor's storage; the number of partitions and their size; the characteristics of the installation's programs and operating environment.

Based on these factors, trade-offs have to be made to arrive at an optimal virtual storage size for the requirements of a particular installation. An operational VSE system provides for virtual storage larger than the available processor storage and, normally, smaller than the maximum limit.

allocation examples

Consider Figures 2-4 and 2-5, which give allocation examples for an 8-partition VSE system as follows: Figure 2-4 for a VSE system on an IBM System/370 Model 138 and Figure 2-5 for a VSE System on an IBM 4331 processor, both with a processor storage of 1024K bytes.

Allocation for	Program run mode	Usage of allocated area	Processor storage allocation	Virtual allocation
Supervisor	real	system control	120K	
Background	virtual	system service, large applications, tests (not frequently used)	0K	256K
Foreground-7	virtual	program-test runs (not frequently used)	0K	384K
Foreground-6	virtual	normal batch production (under control of VSE/POWER)	0K	128K
Foreground-5	virtual	normal batch production (under control of VSE/POWER)	0K	128K
Foreground-4	virtual	normal batch production (under control of VSE/POWER)	0K	128K
Foreground-3	virtual	urgent jobs (not scheduled)	0K	128K
Foreground-2	virtual	VSE/POWER with remote job entry	* 48K	346K
Foreground-1	virtual	VSE/ICCF supporting locally attached terminals	** 40K	1,024K
Shared virtual area	virtual	shareable phases and VSE usage	0K	512K
Total of processor storage allocated			208K	
Total of virtual storage allocated to partitions and the shared virtual area				3,034K
Processor storage available as page frames for program execution:				
Size of processor storage				1,024K
minus:				
resident supervisor routines (page frames they occupy are fixed in storage)				- 120K
* page frames that may be fixed by VSE/POWER (it releases page frames no longer needed)				- 48K
** page frames that normally are fixed by VSE/ICCF (it releases page frames no longer needed)				- 40K
				= 816

Figure 2-4. Example of storage allocation in an 8-partition system – System/370 mode

Allocation for	Program run mode	Usage of allocated area	Virtual allocation
Supervisor	real	system control	120K
Background	virtual	system service, large applications, tests (not frequently used)	256K
Foreground-7	virtual	Program-test runs (not frequently used)	384K
Foreground-6	virtual	normal batch production (under control of VSE/POWER)	128K
Foreground-5	virtual	normal batch production (under control of VSE/POWER)	128K
Foreground-4	virtual	normal batch production (under control of VSE/POWER)	128K
Foreground-3	virtual	urgent jobs (not scheduled)	128K
Foreground-2	virtual	VSE/POWER with remote job entry	346K
Foreground-1	virtual	VSE/ICCF supporting locally attached terminals	1.024K
Shared virtual area	virtual	shareable phases and VSE usage	512K
Total of virtual storage allocated			3,154K
Total allocated to partitions and SVA:			
allocated virtual storage		3,154K	
minus area occupied by the supervisor:		- 120K =	3,034K
Processor storage available as page frames for program execution:			
Size of processor storage			1,024K
minus:			
resident supervisor routines (pages frames they occupy are fixed in storage)			- 120K
page frames that may be fixed by VSE/POWER (it releases page frames no longer needed)			- 48K
page frames that normally are fixed by VSE/ICCF (it releases page frames no longer needed)			- 40K
processor storage required to accommodate micro code routines (normally not more than the value given here)			- 128K
			= 688K

Figure 2-5. Example of storage allocation in an 8-partition system – ECPS:VSE mode

Library Services

One powerful feature of a VSE installation is its range of libraries that enables programming data to be stored online, readily accessible.

VSE/Advanced Functions supports the following types of libraries as both system and private: core image, relocatable, source statement, and procedure. The system libraries are contained in the system residence file (on the SYSRES volume); private libraries can be stored on disk volumes mounted on drives of a different type than the one that accommodates the SYSRES volume. However, the device type used must be supported by VSE/Advanced Functions as a system residence device.

The librarian programs of VSE/Advanced Functions provide for fast execution of library service, organization, and update functions for all of the supported library types. Additional support for efficient use of the core image library is available through the linkage editor program and the loader of VSE/Advanced Functions.

This section of the manual discusses the purpose of the supported libraries, highlights how these libraries may be used, and provides an overview of the available library services.

Purpose of the Libraries

core image library

A core image library serves to catalog programs (in units called phases) that have been processed by the linkage editor and are ready for being loaded into storage for execution. A VSE system must include a system core image library in order to have certain system programs online whenever they are needed.

Normally, the linkage editor builds phases in relocatable format. For a relocatable phase, the loader of VSE/Advanced Functions modifies addresses as required when the phase is being loaded for execution; such a phase can, therefore, be loaded at a storage address different from the one for which it was link-edited.

You may, if you wish, cause the linkage editor to build phases in nonrelocatable format. A nonrelocatable phase is loaded for execution into a partition beginning at the address which was computed by the linkage editor during the link-edit run for that phase.

Programs that have been written as self-relocatable are linked and cataloged by the linkage editor as self-relocatable phases. Any address or constant relocation to be done for a self-relocatable phase must be handled within that phase itself after it has been loaded into storage for execution.

relocatable library

A relocatable library contains the output of language translators in units called modules. When the linkage editor encounters a reference to a name which is not defined within the module that is being processed, the linkage editor searches the available relocatable libraries for a module with this name specified in the external reference. If the search is successful, the module found is linked with the module(s) being processed.

Explicitly specified modules from a relocatable library can be included with modules being link-edited. In this way, sections of code that are used by a number of different programs need be written, translated, and cataloged in relocatable object format only once.

source statement library

A source statement library contains sequences of source language statements, called books. The library consists of a number of sublibraries used, for example, to store macro definitions. When the assembler encounters a source statement with an unknown operation code, it tries to retrieve the book with the same name as this unknown operation code from the available source statement libraries. The assembler then substitutes the code found in one of the libraries for the source statement in question.

Similarly, when a compiler encounters a reference to a book in a source statement library, it gets the specified book from that library and substitutes it for the reference in the source program that is being processed.

procedure library

A procedure library is used to catalog frequently used sets of job control and linkage editor statements in card image format. These sets of control statements are referred to as cataloged procedures.

For example, instead of submitting frequently required job control statements over and over again, you can request inclusion of a cataloged procedure containing these control statements in any control statement set that may require them. This approach can save a considerable amount of key punch or data entry effort.

The statements contained in a cataloged procedure may be modified while these statements are being processed.

A cataloged procedure may contain inline data, that is, data which is read by the associated application or system program from the device that is used for reading the necessary job control statements.

Using the Libraries

An operational VSE system requires one core image library to be online at all times – the system core image library, which is part of the SYSRES file. If, at your installation, programs are to be assembled or compiled for later execution, you need also at least one relocatable and one source statement library, which may be system (included in your installation's SYSRES file) or private. The use of at least a system procedure library is considered an operational must: an automated system start-up requires a number of procedures to be cataloged in the system procedure library; without a system procedure available at your installation, your operator must start up the system using the time consuming interactive IPL method.

private libraries

The use of private libraries can be of advantage as shown by the examples below:

- A number of small libraries instead of a few large libraries greatly eases library maintenance.

- Maintaining one or more private libraries per partition, each of those libraries on a separate disk pack, reduces disk arm movement on the volume that contains the system library and provides for faster access to your cataloged programs in general.

- In order to reduce program retrieval time, private libraries may be chained (also known as concatenated) by establishing search order chains, one per partition if necessary – chaining of libraries is further discussed below.

  Libraries containing programs that are frequently executed in a particular partition would be placed at the head of the associated chain.

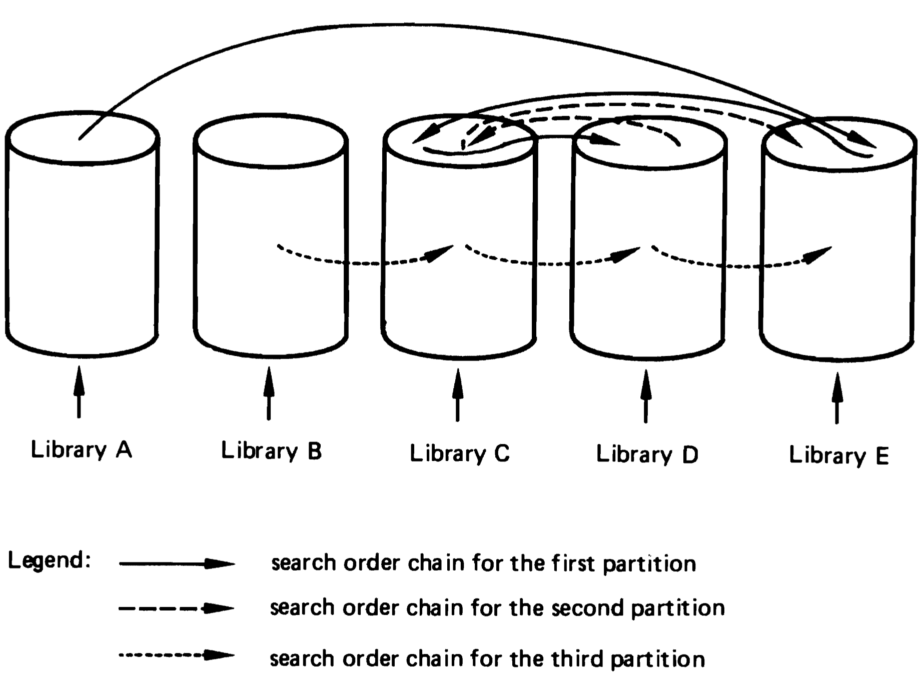
- Private libraries may be shared (between partitions and across systems) besides being chained. This provides for less frequent mounting and remounting of private libraries; less frequently a need for assignment of private libraries, virtually only once per shift; increased system availability because only one set of program or procedure data need be online and serviced instead of one set for each system or perhaps even for each partition.

- In a testing environment, master copies of programs can be kept intact in one library while modifications to work copies of the programs can be used for testing modifications as required.

chaining of libraries

VSE/Advanced Functions allows you to chain private libraries of the same type for access from each of an installation's partition as follows: up to 15 temporarily (for the duration of a job) defined libraries, up to 15 permanently (until next IPL) defined libraries, and the particular system library. The system library may or may not be included in a search order specification. If that library is not included, VSE/Advanced Functions adds it to the end of the permanent chain of libraries, if one exists, or else to the end of the temporary chain.

A private library may be included in a search order chain defined for two or more or all partitions of a system. This concept is illustrated in Figure 2-6 for three partitions (see also "sharing libraries" below).



Library A
Library B
Library C
Library D
Library E
Legend:
search order chain for the first partition
search order chain for the second partition
search order chain for the third partition

Figure 2-6. Example for the chaining of private libraries

sharing libraries

Private libraries may be shared not only across partitions but also across systems if they reside on disk devices that are supported for cross-system data sharing (see the section "Sharing of Data Across Computing Systems" later in this chapter).

A private library that is to be shared across systems may be specified in any library chain of each of the sharing systems.

Available Library Service Programs

VSE/Advanced Functions contains library service programs that perform maintenance and service functions for all libraries. These functions include:

- Cataloging, renaming, and deleting any element in a library.
- Printing any library element or directory.
- Punching any library element.
- Comparing two libraries that are to be merged, listing the library members that exist in the source library but not in the target library, and generating appropriate merge control input for a subsequent merge run. However, this service function is available only for libraries residing on count-key-data devices and defined in the DOS/VS way.
- Copying all elements from one library to another library of the same type, or copying only those elements that do not yet exist in the destination library.

- Condensing a library and changing its size and location.

- Creating a new system disk pack and creating private core image, relocatable, and source statement libraries.

- Link-editing a program to make it executable and writing this executable program to a core image library.

- Relocating a program when it is loaded for execution.

Data Management

The data management support of VSE/Advanced Functions allows you to select an access method which is suited best for the data management requirements of a given application. Some basic questions regarding these requirements are:

- What type of **file organization** is best suited to the processing requirements for the file?

- What are the **processing requirements** of a file:

  - How many records of the file will be accessed each time the file is to be processed?
  - How many records have to be added to or deleted from the file each time it is to be processed?
  - How often within a given period of time (week or month) is there a need for processing the file?
  - Will a sequential organization meet all of the processing requirements?
  - Is direct access of individual records of the file a processing requirement?

- On what **medium** (magnetic tape, cards, disk, etc.) is the file to be stored?

- What **data security** considerations should apply to the file during and after its processing cycle (for example, who besides the owner of the data may have read access or write access or both; is there a need for that data to be kept under lock and key)?

This section highlights the access methods of VSE/Advanced Functions; it indicates which file organization is supported by each of these methods; it addresses the data management support available for programs that are (or have been) written in a high-level programming language. The section also gives a brief overview of the available data protection facilities. VSE/VSAM, another access method available from IBM as a licensed program, is highlighted in Chapter 6.

For a more detailed discussion of data management topics, see *VSE System Data Management Concepts.* A more detailed description of suitable data protection facilities is contained in *Data Security Under the VSE System.*

Sequential Access Method (SAM) and Organization

file organization

Sequential organization means that records physically follow one another in a sequence usually determined by one or more control fields within each record. Examples of control fields are name or employee-number in a personnel file, or catalog number or part number in an inventory file.

Sequential organization is the most widely used method of data organization and is supported for all device types except telecommunication terminals. Card files, print files, diskette unit files, and magnetic tape files are always organized sequentially, simply because the physical characteristics of those devices require the reading or writing of one record after another. Files on disk frequently are also organized sequentially, in control number sequence.

space management

Disk space management for SAM files can be greatly facilitated at VSE installations that use VSE/VSAM, a licensed access method separately available from IBM. VSE/VSAM allows you to include (create) SAM files in VSAM managed data space.

data access

The Sequential Access Method (SAM) can create a sequential file from (normally) sorted records and subsequently retrieve those records for sequential processing. In addition, by utilizing certain macros (in programs written in assembler language), sequential files on disk or tape may be positioned to specific blocks prior to reading or writing. Records from sequential disk files may be **updated**, meaning that each record may be written back onto its original location after having been changed by the program.

Indexed Sequential Access Method (ISAM) and Organization

The physical characteristics of a disk make it practicable to retrieve a record from any location in a file instead of having to go from one record to the next starting at the beginning of the file until the desired record has been located. ISAM is one of the access methods through which you can exploit these characteristics.

data access

From a file that has been created using ISAM, records may be retrieved:

- **Sequentially** – the access method makes records available, one after the other, in ascending order by record key.
- **Randomly** – the access method locates and reads the record identified by a record key that must be made available by the particular user program.

ISAM is available only for the following IBM disk devices:

2311 and 2314/2319
3330, 3333 Models 1 and 2
3340
3344 in 3340 simulation mode
3350 Model 1 in 3330 compatibility mode

To bypass these device restrictions, IBM recommends that you install VSE/VSAM, a licensed program available for operation under a VSE system. For more information on VSE/VSAM, see Chapter 6.

Direct Access Method (DAM) and Organization

DAM concentrates on random retrieval only and accomplishes this function efficiently. DAM requires, however, that you establish a direct relationship between the keys of the records and their physical addresses on disk. This means that the program, by using the key of a record, can calculate or look up in a table the corresponding record address and either directly store the record (on output) or directly retrieve it (on input). Greater burden and responsibility is placed on the programmer; the benefit is a potentially faster record retrieval for specialized applications such as reservation systems or transaction inquiry programs.

DAM is available for programs that process data stored on count-key-data disk devices supported by VSE/Advanced Functions.

Telecommunication Access Methods

The telecommunication access methods you can use with VSE are:

- Advanced Communications Function for VTAM (ACF/VTAM) or, if you use an IBM 4331 processor, Advanced Communications Function for VTAM Entry (ACF/VTAME)

- Basic Telecommunication Access Method – Extended Support (BTAM-ES)

These access methods are available as licensed IBM programming support; they make the required telecommunication services available through assembler language macros.

Normally, you would use the services of these telecommunication access methods through other IBM licensed programming support: the VSE/Interactive Computing and Control Facility (VSE/ICCF) or the Customer Information Control System (CICS/VS) or both. This licensed programming support relieves the programmer of the burden of coding telecommunication access routines and yet makes your computing system's services available for use via a remote terminal.

For more information about the available IBM telecommunication access methods and other telecommunication-oriented licensed programming support, see Chapter 6 of this publication.

High-Level Language Support for Data Management Functions

For use on a VSE system, IBM provides compilers for the following languages:

- COBOL
- PL/I
- RPG II
- FORTRAN IV

Each language has data management facilities based on those provided by the access method available as part of VSE/Advanced Functions or as separate licensed programming support. For more information about this support, see either the pertinent IBM-provided language documentation or the publication *VSE System Data Management Concepts.*

Protection of Data

It is important to protect an installation's data – programs and files, that is – against accidental or intentional misuse or destruction. Under a VSE system, programs in different partitions, possibly invoked from a terminal, may attempt to retrieve and update the same file or even the same record simultaneously; moreover, the installation's online disk storage might contain classified data which may not be accessible by any or all users.

VSE/Advanced Functions provides facilities that help to protect your data against misuse or destruction. These facilities are briefly highlighted below; for a more detailed description, refer to *Data Security Under the VSE System.* The facilities are:

- Resource protection through macros

  VSE/Advanced Functions includes macro support which allows for efficient control of the sharing of resources either on a subtask level or on a cross-partition or cross-system level.

- Access authorization checking and access logging

  An efficient access authorization method is available if your VSE system includes the licensed VSE/ICCF. If you employ this method at your installation, anyone who wants to use the system needs to specify the correct user identification and the associated password as set up by the installation's security administrator. If VSE/Access Control – Logging and Reporting (another licensed IBM program) is also part of your VSE system, you can obtain formatted resource-access reports.

- File labeling

  File labels are records associated with files stored on magnetic tape, on disk, or on diskette. They provide unique identification for the files and for the volumes on which the files reside.

- Protection against duplicate assignment

  In a multiprogramming environment, no two programs in different partitions may gain access to the same device, except when this device is a disk unit (a single disk unit may contain a number of files for processing by programs in more than one partition of the same or of different systems). VSE/Advanced Functions provides the necessary protection against duplicate assignments.

- DASD file protection

  The facility prevents programs from reading or writing outside the specified file limits.

- Track hold specification

  This facility can be used to avoid that two or more programs executing under a VSE system concurrently update the same track on a CKD disk device or the same block on an FBA disk device.

Supported I/O Devices

VSE/Advanced Functions supports the full range of I/O devices that are attachable to the supported central processors. Some of these devices, however, are supported only on an EXCP macro level; in other words, in order to use such devices, you would have to code your own channel programs.

For a complete list of supported central processors and I/O devices, refer to "Appendix A: Device Support." This appendix provides also information about software imposed restrictions, if any. Detailed information about possible device configurations with the various supported processors is given in the following publications:

> *IBM System/370 System Summary,* GA22-7001.
>
> *IBM 4300 Processors Summary, Input/Output and Data Communications Configurator,* GA33-1523.

Sharing of Data Across Computing Systems

If data must be online for concurrent execution of programs on two or more (up to four) systems at one installation, that data nevertheless need be stored on disk only once. As opposed to nonshared data that is stored on a separate disk volume for each of an installation's systems, shared data need be updated only once if and when an update of this data becomes necessary.

share options

Data on disk may be shared by two or more programs as follows:

- A program updates the data and no other program can access that data until this update operation is finished.

- A program updates the data and other programs can access that data concurrently for read-only purposes.

- A program reads the data and does not tolerate concurrent update by another program.

- A program reads the data and it does tolerate concurrent update by another program.

resource share control

The support for sharing of data on disk includes the necessary resource share control facilities. This programming support is available for any type of IBM DASDs supported by VSE/Advanced Functions, except 2311, 2314/2319, and 3310.

For files contained in a VSAM data space, VSE/VSAM controls concurrent access of those files in accordance with the share options you defined for them in your DEFINE command(s). Other than that, no extra programming effort is required for VSAM files.

Communication Between System and Operator

VSE/Advanced Functions provides facilities that ensure timely and effective interaction between the operator and the system.

The operator of a VSE controlled data processing installation is normally called upon to:

- start up the system.
- prepare the I/O devices; for example, mount tapes and disk packs for each individual job.
- respond to system or program originated requests for information or action.

More specifically, the operator can:

- invoke inquiries to obtain information about the operating system's status.
- cancel the execution of a job; for instance, in the case of an unending program loop.
- gather problem information, if necessary, with the help of the available problem determination aids.

For its part, the system:

- alerts the operator when a condition requiring operator intervention occurs.
- provides information such as the allocation of resources (virtual storage and I/O devices, for example).

system-to-operator communication

Communication from the system to the operator is in the form of messages written on the operator communication device, which may be the screen of a display operator console or a printer-keyboard. The messages describe the situation that may or may not require operator action.

operator-to-system communication

Communication from the operator to the system is in the form of commands and replies to messages.

The operator can initiate communication at any time by pressing the Request key on the console keyboard. Subsequently, the system is ready to accept a command.

Operator replies to messages are generally short 1-word answers such as RETRY, IGNORE, or CANCEL. For fast response, the operator can allow the system default option to take effect in most situations requiring an intervention.

the hard-copy file

Most IBM processors use display-type operator consoles. Since a display operator console does not produce a hard copy of the communication between the system and the operator, VSE/Advanced Functions includes support for writing the communication onto disk into a hard-copy file.

A utility program is available for producing a listing of the operator-to-system communication recorded in the hard-copy file. The operator can request a selective redisplay of messages that were written into this file.

operator-to-application program communication

In addition to operator-to-system and system-to-operator communication, VSE/Advanced Functions supports direct operator-to-application program communication. This facility could be useful, for example, for inquiry to an inventory file at unpredictable times. The inquiry routine could be included in a program that always occupies a partition and be invoked by the operator when desired.

reducing message traffic

An installation's job mix may be such that the system writes a considerable number of messages to the central operator's console. VSE/Operator Communication Control Facility (VSE/OCCF), an IBM program product, can help reducing this message traffic significantly either by suppressing certain messages (for example those which require no operator attention) or by routing messages to another operator station. For more information on VSE/OCCF, refer to Chapter 6.

System Utilities

The primary objective of a data processing installation is to do productive work, that is, to execute the installation's application programs whenever there is a need for processing the data these programs are concerned with. However, smooth and efficient operation requires the execution of utility type services that are not directly concerned with the control of execution of application programs. The subsequent paragraphs highlight some of those services, which are available through these components of VSE/Advanced Functions: Device Support Facilities and System Utilities.

file-to-file copying

Often, there is a need for copying or listing card image files. VSE/Advanced Functions includes support that can handle this type of operations with little programming effort on your part.

initialize disk or tape

load print buffers

Frequently, the execution of user problem programs involves job preparation steps such as initializing a disk volume or a volume of magnetic tape. Other examples of job preparation steps are the loading of the control buffer(s) of a printer with the proper buffer image for forms control or for using a universal character set or for both. VSE/Advanced Functions includes programming support that automates these job preparation steps to a great extent.

assign alternate track (or block)

Occasionally, the system detects a defective track on a CKD disk volume or a defective block on an FBA disk device; in that case, an alternate track (block) must be assigned to the defective one and the data, if any, stored on that defective track or block is to be restored or rebuilt on the alternate one. VSE/Advanced Functions provides programming support that automates these operations.

surface analysis program

DASDs with non-removable disk packs (for example, the IBM 3350 or the IBM 3310) require additional surface analysis and track reclamation support. When a malfunction occurs on a DASD with non-removable disk packs, a program controlled diagnostic approach rather than the frequently used disk-volume swap technique must be applied.

backup and restore

The backup and restore utilities available with VSE/Advanced Functions provide for fast copy and restore operations. In addition, they are useful tools for condensing libraries and for migrating an existing library from one DASD type to another, thus eliminating the need for regenerating your VSE system when you intend to change over to a new DASD type for your SYSRES file.

System Serviceability and Debugging Aids

reliability, availability, serviceability (RAS)

The RAS facilities of VSE/Advanced Functions are designed to maintain a high degree of trouble-free performance of your VSE controlled installation. Debugging procedures are provided to help you in choosing the debugging aid best suited to obtain information about a particular type of error or malfunction. The facilities that make up RAS are:

- **Debugging aids** that provide you with information about the system at the time of a program, operator, or hardware error. This includes facilities for taking dumps of selected areas of virtual storage and for tracing events, such as specific instructions or interrupts.

- **Recovery Management Support** (RMS) that, depending on the severity of the failure, enables the system to recover from most of the error conditions caused by any of the following hardware failures: processor storage errors, central processor instruction errors, I/O channel errors, control unit errors, and device errors. It also records hardware failures on the system recorder file (SYSREC), and it prints messages on the console thus keeping the operator informed about the condition of the installation's hardware.

- The **Environmental Recording Editing and Printing Program** (EREP) that you can use for printing the contents of the system recorder file, pre-formatted on SYSLST. EREP has many options that provide for selective printing of summarized and detailed information about the state of an installation's hardware.

- The **Online Test Executive Program** (OLTEP) that, together with a set of device test programs, constitutes an online test system. Its functions include the diagnosing of I/O errors, the verification of I/O device repairs and engineering changes, and the checking of I/O devices.

- The **Reliability Data Extractor** (RDE) that enables you to record the reason for an IPL procedure as well as the number of IPL procedures carried out on a system over any specified time period, for example, during an operator's shift. RDE also provides for an end-of-day time stamp to be recorded on SYSREC during system operation. This RDE information may be of great value to IBM service personnel.

Another important aspect of system reliability and availability is the correction of known problems in IBM supplied programming support. Correction of a known problem before that problem occurs at an installation may well contribute to the overall availability of a VSE system.

PTF installation

For the correction of problems in IBM supplied programming support, IBM provides program temporary fixes (PTFs) which are to be installed on an operational VSE system. VSE/Advanced Functions contains support which keeps track of this type of system update activity; this support ensures that, for the installation of a PTF, prerequisite PTFs (if any) were installed previously.

Assembling of Programs

VSE/Advanced Functions includes an assembler program primarily as a supervisor generation tool. Through the program's macro processor, tailoring the supervisor routines to system configuration and installation operation requirements is reduced to an automated process controlled by user-specified generation macros.

Occasionally, an application requires the utilization of system functions in a fashion that is not supported by the high-level language(s) available at the installation. An assembler-language subroutine linked to the application program normally is the answer to this kind of a problem.

Routines whose execution depends on the occurrence of specific events within the system (for example, routines of complex telecommunication applications or of programs with input from a magnetic ink character reader) normally are written in assembler language. By coding a routine in assembler language, you can closely control the available computing services of your installation.

IBM supplies a variety of macro definitions in a sublibrary of the source statement library of VSE/Advanced Functions. You can make use of any of these macro definitions in the assembler subroutines that you may have to write.

The foregoing paragraphs on the assembling of programs conclude the discussion of the computing services available with VSE/Advanced Functions; the subsequent chapter deals with system generation.

Chapter 3: System Generation

System generation is the creation of an installation-tailored VSE system, a process that includes:

- Generating a supervisor adapted to the installation and its application needs

- Assembling or compiling and link-editing IBM programs and programs of your own as necessary and also cataloging these programs in the appropriate libraries

- Deleting unnecessary components to free additional disk space

- Editing, formatting, or deleting libraries as required

The process of system generation should be planned carefully; planning for system generation saves you unnecessary repetition during the process; it may save you a repetition of the entire process after a short period of time.

Planning for System Generation

Essentially, planning for system generation consists of:

- Planning the options and estimating the size of the supervisor.

  This entails selecting the options that are to be included in the supervisor and estimating the cost of these options in terms of bytes of processor storage.

  The supervisor of an operational VSE system is composed of the routines that were assembled based on the processing options which had been specified for the assembly. In addition, it includes tables and control blocks which are built by the system during system start-up, and which the supervisor needs for proper interaction with your installation's hardware.

- Planning the contents, organization, and ultimate size of the system and private libraries, and of required system areas on disk.

  This entails distributing the available online disk storage among the libraries needed for daily operation and among system work files and system areas such as the hard-copy file, the system recorder file, and the shared virtual area (which is part of the system's page data set). Important items of consideration are, for example:

  - The number and size of the programs that are to be online

  - The number and size of the programs that should be resident in the shared virtual area

- The impact that the expansion of one library has on other libraries on the same disk volume

VSE/Advanced Functions includes programming support that assists in computing library space requirements; it includes support that merges libraries and condenses newly created libraries.

Shipment of VSE/Advanced Functions

VSE/Advanced Functions as shipped by IBM includes the required complementary SCP code; it is shipped in the form of a system core image library, a system relocatable library, a system source statement library, and a system procedure library. The contents of these libraries are listed in the *Memorandum to Users* that accompanies the shipment.

The system core image library must be retained on the operational volume because it contains, in executable format, the programs that make up VSE/Advanced Functions. The other libraries may be deleted if they are not needed. Given below is a summary of the contents of the system libraries when they are shipped:

- The core image library contains the VSE/Advanced Functions support that can be invoked via job control; in addition, it contains prelinked IOCS modules. Delete procedures (in the procedure library) are available for ease of deletion of components for which there is no need at your installation. VSE/Advanced Functions is shipped link-edited and ready for execution once it is installed.

- The relocatable library contains the support such as data management modules in object-code format, but not link-edited. You may, if you wish, link data management modules into your problem programs. In addition, the library contains most of the code of VSE/Advanced Functions, also in object-code format. Availability of that code at the VSE controlled installation helps to reduce the size of and handling effort for PTFs which may have to be applied to the system later on.

- The source statement library contains IBM-supplied macro definitions. You may assemble the macros that you require. For your convenience, the source statement library also contains sample programs and system generation job streams that can be retrieved as needed.

- The procedure library contains procedures (sets of control statements) for the following:

  - Linking system components to the core image library
  - Deleting IBM supplied library members from any of the three other types of libraries
  - Defining standard labels
  - Using the Maintain System History program (MSHP), an IBM supplied system installation and service aid
  - Performing an automated initial program load for the shipment supervisors

IBM supplies VSE/Advanced Functions on magnetic tape; this programming support must be copied onto disk at your installation before the system generation procedure can begin.

Additional nonlicensed or licensed IBM programming support available for use at a VSE controlled installation must be ordered separately. For information about this additional programming support, turn to Chapter 6 of this manual.

Installation of an Operational VSE System

IBM's installation and service tool for a VSE system, the Maintain System History program, is part of VSE/Advanced Functions. The program greatly facilitates the system generation process. It automates to a great extent the inclusion of required IBM supplied and user-written programming support in an installation's VSE system.

This ends the discussion of the system generation process. The next chapter, Chapter 4, highlights the advantages of VSE/Advanced Functions over the Release 34 SCP.

Chapter 4: Advantages of VSE/Advanced Functions

This chapter highlights programming support that was not available with the DOS/VS Release 34 SCP. It briefly discusses the newly available services and facilities grouped by functional areas such as: extended device support or performance improvements.

This chapter of the manual includes references to licensed IBM program products other than VSE/Advanced Functions. For more information about available licensed program products referred to in this chapter, see Chapter 6 of this manual.

Extended Hardware Support

central processors

On a 4300 series processor, you can, by generating your installation's supervisor for operation in ECPS:VSE mode, ensure full utilization of this support. If you prefer to operate this processor in System/370 mode, you have the option of generating a supervisor for operation in that mode.

On a processor of the System/370 product line (or an IBM 3031) you can operate only in System/370 mode, of course.

new device support

For a complete list of I/O devices supported by VSE/Advanced Functions, see "Appendix A: Device Support". Given below is a list of I/O devices newly supported by VSE/Advanced Functions:

- The IBM DASDs 3310 and 3370, which use the fixed-block-architecture (FBA) concept.

  On 4331 processors, stand-alone support is available for formatting specified areas on a volume of an FBA DASD for emulation of CKD extents as follows:

FBA device	Emulated CKD device
3310	2311, 2314/2319
3370	2311, 2314/2319 3330 Mod. 1, 3340

  For more information about the FBA support, see *VSE System Data Management Concepts.*

- The IBM 8809 – A magnetic tape unit for operation in two speeds: low speed (called start/stop mode) and high speed (called streaming mode). When operated in start/stop mode, the IBM 8809 is fully compatible (and interchangeable) with IBM's half-inch magnetic tape units.

- The IBM 5424 – A multifunction card unit whose functional capabilities are fully supported. Your coding requirements for utilizing the support are similar to those for utilizing the support for the 5425 multifunction card unit in the past.

- The IBM printers 3262 and 3289 Model 4.
- The IBM terminal (console) printers 3284, 3286, 3287, 3288, and 3289 (except Model 4).
- The IBM 3890 – A magnetic ink reader/sorter device (the logical IOCS support for this device is available as a licensed program product).
- The IBM 3278 Mod. 2A – A display console.

Performance Improvements

more partitions

VSE/Advanced Functions includes the support for multiprogramming in one background partition and up to eleven foreground partitions. This may be of particular advantage if your installation uses several continuously running programs such as IBM's licensed programs VSE/POWER and VSE/Interactive Computing and Control Facility (VSE/ICCF).

up to 208 subtasks

Up to 208 subtasks may be active within the system at any point in time, up to 31 subtasks may be active concurrently within a single partition. This extended support greatly facilitates the implementation of complex applications; it may well result in better utilization of processor time.

partition balancing

You may specify two or more partitions to participate in partition balancing. For partitions so specified, the system checks their processor usage in regular intervals; the system then reassigns the processing priorities of those partitions such that the partition with the highest record of processor usage will be last in line in competing for control over the processor.

pageable supervisor

Supervisor routines that have a relatively low usage rate and can tolerate a page-in delay when their services are required have been redesigned to be pageable.

logical transient area contention removal

Certain system service routines that are executed in the logical transient area under DOS/VS are executed from the shared virtual area in a VSE system. While this change does not affect the operational characteristics of the system, it improves system throughput.

fast transient-phase fetch

Frequently used B- or C-transients can be loaded into the SVA.

Whenever such a transient needs to be executed, it is moved from the SVA to the transient area in the supervisor rather than fetched from the core image library. If an installation's paging rate is not too high, the fast transient-phase fetch facility can save I/O completion wait time and free system resources such as channels and system tasks for use by programs executing in other partitions.

high-level SDL search

To avoid several, successive page faults during a search for a specific entry in the system directory list (SDL), VSE/Advanced Functions builds a high-level index of the SDL and locates the required SDL page through this index. This restricts the paging activity for an SDL search to one page-in operation (at most two if an SDL entry crosses page boundaries).

page data set on up to 15 extents

Defining the page data set in several extents allows you to take advantage of fixed-head areas on disk for performance critical applications in a certain partition. You may distribute your installation's page data set over up to 15 extents, up to three extents per volume.

fast ϘPEN of the hard-copy file

If the hard-copy support is enabled, processing of the first // JOB statement following the SET command is processed faster than by the DOS/VS Release 34 SCP. This processing includes the opening of the hard copy file.

fast CCW translation for I/O requests with IORB

The service is an extension of the fast CCW translate support which has existed for I/O requests with a CCB in the DOS/VS Release 34 SCP. This service is applicable to I/O requests that repeatedly use the same areas for I/O.

switchable fast CCW translation

Fast CCW translation may be turned off for individual jobs. This may be of significance for jobs that are unlikely to use the same I/O areas and I/O control blocks repeatedly. Jobs of this kind would be telecommunication applications or non-sequential update programs.

avoiding channel program scan

If, in a program, you start an I/O operation by issuing an EXCP macro that refers to a CCB (command control block), the system scans the associated channel program to determine which of the program's areas need be fixed in processor storage for the duration of the requested I/O operation. Instead of a CCB, you can now define (and refer to in the EXCP macro) an input/output request block (IORB) which passes to the system the beginning and end addresses of the areas that are to be fixed for that I/O operation. In that case, the system suppresses this scan of the associated channel program.

improved linkage editor

If modules stored in a relocatable library are to be included in the program that is being link-edited, the linkage editor of VSE/Advanced Functions performs faster than the linkage editor of the DOS/VS Release 34 SCP.

high performance in a virtual-machine environment

Your VSE system, in order to operate with high performance under VM/System Product (VM/SP) or VM/370, can be generated to include the VM/370 linkage facility. This facility maintains a high performance level by

- Avoiding many of the instructions that are redundant in this environment (it avoids functions such as load leveling and paging as well as page fixing and page freeing; it executes fewer privileged instructions).

- Returning control to VSE immediately after VM/SP or VM/370 has handled a VSE detected 'pseudo' page fault, thus enabling VSE to give control to another program or task that is being executed on the virtual VSE machine.

- Avoiding a channel-program check interrupt (PCI) linkage that was used previously by the DOS/VS BTAM routines for auto-polling. This linkage has been replaced by a direct linkage to VM/SP or VM/370.
- Automatically closing the printer and punch files that are spooled by VSE/POWER. These files are printed (punched) without a specific request.

Usability Improvements

automated system initialization

If you write appropriate procedures and catalog them in your system's procedure library, your operator can start up VSE with virtually no interaction on his/her part.

device independence

A program need no longer define the device type for a disk file if that program accesses this file by using one of the following access methods:

- Sequential access method (SAM) with the particular file defined by means of the DTFSD macro.
- Direct access method (DAM) with the particular file defined by means of the DTFDA macro.

If, for example, an existing assembler language program contains device type specifications and has to be reassembled to include additional functions, there is no need to remove those specifications; they will be ignored by the particular access method when the program is executed.

more programmer logical units

For each partition, you may define up to 255 programmer logical units (SYS000 to SYS254). This adds to the flexibility of your VSE system in the area of late binding of I/O devices.

dynamic tape drive assignment

A program written in assembler language can request the operating system to find a magnetic tape drive that is not tied to one of the system's partitions by a previous assignment of a logical unit and to make this drive available to the program. The support provides also for dynamic release of that drive when the program has no further use for it. This may be particularly useful in long running, complex applications that require a magnetic tape volume to be available only for a short period of time (a one time storage of intermediate processing results, for example).

loading the SVA

Phases required by the operating system to be resident in the shared virtual area (SVA) are loaded into that area automatically during IPL. Phases that you require to be SVA resident for improved performance of your own programs may be loaded any time after IPL between jobs or job steps. Once a phase has been loaded into the SVA, that phase remains available for execution from that area until next system start-up.

GETVIS space for programs running in real mode

Programs running in real mode may now request (by means of a GETVIS macro) to have processor storage allocated dynamically. This new service is available for programs (or routines of programs) written in assembler language. Your VSE system allocates storage so requested within the constraints of processor storage allocated to the particular partition.

job-to-job communication

The new JOBCOM macro provides for up to 256 bytes of information to be communicated from one job step to another or from one job to another in the same partition.

program input via job control

Via the EXEC statement, a string of up to 100 characters may be passed to the program that is being invoked. This may be a convenient way of controlling the program's control flow (if you pass control information), or of providing variable data, or of both.

improved label processing

Label information can now be added to or deleted from previously stored label information without a need for resubmitting the previously stored label information. This may well result in less job preparation effort.

Label information to be stored permanently for foreground partitions can now be loaded from the background partition. You might consider loading all of the permanent label information for your system's foreground partitions by executing just one cataloged procedure in the background partition.

larger standard label information area

The size of the standard label information area within the system residence (SYSRES) file has been increased by one cylinder, except for a SYSRES file on an IBM 3350.

The concept of allocating a fixed disk space to each of the partitions has been changed; the label processing routines now dynamically acquire the space they need for storing a partition's labels. This new concept improves exploitation of the available label information area.

label information area as a separate file

By defining a label information area outside the SYSRES file, you can have different processors share one SYSRES volume. You may define this separate label information area to reside on any supported IBM disk device except 2311.

Separate label information areas may be particularly useful at installations that use VSE on two or more processors or on several virtual machines under VM/System Product or VM/370.

A separate label information area may be useful also if the standard one within the SYSRES file is too small and there is no room for extension of that area within this file. In that case, however, the standard label information area will no longer be used by your VSE and, as a result, standard labels stored in that area must be resubmitted for storing them in the separate label information area.

A separate label information area can be defined (or redefined) during system start-up. Also during system start-up, you indicate to the operating system whether to use the standard or a separate label information area.

asynchronous operator communication

The operator need not reply to messages in the same sequence as they are displayed on the console. He can defer his reply to a message and either respond to other messages or enter a command. The operator can defer his reply to one message per active task in the system, regardless of the partition in which a task is being executed.

If the system requests a response to a previously displayed message and this message has not yet been rolled off the screen, then the system flags this message as not deletable. As a result, most of the messages that require an operator's response are retained on the screen until the operator has entered this response, and the need for the operator to have a message redisplayed will arise less frequently.

disk volume recognition

A command is now available for the operator to request a display of disk volume status information for one, a selected group, or all disk devices that are part of the installation. The display contains information on the specified volumes such as: in use or not in use, reserved, shared across systems.

early volume mount request

Support is available which allows VSE/VSAM to issue an early volume mount request, if necessary, and to reserve a free disk volume, if this is appropriate and one is available. The DASD status display provided by the system in response to the VOLUME command has been extended to show a drive's "reserved" status. An operator command is available to free a reserved drive.

improved assembler

The assembler now allows **long SETC** values: a character string of up to 255 characters may be assigned to a SETC symbol.

By specifying SXREF in the // OPTION statement for an assembly, you can **suppress the printing of unreferenced labels** in the label cross-reference list.

link and go

By specifying GO as an operand in the // EXEC statement for an assembly or a compilation of a source program, you can request the system to link-edit the program immediately after completion of the language translation and to have the program executed immediately after link-editing. There is no need for // EXEC statements to start the link and program execution steps.

files in VSE/VSAM managed data space

VSE/Advanced Functions includes support that simplifies considerably job control information needed to process files in VSE/VSAM managed data space. Only one statement – the // DLBL statement – is needed to relate a particular file to the program that is to process this file. The file may be a VSAM file, a SAM file on disk, or a linkage editor or assembler work file on disk; it may be the work file of an IBM compiler if that program includes the required interface code.

The // DLBL statement now includes operands that define the size of a file (in terms of average size and number of records) and its disposition at the end of the particular job (keep or delete).

backup/restore improvements

If the library file identifiers for the restored library are to be the same as for the original (backed up) library, these identifiers need not be redefined.

The restore program uses an improved sorting algorithm for faster sorting of the core image library directory.

The stand-alone restore program, if required, need no longer be punched into cards. The backup program now writes the stand-alone restore program onto the backup tape.

simplification of system generation

Supervisor assembly has been simplified by significantly reducing the number of supervisor generation parameters. IBM's development effort in this area started with Release 1 of of VSE/Advanced Functions.

In view of the available automated system initialization support, many of the supervisor functions previously available as generated options are now available as IPL options and may be specified in system start-up procedures cataloged in the system's procedure library. Device specifications previously made for system generation are also included in a start-up procedure. Many of the tables needed by your installation's supervisor to control hardware/software interaction are now created automatically during IPL rather than included in the supervisor during assembly.

As a result of these conceptual changes, supervisor assembly has been greatly facilitated: you need to specify no more than three macros with approximately 30 options, from which you select the ones that meet your installation's requirements; instead of having several supervisors stored online, your installation might now do with only one supervisor together with two or more sets of cataloged procedures for system start-up.

Availability Improvements

sharing of data on disk

Macros are available for efficient control of the sharing of data on disk by two or more tasks that are active in the same system or in separate systems that share this disk data. The types of data whose sharing can be controlled are: libraries, disk volumes, catalogs, and files. The macros may also be used to control shared access to particular control blocks in virtual storage.

No extra programming effort is required for the sharing of the system and private libraries of your VSE if these private libraries were created using the IBM supplied librarian. There is no need for inclusion of these macros in programs that access files in VSAM managed data space because VSE/VSAM routines provide for the required share control functions.

The advantage of this new support is obvious: if the same data must be online for concurrent execution of programs in two or more systems at the same installation, that data may be stored on disk only once. It follows that such data need be updated only once whenever an update is necessary; moreover, the disk space needed to store that data must be made available at the installation only once.

More information about the available resource share control support is given in Chapter 2 under "Sharing of Data Across Computing Systems".

chaining and sharing of libraries

Libraries of the same type (core image, or relocatable, for example) may be chained for each of the partitions of your VSE by defining appropriate search orders.

You can define a search order to be temporary (for the duration of a job) or permanent (until next system start-up). Such a search order may include the names of up to 15 different libraries of the same type, and it may or may not include the system library of the particular type. If a defined search order does not include the system library, that library takes the place of the last one in the chain. Libraries defined in a search order may reside on different type disk devices.

Since the operating system's librarian programs make use of the available resource share control facilities, the same library may be defined in the search order chains of different partitions; if a library resides on a disk device for which support for the sharing of data across systems is available, that library may be defined in search order chains of partitions of systems that share the particular disk volume.

Chaining and sharing of libraries provides for more flexibility in the area of job scheduling and system management: less frequent mounting and remounting of private libraries; less frequently a need for assignment of private libraries, virtually only once per shift – immediately after system start-up; increased system availability because only one set of program or procedure data need be online and serviced instead of one set for each of the sharing systems.

channel switching for disk devices

The channel switching support available for magnetic tape units in the past has been extended to be available also for disk devices attached to a processor via two channels. The system automatically switches to the second channel should the first one be busy.

installation/service tool

The Maintain System History program (MSHP) has been included in VSE/Advanced Functions to facilitate the process of installing, servicing, and upgrading an installation's VSE system.

By maintaining a history of the installed system components and the IBM supplied program temporary fixes (PTFs) applied to these components, the program is capable of

- Automating the process of prerequisite checking for the application of cross-component program temporary fixes.
- Indicating whether a library member (for example, an installations's supervisor) needs to be reassembled and relinked for the application of a fix.

A preprocessing function of MSHP provides, on request, a printout of the handling information for an IBM service tape (also referred to as PUT tape) or a printout of cover letters for individual PTFs or both. The function can be used to selectively apply those PTFs of a PUT tape that are specified by the user.

removal of the job-information-block (JIB) table

Previous JIB limitations have been removed by the deletion of the JIB table. If assignments of logical units to actual I/O devices need be saved, the system writes the required information into an extendable table.

fewer IPL situations

If a device-not-ready condition occurs for the SYSRES device, the page data set, or a private core image library, there is no need for an initial program load operation in order to recover from this condition.

improved handling of malfunctions

A new way of handling I/O errors helps avoid cancel situations. Soft waits caused by missing device interrupts are detected and corrective action is initiated. A soft-wait state can no longer go undetected for an extended period of time.

improved partition space allocation algorithm

Allocation of partition space can be requested explicitly for the program that is about to be executed. This avoids a job cancel situation which occurs under DOS/VS and Releases 1 and 2 of VSE/Advanced Functions if a phase larger than the program being invoked is cataloged by the same generic name in the same library or chain of libraries.

Serviceability Improvements

improved storage dumps

The system's dump routines now execute from the SVA rather than from the logical transient area, an area of the supervisor.

If a dump is taken as a result of a job-canceled situation, the system's dump routine is executed before any resources are freed by automatic system action and before an abnormal-end user exit receives control, if there is such an exit.

VSE/Advanced Functions includes a new stand-alone dump generation and print program. The program is capable of:

- Generating a stand-alone dump program for IPL from cards, magnetic tape, disk, or diskette.
- Printing all or selected parts of the output of a stand-alone dump program execution or of a DUMP command execution.
- Printing the output of the system's tracing program, SDAID.

You request execution of the various functions of the program via the operator's console. To facilitate this operator-program interaction, the program displays menus and, on request, explanations for the use of the program.

If a stand-alone dump is taken, the dump includes the last 200 console communication lines.

alternate dump files

VSE/Advanced Functions supports fast writing of dump information onto a disk volume to which a new system logical unit, SYSDMP, is assigned. On this volume, you can create two dump files. When one of the files is full, the system's dump facilities write into the other one.

Writing dumps into a dump file on SYSDMP is faster then writing them into the disk dump file that includes the stand-alone dump program (should this have been generated on disk).

This alternate dump file support is available for system dumps and dumps requested by the DUMP command; it is a prerequisite for the use of VSE/Interactive Problem Control System (VSE/IPCS), a licensed IBM program designed to assist in problem determination.

phase load trace tables

For each partition in virtual storage, the supervisor maintains one wraparound table. The table contains one entry per phase load request from the associated partition, and each entry indicates the name of the particular phase, the phase's load address, and its length. In each table, the supervisor records 15 load requests before a wraparound occurs.

extended tracing facilities

The tracing facilities available with VSE/Advanced Functions are all consolidated in one program, SDAID. Its most significant extensions over the tracing facility of the past are:

- Up to eight different traces may be requested for one execution of the program.
- Trace information is provided formatted for ease of evaluation of the recorded events.
- A variety of additional output options per traced event may be requested; examples are: the contents of registers, the contents of certain control blocks, the contents of specific areas.
- Unrestricted tracing of buffer overflow without loss of any event records.
- Tracing of an event can be limited to a specific number of occurrences or to a specific area (a partition, for example).
- The range of a trace may be defined symbolically.

You invoke the program by a 1-word attention routine command; the program then prompts you for the required control information.

console communication on SYSLST

The available list log utility program is capable of producing a list of all information which the system collected for a particular job in its hard copy file. The program writes this list to the device assigned to SYSLST.

The program may be invoked implicitly by the operating system's dump routines for inclusion of job-related console communication in the dump output. Console communication so included can be retrieved for problem evaluation by VSE/IPCS.

EREP common for VSE, VM/SP, and OS/VS

The Environmental Recording Editing and Printing program shipped with VSE/Advanced Functions is the same as the version of this program available at OS/VS installations. This is useful, for example, if your installation operates computing systems under OS/VS and VSE in parallel.

improved OLTEP

The Online Test Executive program can be executed in any partition.

Data Protection Improvement

protection by user-ID and password

If you installed VSE/ICCF, your VSE system provides for the verification of user identification and access authorization also for jobs submitted in partitions of your VSE. Execution of a job stream is rejected if this job stream contains an invalid user identification or password or if the system detects an attempt to access protected data without authorization.

access logging

If your installation's VSE includes, in addition, IBM's licensed Access Control–Logging and Reporting program, you get a listing of attempted unauthorized accesses and, optionally, of any accesses to protected data.

Existing job streams may be submitted unchanged as long as they do not access protected data.

Chapter 5: Program Compatibility

This chapter discusses program compatibility for distinct groups of migrating users as follows: from a previous release of VSE/Advanced Functions, from DOS/VS, from DOS. The chapter also discusses compatibility considerations for users of VSE/Advanced Functions (DOS/VS or DOS) under VM/370 or VM/System Product.

Migrators from a Previous Release of VSE/Advanced Functions

VSE/Advanced Functions Release 2: An application program that has been executed successfully in the specified operating environment of VSE/Advanced Functions Release 2 will be executed successfully also in the specified operating environment of Release 3 if this application program interfaces with the operating system through IBM provided macros or through high-level language statements of supported compilers.

VSE/Advanced Functions Release 1: An application program that has been executed successfully under the DOS/VSE SCP with VSE/Advanced Functions Release 1 will be executed successfully also in the specified operating environment of Release 3 if this application program

- interfaces with the operating system through IBM provided macros or through high-level language statements of supported compilers.
- does not alter its DASD SAM or DAM DTF blocks.

A definition of the order of processing priority for partitions via the PRTY parameter at the time of system generation is no longer valid; this processing priority may be defined during partition start-up within an automated system initialization procedure.

FORTRAN programs must be relinked with DOS FORTRAN Library Option 1 Release 3 installed.

Migrators from DOS/VS

Incompatibilities and compatibility prerequisites indicated above for users migrating from a previous release of VSE/Advanced Functions apply also to users migrating from DOS/VS. Moreover, for a program that has been executed successfully under DOS/VS to be executed successfully also under VSE/Advanced Functions, you should ensure that the program:

- has not been linked to absolute addresses – if it has been, the program must be relinked to obtain a relocatable version of the program.
- accesses supported I/O devices only.

- does not contain IDA lists if your VSE system is to operate in ECPS:VSE mode – if it does, the program must be changed to replace the pertinent CCWs.

- does not access the interval timer at location 80 or the system time of day (SYSTOD) at location 84 – if it does, the program must be changed to use the GETIME macro instead.

- does not access libraries by user-written library access routines – if it does, the library (or libraries) to be accessed must be retained on the previously used library device type.

- does not include BTAM modules – if it does, the program must be relinked with a new BTAM-ES module that has been assembled using the current BTAM-ES BTMOD.

- does not interface with QTAM (queued telecommunication access method) – if it does, the program must be changed to use BTAM-ES or ACF/VTAM (ACF/VTAME) instead.

- does not include, for a DTFSD or DASD-DTFDI file, an error exit routine which issues one or more imperative macros (such as GET or PUT) for that file – if it does, the program must be changed to avoid imperative macros in that exit routine.

- does not make use of the following DOS/VS components:

  - System/360 Model 20 emulator on System/370
  - deblock system utility

  These programs are not part of VSE/Advanced Functions.

- does not process data stored on mixed-parity tapes.

Migrators from DOS

Current DOS users who plan to change to VSE will have to consider the following in addition to the applicable items given for migrators from DOS/VS.

- Current DOS data files can be processed under a VSE system if compatible I/O devices are used:

  Programs written to process data for a 2311 can be executed to process the same type of data through the use of the available 2311 compatibility feature for a

  - 3330/3333 or a 3340 attached to a System/370 Model 125.
  - 3340 attached to a System/370 Model 115 or 135.
  - 3310 or 3370 attached to a processor with ECPS:VSE.

  Programs written for the 1052 Console Printer-Keyboard can be processed on the video display and 5213 Console Printer of System/370 Models 115 and 125 through the use of the available 1052 compatibility feature.

- Existing assembler language source programs can be assembled by the available assembler, provided that no user written macros are called. If such macros are called, the user must either supply COPY instructions for the macro definitions at the beginning of all source decks in which the macros are used, or convert his library macros to edited macros and include them in the macro sublibrary.

- Existing high level language source programs can be compiled if the appropriate compiler is available for use under VSE. COBOL D programs must be changed or converted with the Language Conversion Program before they can be processed by the DOS/VS COBOL compiler. RPG programs must be adapted to RPG II.

- Previously compiled or assembled DOS object programs can be link-edited without modification under VSE. User programs will be executed provided the following points have been taken into account:

  - Devices specified by the program must be available on the system on which VSE/Advanced Functions will run.
  - Programs that depend on central processor circuitry not supported on System/370 or on processors with ECPS:VSE may not execute properly.
  - Proper processing of time dependent programs that run under DOS is not ensured. IBM recommends to have these programs executed in real mode (without paging).
  - Programs that deliberately create program checks may not run properly.

- Supervisor sizes will be definitely larger than under DOS. Therefore, programs will have to be relink-edited if they were not written to be self-relocating.

- User written I/O appendage routines must either run in real mode or adhere to certain restrictions which are described in *VSE/Advanced Functions System Management Guide.*

- Programs with self-modifying channel programs must run in real mode.

Users in a Virtual Machine Environment

The operating characteristics of a VSE system under VM/370 or VM/System Product (VM/SP) are, in principle, the same as those of a VSE system that operates on a real machine. Therefore, the compatibility considerations which apply to migrators in a real-machine environment apply also to migrators on a virtual machine. However, the following should be considered:

- Programs that function correctly under VM/SP or VM/370 without the VM/370 linkage facility available may not function correctly if that facility is available. For example, if more than 255 PFIX macros are issued for a particular page without a PFREE for the page in between, an error occurs if the program is executed on a virtual machine under a VSE with the linkage facility included in the supervisor.

- Programs that have been executed successfully under VSE on a real machine (or on a virtual machine without the VM/370 linkage facility) can be executed successfully also under VSE with the VM/370 linkage facility on a virtual machine; the reverse is not necessarily true.

- If VSE has been generated to operate in ECPS:VSE mode, the VSE generated stand-alone dump program cannot be used; the CP dump can be used instead.

Chapter 6: Additional Licensed and Nonlicensed Programs

IBM offers a variety of programs for inclusion in an installation's VSE system. This section gives an overview of such programs, limiting the discussions to general purpose type programs. For a complete list of programming support available from IBM for inclusion in an installation's VSE, contact the IBM branch office serving your locality.

Additional programming support available from IBM for inclusion in an installation's VSE system can be categorized as follows:

- Service programs (or subsystems)
- Language translators
- Application programs (or systems)

The last category will not be dealt with in this chapter.

Any licensed IBM program is available only through a license agreement between IBM and the user who intends to utilize such a program. For more information on license agreements for programming support, contact the IBM branch office serving your locality.

Service Programs

VSE/Virtual Storage Access Method (VSE/VSAM)

VSE/VSAM is an efficient access method for direct and sequential processing of fixed and variable-length (including spanned) records on direct-access devices. VSE/VSAM offers, on a high level of performance, more functions than the access methods available with VSE/Advanced Functions; it provides for a more flexible data organization.

file organization

The records in a VSAM file can be organized in logical sequence by a key field (key-sequenced file), in the physical sequence in which they are written into the file (entry-sequenced file), or according to the relative record numbers in the file (relative record file). The user can read, add, delete, and modify records in a VSAM file.

access of a file

VSE/VSAM allows for retrieval, storage, update, and deletion of records. Such access can be by record key, by record address, or by relative record number. In case of an access by record key, the key can be that of an individual record or it can be a generic key specifying a group of individual records.

Blocking and deblocking of records is done by the access method, which optimizes block length to suit the device on which the file is written.

With a key-sequenced file, several records in sequence can be inserted as a group at one point in the file (this is faster than inserting them one at a time as ISAM requires). Also, VSE/VSAM can access several records in key-sequence and then skip to another portion of the file and access more records in sequence without having to search the entire index to find the new group of records (this is called skip-sequential access). For keyed or addressed sequential (but not skip-sequential) processing, there is also an option to process records backwards. Key-sequenced and entry-sequenced files may be accessed via alternate indexes.

alternate indexes

VSAM catalogs

VSE/VSAM keeps control over the creation, access, and deletion of files and over the direct-access storage space allocated to those files. This is done by keeping information on file and space characteristics in a VSAM catalog. VSE/VSAM is capable of keeping control not only over VSAM files but also over SAM disk files (see also "space management for SAM files" below).

There are two kinds of VSAM catalogs, master and user catalogs. One master catalog is required; any number of user catalogs are optional. User catalogs increase data integrity and facilitate volume portability.

data protection

A file under control of VSE/VSAM can be protected against unauthorized use through passwords. The various password levels available grant authority to read a file, to read and update a file, or to read and update both a file and the VSAM catalog.

data integrity

Data integrity is improved by minimizing data movement and index updating when records are added, by preserving both new and old index paths to data until an update is completed, by special formatting to indicate the end of a file as it is being created or extended, and by a provision for recovery from damage to the catalog.

file sharing

VSE/VSAM allows files to be shared across partitions or across systems. When a record is updated, the control area containing the record is protected under exclusive control. The remainder of the file may be accessed by programs running in other partitions or systems.

space management for SAM files

Additional enhancements for SAM files in VSAM managed data space are available as a separate feature. These enhancements are:

- Definition and deletion of a SAM file in VSAM data space either explicitly by an access method services DEFINE or DELETE command or implicitly by the execution of an OPEN or CLOSE for the file. A SAM file in VSAM data space (hereafter called SAM ESDS) may be accessed either
  - via a DTFSD macro or a VSAM ACB if VSE/VSAM has created the file using the VSAM CI (control interval) format, or
  - via a DTFPH macro with MOUNTED=SINGLE specified, in which case the file may have been created by VSE/VSAM in CI or non-CI format.
- Dynamic secondary allocation for a SAM ESDS during creation or extension of the file according to the secondary allocation size that was defined for the file.

backup/restore feature

If the backup/restore feature is installed, backup of VSAM files onto magnetic tape and restoring these files to disk is accomplished faster. The feature supports also writing onto and reading from magnetic tape on an IBM 8809 in streaming mode.

language support and implementation

Support of VSE/VSAM is provided through macros in the assembler language and through high-level languages such as DOS/VS COBOL, PL/I Optimizer, and DOS/VS RPG II.

For more information about VSE/VSAM see *VSE/VSAM General Information,* GC24-5143.

VSE/POWER

VSE/POWER, a licensed program, provides for improvement of system performance. It is designed to reduce central processor dependence on the relatively slow speeds of unit-record devices.

VSE/POWER decreases the execution time of unit record I/O-bound jobs by servicing I/O requests addressed to such devices at disk I/O speed; it reads and punches cards or prints reports or accomplishes both in parallel during the execution of other jobs. The additional processor time used by VSE/POWER is negligible. In a typical environment of jobs with mixed characteristics, throughput may be improved substantially.

VSE/POWER requires one of the generated partitions and allows you to execute programs in the remaining ones without the need for separate unit-record devices for each of these partitions. The unit-record devices used by VSE/POWER can handle the unit-record I/O activities for all the partitions that are being serviced by VSE/POWER.

Processing with VSE/POWER is a follows (see Figure 6-1):

- **Input.** VSE/POWER reads the job streams (job control statements, programs, and data cards) for the individual partitions and stores these in input queues on disk.

- **Execution.** From disk, the jobs are transferred by VSE/POWER to the designated partitions and executed.

- **Output.** Unit-record output (printer and punch) of every job is stored on disk (or tape) by VSE/POWER before it is finally processed as output.

Job input as well as job output may be held in the queues for execution or printing/punching at a later time. This allows you to hold jobs that need, for example, two hours of execution or printing time until the system is less occupied.

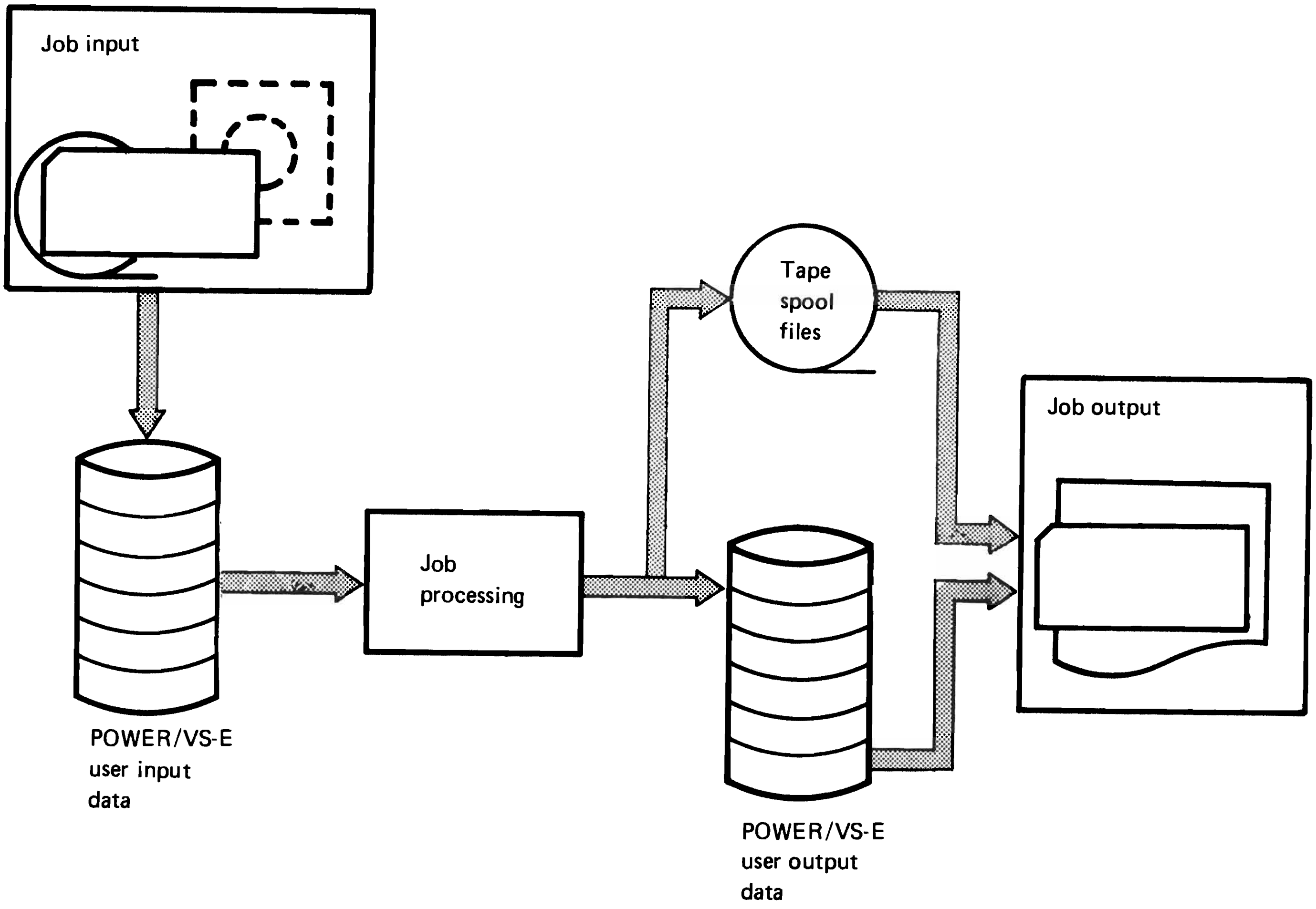


Figure 6-1. Processing with VSE/POWER

shared spooling feature

With the separately available shared spooling feature installed, and if all requirements for a shared DASD environment are met, VSE/POWER can share the POWER files (queue, data, and account) with a VSE/POWER operating on another processor. The feature provides the same sharing capability for virtual machines under VM/SP or VM/370.

remote job entry feature

VSE/POWER offers a telecommunication facility for remote job entry. With this facility (VSE POWER RJE), which is available as a separate feature, jobs may be submitted from remote terminals in very much the same way as if they were submitted from a unit-record device at the central location. Output of remotely submitted jobs may be routed to a local output unit at the central station or to a remote terminal.

job accounting

VSE/POWER also provides, optionally, an accounting facility. If you use this facility, you need not write your own data collection routine; instead you process the account file, sorting and summarizing the records to suit your particular accounting requirements.

For more information about VSE/POWER, see *VSE/POWER General Information Manual,* GH12-5131.

VSE/Operator Communication Control Facility (VSE/OCCF)

VSE/OCCF is designed to reduce the number of messages that are displayed by a VSE system at the operator's console and thus to help controlling the operation of a VSE system more efficiently. The program allows the operator at the system console to concentrate on activities which do require operator attention, activities such as monitoring job throughput and mounting volumes of data needed for program execution. VSE/OCCF provides functions that allow the operation of a VSE system to be controlled from a remote operator station.

The program's functions are:

- Suppressing the display of messages

  You can, for example, request VSE/OCCF to suppress the display of messages which a VSE system writes to the console only for information. Messages whose display is so suppressed are nevertheless written into the system's hard-copy file.

- Replying to messages automatically

  You can, for example, define replies with which an operator would have to respond to certain messages according to your installation's procedures. When it finds a reply for a message, VSE/OCCF automatically picks up and presents this reply to the program that originated the message.

- Routing of messages

  You can request VSE/OCCF to route messages to the system's console or to an operator station from which you want to control the system operation. This operator station may be located remotely from the system. If VSE/OCCF routes messages to a local or remote operator station, the program accepts replies to routed messages as well as commands from that station.

  Via an option of the program, you can request an audible alarm to be sounded at the system console whenever a message is routed to it by VSE/OCCF.

- Translating message texts, replies, and commands

  The program allows you to define translations of or suitable substitutions for message texts, replies to messages, and commands. VSE/OCCF automatically uses the defined translations or substitutions (for example, the code NV for NEWVOL) for the text presented or expected by the system.

For more information about the program, refer to *VSE/Operator Communication Control Facility, General Information,* GC33-6113.

VSE/Interactive Computing and Control Facility (VSE/ICCF)

VSE/ICCF is an adaptation of the widely used DOS/VS Entry Time Sharing System, which has been available as a field developed program; it includes significant functional enhancements.

By including VSE/ICCF in your installation's VSE, you change the installation's basic operating characteristic from batch to interactive. Through VSE/ICCF, all of your installation's computing services are available, on a time-sliced basis, to the authorized users of terminals that are linked to your installation's central processor.

VSE/ICCF, which is available as a licensed program product, includes efficient editing and command-processing facilities that allow a terminal user to:

- build job streams and prepare input data for immediate or later processing.
- store such job streams or data (or both) in a library.
- change (edit) previously stored job streams and data.
- request the execution of a job stream, monitor its processing status, and cancel execution of the job stream if this is appropriate.
- have output of processing requested by him directed to his own terminal, to an output device at the central location, or to another terminal.

VSE/ICCF includes procedures and utilities which greatly facilitate utilization and control of your installation's computing services from a terminal.

VSE/ICCF provides for a high level of data integrity and data protection at a VSE operated installation. Via a user profile, VSE/ICCF allows you to control the particular user's access to members of the VSE libraries (read-only, update, or not at all). A user's profile, and thus that user's ID and password, determines how other users may access VSE or VSE/ICCF library members created by that particular user. In addition, a terminal user can protect individual files or job streams by a seperate (member) password.

For more information about VSE/ICCF, refer to *VSE/Interactive Computing and Control Facility, General Information,* GC33-6066.

Advanced Communications Function for VTAM Entry (ACF/VTAME)

This licensed program product has been designed for use with VSE/Advanced Functions on an IBM 4331 processor. ACF/VTAME allows for building and operating a variety of data communication application systems.

ACF/VTAME, which can be used in a single-system as well as in a multiple-system environment, provides numerous communication functions and services. For a complete overview of ACF/VTAME support, see the publications

> *Advanced Communications Function for VTAM Entry (ACF/VTAME) General Information:*
>
> *Introduction,* GC27-0438
> *Concepts,* GC27-0451

Some of the functions and services available with ACF/VTAME are highlighted below:

- Controlling sessions and data flow between application programs and terminals. A single request for a session or for data can be directed simultaneously to more than one terminal.
- Direct transmission of messages between application programs and terminals. It makes the communication lines and the Communications Adapter transparent to the application program.
- The sharing of network resources among the various applications, which results in a more efficient use of lines, terminals, and the Communications Adapter.
- Capability of being connected to other processors that have installed one of the following IBM program products:
  - ACF/VTAME
  - ACF/VTAM with the Multisystem Networking Feature
  - ACF/TCAM with the Multisystem Networking Feature

  This allows application programs and terminals that are controlled by ACF/VTAME to communicate with application programs and terminals in another domain of the particular multiple system network.
- An installation's network operator can monitor and control the data communications network through the IBM Network Operation Support Program (NOSP) and Network Communication Control Facility (NCCF) program products.

ACF/VTAME provides, for users of a VSE system on an IBM 4331, many potential advantages:

- More flexibility in application processing and in accessing information between multiple system networks and the associated operating systems.
- Expansion of the scope of network facilities (such as application programs and terminals) that are available for communication of data and for processing.
- Under ACF/VTAME, in a multiple system network, continued operation of terminals (or devices) independently of a specific central processor.
- A wide range of multiple system configurations that support orderly applications growth under the system network architecture (SNA).

- Elimination of redundant network applications.
- Consolidation of communications management functions (such as terminal ownership control, session establishment).

Advanced Communications Function for VTAM (ACF/VTAM)

For telecommunication applications under a VSE system on an IBM processor other than 4331, functional support is available with ACF/VTAM. For more information about ACF/VTAM and its usefulness at a VSE controlled installation, consult these publications:

> *Advanced Communications Function for VTAM (ACF/VTAM) General Information:*
>
> > *Introduction,* GC27-0462
> > *Concepts,* GC27-0463

Introduction describes the product in a general manner, names the programs and devices that can be used with it, and lists the steps that must be accomplished to install the product.

Concepts contains extensive descriptions of the ACF/VTAM facilities and describes the installation process in more detail. Both publications describe the entire ACF/VTAM documentation library.

Basic Telecommunications Access Method – Extended Support (BTAM-ES)

This licensed program product facilitates the coding of telecommunication applications in assembler language; it is the prerequisite support for a variety of IBM supplied licensed and nonlicensed programs.

Through BTAM-ES macros, this support dynamically builds the channel programs needed for the execution of I/O operations involving the telecommunication devices that are hooked up with your installation's central processor. BTAM-ES provides a convenient method of coding efficient telecommunication routines in a problem program; however, as opposed to ACF/VTAME, BTAM-ES requires the programmer to be familiar with the installation's line configuration. Moreover, a problem program using BTAM-ES may have to be adapted to an installation's line configuration should that configuration change.

BTAM-ES is discussed in more detail in *BTAM-ES General Information,* GC38-0292.

Data Language/I (DL/I)

DL/I, a licensed data management control system, executes as an application program under a VSE system. DL/I satisfies many diverse data processing requirements. It simplifies the task of creating and servicing large common data bases that are accessed by user-written application programs. DL/I allows growth from a batch-only processing environment to a telecommunication environment such as CICS/VS.

For more information about the capabilities and advantages of DL/I, consult the publication *Data Language/I Disk Operating System/Virtual Storage (DL/I DOS/VS) General Information,* GH20-1246.

Customer Information Control System/VS (CICS/VS)

CICS/VS, a general purpose data base/data communication (DB/DC) control system, controls online DB/DC applications. CICS/VS, a licensed program product, includes

- most of the standard functions needed by application programs for communicating with remote and local terminals and subsystems.
- controls for concurrently executing application programs that serve many online users.
- data base access capabilities.

Furthermore, data used by online applications may, in general, also be used by non-interactive programs that are being executed in other partitions.

CICS/VS can be tailored to the needs of most combinations of concurrent online applications, serving a network that consists of a wide variety of terminals and subsystems.

For further information about this licensed program, consult the publication *Customer Information Control System/Virtual Storage (CICS/VS) General Information.* GC33-0066.

VSE/Access Control – Logging and Reporting Program

This program, together with VSE/ICCF, implements, for your VSE system, a high level of data protection.

If an attempt is made to access, without authorization, a data element that has been defined to be protected, the program records the event on disk. Optionally, the program records in addition each authorized access of such a data element. The information logged by this program for a recorded event consists of

– identification of the involved data element
– identification of the user accessing (or attempting to access) the data element

The program can be invoked explicitly to provide a report. For this purpose, a variety of options are available to control the scope and level of detail of that report. You can, for example, request the printout of all recorded events or only a summary of these events, or you can request a printout of events by specified users or by specified data elements.

Other major functions of the program are: dumping the recorded events onto magnetic tape and restoring the dumped information to disk; reset and clear the file(s) of recorded events.

A data element in this context is a private library, a member of a library, or a file.

For more information about the program, refer to *VSE/Access Control – Logging and Reporting Program General Information,* GH12-5130.

Subsystem Support Services (SSS)

SSS, a nonlicensed support package, is an installation and service program for industry subsystems such as:

- IBM 3600 Finance Communication System
- IBM 3660 Supermarket System.

Each of the supported subsystems has a control and data collection unit, called a subsystem controller, that contains control information necessary for the operation of all terminals and components attached to it.

SSS, which is available as a separately shipped component, can be used to place this control information on the disk file that resides in the subsystem controller; the program provides means for servicing that data. SSS accomplishes this by maintaining, at the host processor, a library of control information required to operate the industry subsystem. This library contains user-coded application programs and user-defined control records that define the operational environment for each subsystem controller.

An operational environment is based on the configuration of all devices within the subsystem, and on the various industry-related or application-related options that are available with each industry system.

By maintaining a central library at the host processor, SSS provides the capability for installing and updating several subsystems through a single control facility. This capability is in the form of control statements, which may be used to create the subsystem library, make modifications to it, and transmit selected portions of its contents to the subsystem controllers. In addition, SSS control statements may be used to obtain printouts of selected portions of the subsystem library, for use in monitoring subsystem activity.

The facilities of SSS, the industry subsystem supported by SSS, and instructions how to install the component are documented in *IBM System/370 Subsystem Support Services User's Guide,* GC30-3022.

DOS/VS Sort/Merge

This licensed program product enables you to sort multiple files of logical records into a predetermined sequence, or to merge files of previously sequenced records.

Besides giving improved performance in virtual mode over DOS Sort/Merge, this program offers a number of additional functions. These include:

- Utilization of any of the supported magnetic tape and disk devices for input, output, and work files.
- Support of SAM files on magnetic tape or disk and VSAM files on disk for both input and output.
- Support of a variety of user exits that allow the execution of user-written routines under specific conditions during a sort/merge run.
- Ability to execute an analysis run which ends after the program has finished analysing the input control stream and issuing appropriate messages.
- New control statements for
  - specifying a selection of records to be included in the sort/merge.
  - specifying reformatting of records.
  - requesting a summary of records.
  - specifying a user-defined collating sequence.

The DOS/VS Sort/Merge program can be invoked as a problem program executing under DOS/VSE or through a call from within a program written in assembler language, COBOL, PL/I, or RPG II.

For more details on this program and the manuals available for it, consult *DOS/VS Sort/Merge General Information,* GC33-4030.

VSE/Fast Copy Data Set Program

This licensed utility program is designed for fast copy data operations from disk to disk and for dump/restore operations via an intermediate dump file on magnetic tape or on disk. Specifically, the program provides functions as follows:

- Copy data – to copy from one disk volume onto a disk volume of the same type as follows:
  - all data that is referenced in the volume's table of contents.
  - all of that data except one or more files.
  - a single file, with relocation on the target volume if this is a SAM file.
- Dump data (for later restore) – to retrieve data stored on a disk volume and write it onto magnetic tape for later restore on a disk volume of the same type as the source volume. The subfunctions available for a copy operation are also available for the program's dump operation.

- Restore dumped data – to write data previously dumped back onto disk. The subfunctions available for a copy operation are also available for the program's restore operation.

The program can well contribute to overall system availability due to its selective copy/dump/restore capability, particularly so in a non-removable disk environment. A disk volume which, for example, is known to contain a number of expired files, need not be copied in its entirety; if only one or two files of a volume have to be restored (from a backup tape), the program can do this without impacting the integrity of the other files on the restore volume.

For more information on this program, see *VSE/Fast Copy Data Set Program, General Information,* GC33-6081.

VSE/IBM System/3 – 3340 Data Import

This licensed IBM program is available for use on a 4331 processor that is equipped with the System/3–3340 Compatibility feature. The program reads files written into the main data area of a 3348 data module by a System/3 Model 12 or 15 and converts these files to VSE files. You may choose to have the converted files written onto a data module of the same or other type or onto disk on an IBM 3310.

Any number of System/3 files may be converted during one run of the program, but each file requires its own set of utility control statements. If files in a 5444 simulation area of a 3348 data module are to be converted, those files must be copied into the data module's main data area prior to the conversion run. You can use the System/3 $COPY utility program for that purpose.

IBM Systems 1401/1440/1460 Emulator Program

This emulator, a licensed program product, allows programs written for execution on a 1401 (or a 1440 or 1460) to be executed on a 4300 series processor. This emulator may be operated under DOS Release 26, under DOS/VS Release 34, or under VSE/Advanced Functions.

A program running under this emulator can be executed in any of your installation's partitions without affecting the processing in the other partitions. More than one program, but only one per partition, may be emulated at any one time.

If an emulated program accesses data stored on disk or magnetic tape, the following requirements must be met:

- Disk files must be converted to CKD format, unless the CS format option (for CS30 or CS40) has been selected for emulator generation.

- Files on 7-track tapes can be processed only with the data converter feature turned on. Data on mixed-density or mixed-parity tapes cannot be processed.

1401/1440/1460 DOS/VS Emulator on System/370

This emulator, a nonlicensed support package, combines with the 1401/1440/1460 compatibility feature to allow programs written for execution on a 1401 (or a 1440 or 1460) to be executed on a System/370 under control of VSE/Advanced Functions.

A program running under this emulator can be executed in any of your installation's partitions without affecting the processing in the other partitions. More than one program, but only one per partition, may be emulated at any one time.

If an emulated program accesses data stored on disk or magnetic tape, the following requirements must be met:

- Disk files must be converted to CKD format before they are used by the emulator, unless the CS30 or CS40 compatibility option has been selected.
- Tape programs may be in original or converted format. The DOS spanned variable record (VRE) format is set as standard for the System/370 emulators.
- Data on mixed-parity tapes cannot be processed.

For more information about this emulator, see the publication *1401/1440/1460 DOS/VS Emulator on System/370,* GC33-5384.

VSE/Interactive Problem Control System (VSE/IPCS)

VSE/IPCS is an efficient programming tool designed to assist in software problem determination and in submitting operating system problems to IBM; it is available as a licensed IBM program.

VSE/IPCS requires that system dumps, dumps that result from operator issued DUMP commands, and the output of the system stand-alone dump program are written into an extent to which the system logical unit SYSDMP is assigned. These dumps are the source for the problem-determination assist functions of VSE/IPCS.

VSE/IPCS is capable of:

- Assisting in problem analysis – you can request VSE/ICPS to selectively display or print the contents of
  - specific control blocks of VSE/Advanced Functions or VSE/POWER at the time these blocks were dumped by a preceding dump operation.
  - certain areas of virtual storage by specifying beginning and end addresses.

- Assisting in APAR (authorized program analysis report) preparation by
  - providing information needed for the generation of a symptom string (which may be used as a search argument for an inquiry of IBM's known problems data base).
  - collecting information for a standardized problem description report as is normally required for an APAR.
- Writing (offloading) problem information from the SYSDMP extent onto magnetic tape if, for any reason, problem determination must be postponed.

  Later, when problem determination is to be performed, VSE/IPCS can be used to load this offloaded problem information for processing by VSE/IPCS.

VSE/IPCS can be executed either in a partition defined to VSE/Advanced Functions or in an interactive partition of VSE/ICCF. To control the operation of VSE/IPCS, commands can be entered from the system input device, from the operator's console, or from a VSE/ICCF controlled terminal.

More information about the functions available with VSE/IPCS is given in *VSE/Interactive Problem Control System, General Information,* GC34-2017.

VSE/Data Interfile Transfer, Testing and Operations Utility (VSE/DITTO)

This program provides file to file services for card I/O, magnetic tape, and disk devices. VSE/DITTO, a licensed program product, allows for files or portions of files to be created, listed, copied, and altered. The program's operational flexibility makes it a useful tool in a program testing environment; it reduces the need for separate special purpose utility programs and contributes to greater operational productivity.

For more details on this program and its documentation, refer to *VSE/Data Interfile Transfer, Testing and Operations Utility General Information,* GH19-6072.

Language Translators

DOS/VS RPG II Compiler

The DOS/VS RPG II compiler, a licensed program product, offers significant enhancements over the previously available DOS RPG and DOS RPG II compilers. Some of the enhancements are highlighted below:

- Performance improvements over the DOS RPG compiler in two areas:
  - Storage efficiency for object programs.
  - Throughput for processor bound programs.

- Data communication (DC) support through an interface to CICS/VS. Application programs written in RPG II may now run as transactions under CICS/VS.

- Data base (DB) support through an interface to DL/I DOS/VS in a batch environment and also under CICS/VS.

- Online source entry. Under a VSE system with VSE/ICCF or VM/CMS, source input may be entered into or changed in a work space via a display terminal in one of three modes: edit, input/alter, and prompt.

  In input/alter mode, the online source-entry facility displays a template of the RPG II specification that is about to be entered; it performs a syntax check of the entered values. In prompt mode, the facility prompts a user, field by field of the particular RPG II specification, for the applicable values.

  When finished, the source entry is filed in one of the available libraries for a subsequent compilation.

  Under VM/CMS, the compiler provides for a CALL interface to that system's display management system (DMS/CMS) or to its interactive file sharing (VM/IFS) support.

- Support of a defined set of VSAM functions.

- Integration of the Auto Report facility.

- System/3 RPG II equivalent functions:

  - No need to specify input array decimal positions.
  - TIME operation code to access the system time of day.
  - PRINT operation code to have the 2560 print the contents of punched fields.
  - Added device independence.

- Execution of DOS/VS RPG II programs in hierarchies of RPG II, COBOL, PL/I, and/or assembler programs.

- Revised DOS/VS RPG II cycle. File description, input, and output specifications as well as primary files are no longer mandatory; the revised cycle allows your calculations to exit from a DOS/VS RPG II subprogram at any point in the calculations.

- Data structures. Data structures, sets of named areas in virtual storage, increase the flexibility in defining, using, and redefining subfields within DOS/VS RPG II programs.

- Figurative constants. Figurative constants are available for clearing a data structure with blanks or zeros, in a single operation.

- Symbolic dump. You can request a printout of the contents of fields, data structures, arrays, and tables as a numeric value or a character string, whichever is appropriate. This printout includes the names of indicators that are ON.

- Error handling. For improved object time error handling (and recovery, if possible), error and exception information is made available as follows:

  - In a program status data structure for program errors.
  - In one or more information data structures for file errors.

  For program errors, a program may define an error subroutine which is to be executed whenever a program error occurs.

For more details on this licensed program and the manuals available for it, consult the publication *DOS/VS RPG II General Information,* GC33-6030.

DOS/VS COBOL Compiler

This licensed program product compiles source programs written in the ANS COBOL language; it is available for use under a VSE system. The compiler contains all the functions of the DOS COBOL compiler, Version 3, and includes additional support as follows:

- Support of a defined set of VSAM functions.

- Device support also for devices that are supported by VSE/Advanced Functions but not by DOS.

- The FIPS flagger, which identifies areas of a user's program that do not conform to the Federal Information Processing Standard.

Note: For COBOL source programs to be in a form suitable for this COBOL compiler, they can be converted by using the *COBOL-to-American National Standard COBOL Language Conversion* program. However, some direct programming may still be required to accomplish full conversion. The amount of this programming varies with each application program.

For more information about this compiler and the manuals available for it, consult the publication *DOS/VS COBOL General Information,* GC28-6473.

PL/I Optimizing Compiler

Pl/I is a general purpose programming language for both commercial and scientific programs. It is particularly useful when one application requires the handling of commercial and scientific problems in one and the same program.

PL/I support under a VSE system is provided by the PL/I Optimizing Compiler and by two libraries, the resident and transient libraries, all of which are available as licensed program products.

the compiler

The PL/I Optimizing Compiler is designed to provide optimized object programs from a comprehensive level of PL/I; it includes diagnostics at both compile-time and object-time.

If optimization is specified, the compiler processes the PL/I source program, reorganizing it, if necessary, so as to produce an efficient object program. If optimization is not specified, the compiler requires less compile time.

A facility is provided by the compiler to allow communication between PL/I modules and modules produced by certain FORTRAN, RPG, and COBOL compilers.

Source programs which were written for the less powerful PL/I D compiler can be compiled by the optimizing compiler provided that those programs use valid PL/I language.

the libraries

Two libraries are required for the execution of programs compiled by the optimizing compiler. These libraries contain subroutines which must be combined with the object module to produce an executable program (the PL/I resident library), and other subroutines which are required dynamically as the program is being executed (the PL/I transient library).

Both the resident and the transient libraries are separate licensed programs.

For more information about the PL/I Optimizing Compiler and the manuals available for it, consult the *PL/I Optimizing Compiler: General Information,* GC33-0004.

FORTRAN IV Library, Option 1

FORTRAN is a programming language designed for the solution of scientific and computational problems. For users of FORTRAN, the DOS FORTRAN IV compiler is available as a nonlicensed IBM-supplied Type I program.

The FORTRAN IV Library, Option 1, is available as a licensed program. Together with the Type I DOS FORTRAN IV compiler, the library allows the programmer to write, and have compiled, FORTRAN programs that:

- access files on DASDs that have been designed for attachment to System/370 central processors.

- create and process magnetic tape files which conform to the American National Standard Code for Information Interchange (ASCII).

- use larger block sizes for EBCDIC tape records.

Appendix A: Device Support

This appendix lists the various processors and also the I/O devices that are supported by VSE/Advanced Functions. The supported I/O devices are listed by classes (for example direct access storage devices, magnetic tape units, punched card devices, and so on) and within their classes by type numbers in ascending order.

IBM supplied peripheral equipment which is attachable via a supported control unit attached to an I/O channel is supported by VSE/Advanced Functions through this control unit.

Minimum machine requirements are given in "Appendix B: Operating Environment."

Central Processing Units

For available processor storage sizes less than maximum, refer either to *IBM System/370 System Summary,* GA22-7001, or to *IBM 4300 Processors Summary Input/Output and Data Communication Configurator,* GA33-1523, or else contact your IBM sales representative or the nearest IBM branch office.

	Max. storage in			Max. storage in	
Processor Model	**K bytes**	**No. of bytes**	**Processor Model**	**K bytes**	**No. of bytes**
3031	6,144	6,291456	3145**	1,024	1,048,576
3115-0*	192	196,608	3145-3	1,984	2,031,616
3115-2*	384	393,216	3148	2,048	2,097,152
3125-0*	256	262,144	3155-II	2,048	2,097,152
3125-2	512	524,288	3158	6,144	6,291,456
3135**	512	524,288	4331-1 ‡	1,024	1,048,576
			4331-2 ‡	4,096	4,194,304
3135-3	512	524,288	4341 ‡	4,096	4,194,304
3138	1,024	1,048,576			

* Must have a processor storage of 160 K bytes or more.

** Requires the optional CPU timer and clock comparator in order to establish the specified operating environment.

‡ Central processors with ECPS:VSE

Direct Access Storage Devices

IBM device type	Device name	Remarks
2311	Disk Storage Drive	See Note 1
2314	Direct Access Storage Facility	
2319	Disk Storage	See Note 2
3310	Direct Access Storage Device	A fixed-block-architecture (FBA) device (see Note 5)
3330	Disk Storage	
3333	Disk Storage	
3340	Disk Storage	
3344	Direct Access Storage	See Note 3
3350	Direct Access Storage	See Note 4
3370	Direct Access Storage Device	A fixed-block-architecture (FBA) device (see Note 5)

Note 1: Supported only as input/output device for user data.

Note 2: Supported as a 2314.

Note 3: Supported as a 3340 with one head/disk assembly of the 3344 simulating four 3340s with 3348 Model 70 data modules mounted.

Note 4: Supported also in 3330-1 compatibility mode. In that case, one non-removable head/disk assembly is equivalent to two 3330-1 volumes.

Note 5: Supported in System/370 mode on a processor with ECPS:VSE only if your VSE supervisor was generated with VM/370 linkage enhancements for operation under VM/SP or VM/370.

Magnetic Tape Units

IBM device type	Device name	Remarks
2401	Magnetic Tape Unit	
2415	Magnetic Tape Unit	
2420	Magnetic Tape Unit	
3410	Magnetic Tape Unit	
3411	Magnetic Tape Unit and Control	Supported as a 3410
3420	Magnetic Tape Unit	
8809	Magnetic Tape Unit	

Punched Card Devices

Model information for the various device types is provided only if this information is of any significance in the given context.

IBM device type	model	Device name	Remarks
1442	N1	Card Read Punch	
	N2	Card Punch	
2501		Card Reader	
2520	B1	Card Read Punch	
2520	B2/B3	Card Punch	
2540		Card Read Punch	
2560		Multifunction Card Machine	
2596		Card Read Punch	cannot be used as system input or system output device.
3504		Card Reader	
3505		Card Reader	
3525		Card Punch	
5424		Multifunction Card Unit	
5425		Multifunction Card Unit	

Printers

IBM device type	Device name	Remarks
1403	Printer	
1443	Printer	
2245	Printer	Kanji – programs using Kanji output cannot be executed in virtual mode.
3203	Printer	Models 4 and 5 are also supported as PRT1 printers.
3211	Printer	Supported as a PRT1 printer.
3213	Console Printer	For use with the console unit of a 3158.
3262	Printer	Supported as a PRT1 printer.
3284	Printer	See Note below.
3286	Printer	See Note below.
3287	Printer	See Note below.
3288	Line Printer	See Note below.
3289	Line Printer	See Note below. Model 4 is supported as a PRT1 printer.
3800	Printing subsystem	The logical I/O support must be ordered separately; for ordering procedures, contact the IBM branch office serving your locality.
5203	Printer	
5213	Console Printer	For use with the console of a 3115 or a 3125.

Note: A console (terminal) printer supported via a control unit as a 3277.

Paper Tape Equipment

IBM device type	Device name	Remarks
1017	Paper Tape Reader	
1018	Paper Tape Punch	
2671	Paper Tape Reader	

Consoles and Console Printer-Keyboards

IBM device type	Device name	Remarks
3210	Console Printer-Keyboard	
3215	Console Printer-Keyboard	
3277	Operator Display Console	
3278 Mod. 2A	Display Console	

Optical and Magnetic Character Equipment

IBM device type	Device name	Remarks
1255	Magnetic Character Reader	See Note 1
1259	Magnetic Character Reader	See Note 1
1270	Optical Reader/Sorter	See Notes 1 and 2
1275	Optical Reader/Sorter	See Notes 1 and 2
1287	Optical Reader	
1288	Optical Page Reader	
1419	Magnetic Character Reader	
3881	Optical Mark Reader	
3886	Optical Character Reader	
3890	Document Processor	The logical I/O support must be ordered separately; for ordering procedures, contact the IBM branch office serving your locality.
3895	Document Reader/Inscriber	

Note 1: Support for the device is the same as for the 1419 Magnetic Character Reader.

Note 2: Not available in the United States of America.

Control Units for Terminal Devices

VSE/Advanced Functions supports the full range of terminals that may communicate with one of the control units listed here. For detailed information about terminal support through these control units, refer to the IBM System Library publications describing the control units, or else contact your IBM sales representative or the nearest IBM branch office.

IBM device		**Device name**	**Remarks**
type	**model**		
2701		Data Adapter Unit	May be integrated in central processor (see Note 1).
2702		Transmission Control Unit	
2703		Transmission Control Unit	May be integrated in central processor (see Notes 2 and 3).
3274	1A	Control Unit	Supported as a 3791L.
3704		Communication Controller	
3705		Communication Controller	May be integrated (see Note 3).
3791L		Local Communication Controller	

Note 1: Is integrated, for example, in System/370 Model 135.

Note 2: Is integrated, for example, in System/370 Models 115 and 125; the support for the 2703 includes the support for a 3704/3705 Communication Controller in emulation mode.

Note 3: The communications adapter feature of the 4331 can be used as a 2703 or a 3705 depending on the line configuration.

Locally Attachable Display Devices

IBM device type	**Device name**	**Remarks**
2260	Display Station	
3277	Display Station	

Miscellaneous Peripheral Equipment

IBM device type	**Device name**	**Remarks**
3540	Diskette Input/Output Unit	See Note.
Feature 3401	Diskette Drive	See Note.
7443	Service Record File	On the 3031 service support console.
7770	Audio Response Unit	

Note: Supported as a unit record I/O device of type 3540.

Appendix B: Operating Environment

The operating environment of VSE/Advanced Functions is as follows:

Machine Requirements

The minimum machine requirements for Release 3 of VSE/Advanced Functions are:

- One of the following IBM processors:
  - System/370 Models 115 through 158 with a processor storage of at least 160K bytes. Recommended minimum processor storage is 256K bytes.

    VSE/Advanced Functions requires the clock comparator and CPU timer feature, which is standard on all supported processor models except Models 135 and 145; it requires the floating point feature, which is standard on all supported processor models except Models 115, 125, 135, 145, and 155.
  - An IBM 3031 processor.
  - An IBM 4300 series processor.
- A supported IBM console.
- Either of the following:
  - A supported IBM card reader.
  - A supported IBM diskette I/O unit.
- A supported IBM printer.
- One supported IBM magnetic tape drive.
- Either of the following:
  - One supported IBM disk drive with a volume capacity of 120 megabytes of disk storage.
  - Two or more supported IBM disk drives with removable disk packs if the volume capacity is less than 120 megabytes of disk storage. The total capacity of the volumes on the available disk drives must be at least 120 megabytes of disk storage. None of the disk drives may be an IBM 2311.

  If data on DASD is to be shared using the available share control facility, that data must reside on an IBM disk other than 23xx or 3310.

For a list of supported devices, see Appendix A.

Programming Requirements

VSE/Advanced Functions Release 3 combines with the DOS/VSE SCP, Program Number 5745-030-3, to provide operating system support.

Appendix C: VSE Documentation

A set of manuals and educational courses is available to describe (1) VSE/Advanced Functions, the minimum operating system support for a VSE operated installation, and (2) IBM programs that operate under control of this minimum VSE support. For education courses and manuals that answer the needs of both users new to data processing and users new only to VSE or some of its applications, consult your IBM representative or contact the IBM branch office serving your locality.

This section discusses the various types of manuals which document IBM supplied VSE programs and which comprise IBM's VSE library of publications. Following this discussion is an overview of available VSE/Advanced Functions publications. This overview (Figure C-1) may be used as a reading guide. Publications listed in this reading guide between dotted lines are not required for using VSE/Advanced Functions; references to documentation that is not part of the VSE/Advanced Functions library of publications but required for using an IBM data processing system appear between lines of equal signs (===).

types of manuals in the library

Wherever appropriate in the VSE library of publications, a distinction is made between several levels of information, each level serving a different purpose as discussed below.

General Information: General type information enables a reader to evaluate the program product's usefulness at a data processing installation. This type of information should give the reader a feel for the product's potential regarding the implementation of new applications or the enhancement of existing ones. This publication, for example, contains general type information.

Guidance Information: Guidance information describes the purpose of an available computing service and how that service can be used. The required information is presented primarily as discussions of examples and as procedures. The publication *VSE/Advanced Functions System Management Guide,* for example, contains primarily guidance information.

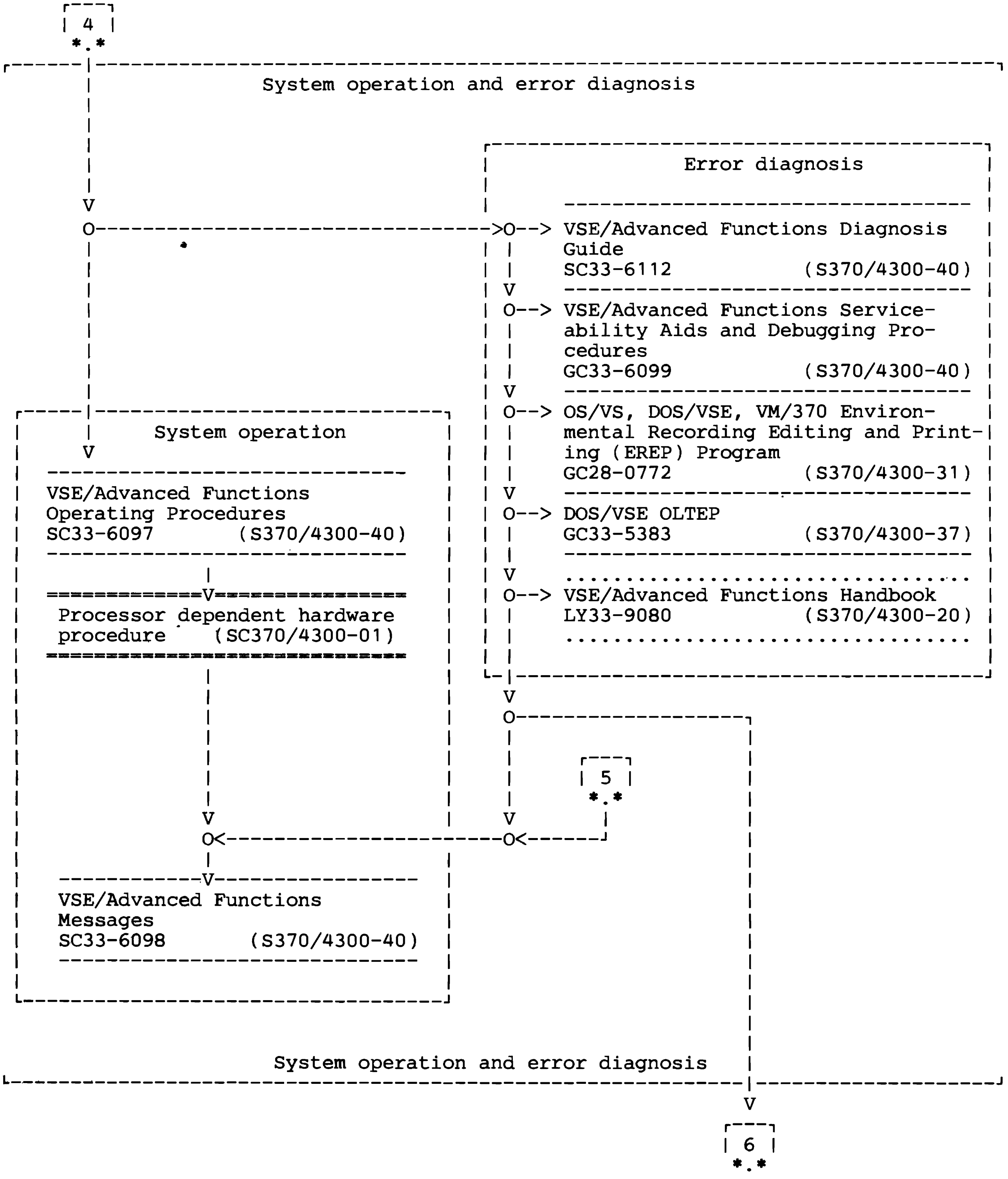
Reference Information: This type of information represents the concise specifications for using the services of VSE; it is contained in manuals that are reference sources. Accompanying explanatory text is reduced to a minimum, allowing rapid retrievability of information. The publication *VSE/Advanced Functions System Control Statements,* for example, is a quick-reference source for required control statements.

Guidance and Reference Information: In some instances, both descriptive and reference information may rightfully be contained within a single manual, one that fully covers a topic, such as *VSE/Advanced Functions Operating Procedures.*

Diagnosis Reference Information: These manuals present internal details of system programs and components and contain reference and descriptive or tutorial information. These manuals are available primarily for use by IBM programming support representatives.

```
   ┌---┐
   | 6 |
    *.*
     |
┌----|-----------------------------------------------------------------┐
     |     Error diagnosis (cont'd)
     |
     |
     V     ...................................................
     O---> VSE/Advanced Functions Diagnosis Reference:
     |     Supervisor
     |     LY33-9091                          (S370/4300-36)
     |     ...................................................
     O---> VSE/Advanced Functions Diagnosis Reference:
     |     Error Recovery and Recording Transients
     |     LY33-9092                          (S370/4300-36)
     |     ...................................................
     O---> VSE/Advanced Functions Diagnosis Reference:
     |     Logical Transients and $IJBSxxx Phases
     |     LY33-9083                          (S370/4300-36)
     |     ...................................................
     O---> VSE/Advanced Functions Diagnosis Reference:
     |     Initial Program Load and Job Control
     |     LY33-9084                          (S370/4300-36)
     |     ...................................................
     O---> VSE/Advanced Functions Diagnosis Reference:
     |     Linkage Editor
     |     LY33-9085                          (S370/4300-31)
     |     ...................................................
     O---> VSE/Advanced Functions Diagnosis Reference:
     |     Serviceability Aids
     |     LY33-9086                          (S370/4300-36)
     |     ...................................................
     O---> VSE/Advanced Functions Diagnosis Reference:
     |     Librarian
     |     LY33-9087                          (S370/4300-31)
     |     ...................................................
     O---> VSE/Advanced Functions Diagnosis Reference:
     |     System Utilities
     |     LY33-9088                          (S370/4300-32)
     |     ...................................................
     |
     |
     |     Error diagnosis (to be cont'd)
└----|-----------------------------------------------------------------┘
     V
   ┌---┐
   | 7 |
    *.*
```

Figure C-1. VSE library of publications – overview and reading guide (part 4 of 7)


4
System operation and error diagnosis
Error diagnosis
System operation
5
System operation and error diagnosis
6

VSE/Advanced Functions Diagnosis Guide
SC33-6112 (S370/4300-40)

VSE/Advanced Functions Serviceability Aids and Debugging Procedures
GC33-6099 (S370/4300-40)

OS/VS, DOS/VSE, VM/370 Environmental Recording Editing and Printing (EREP) Program
GC28-0772 (S370/4300-31)

DOS/VSE OLTEP
GC33-5383 (S370/4300-37)

VSE/Advanced Functions Handbook
LY33-9080 (S370/4300-20)

VSE/Advanced Functions Operating Procedures
SC33-6097 (S370/4300-40)

Processor dependent hardware procedure (SC370/4300-01)

VSE/Advanced Functions Messages
SC33-6098 (S370/4300-40)

Figure C-1. VSE library of publications – overview and reading guide (part 3 of 7)

```
                 ┌---┐
                 | 2 |
                  *.*
                   |
┌------------------|--------------------------------------------------------------┐
                   |    Programmed system control and data management
                   V
                   O-----User new in data management-------┐
                   |                                       |
                   |                                       V
                   |                       -------------------------------
                   V                        VSE System Data Management
                   O<---------------------- Concepts
                   |                        GC24-5209      (S370/4300-30)
                   |                       -------------------------------
                   |
                   V
                   O-----High level language user----------┐
                   |                                       |
                   |                                       V
                   |                       ===============================
               Assembler          ┌---┐       Compiler Documentation
               language           | 3 |    ===============================
                  user             *.*                     |
                   |                |                      |
                   |                └--------------------->O-----------------┐
                   |                                       |                 |
                   V                                       V                 |
     ------------------------------        -------------------------------   |
     Guide to the DOS/VSE Assembler         VSE/Advanced Functions           |
     SC33-4024       (S370/4300-21)    ┌---> Tape Labels                     |
     ------------------------------    |    SC24-5212      (S370/4300-30)    |
                   |                   |   -------------------------------   |
                   |                   |                                     |
                   V                   |                                     |
                   O------------------>O                   ┌-----------------┘
                   |                   |                   |
                   V                   └------------------>|
     ------------------------------                        |
     OS/VS-DOS/VSE-VM/370                                  V
     Assembler Language                    -------------------------------
     SC33-4010       (S370/4300-34)         VSE/Advanced Functions
     ------------------------------         DASD Labels
                   |                        SC24-5213      (S370/4300-30)
                   |                       -------------------------------
                   V
     ------------------------------
     VSE/Advanced Functions
     Macro User's Guide               ------┐
     SC24-5210       (S370/4300-30)       |
     ------------------------------     ┌---┐
                   |                    | 3 |
                   |                     *.*
                   V
     ------------------------------
     VSE/Advanced Functions
     Macro Reference
     SC24-5211       (S370/4300-30)
     ------------------------------
                   |
                   |    Programmed system control and data management
└------------------|--------------------------------------------------------------┘
                   V
                 ┌---┐
                 | 5 |
                  *.*
```

Figure C-1. **VSE library of publications – overview and reading guide (part 2 of 7)**

```
New users -------------------------------------------------┐
                                                           |
Others ------------┐                                       |
                   |                                       |
     ..............V...............         ---------------V--------------
     Introduction to the VSE System          DOS/VSE Entry User's Guide
     GC33-6108       (S370/4300-20)          GC33-6047      (S370/4300-20)
     ..............................         ------------------------------
                   |
                   V                                                  ┌───┐
                   O--Programmed system control / data management --> | 2 |
                   |                                                   *.*
                   |
                   |
                   V                                                  ┌───┐
                   O--System operation / error diagnosis -----------> | 4 |
                   |                                                   *.*
                   |
                   |
                   V                                                  ┌───┐
                   O---Program product information -----------------> | 8 |
                   |                                                   *.*
                   |
┌------------------|---------------------------------------------------------------┐
                   | System planning, system generation, system management
                   |
     --------------V---------------         ------------------------------
     VSE/Advanced Functions                  Data Security Under the
     System Management Guide        ------>  VSE System
     SC33-6094       (S370/4300-34)          GC33-6077      (S370/4300-36)
     ------------------------------         ------------------------------
                   |
       ┌-----------O-------------------------┐
       |                                     |
  ┌----|-----------------------------┐   ┌---|------------------------------┐
  |    | System generation           |   |   | System management            |
  |    |                             |   |   |                              |
  |  --V---------------------------  |   |  -V----------------------------  |
  |  VSE/Advanced Functions          |   |   VSE/Advanced Functions         |
  |  System Generation               |   |   System Control Statements     ------┐
  |  SC33-6096       (S370/4300-34)  |   |   SC33-6095      (S370/4300-36)  |    |
  |  ------------------------------  |   |  ------------------------------  |  ┌───┐
  |                |                 |   |                |                 |  | 5 |
  |                |                 |   |                |                 |   *.*
  |  --------------V---------------  |   |  --------------V---------------  |
  |  VSE/Advanced Functions          |   |   VSE/Advanced Functions         |
  |  Maintain System History Pro-    |   |   System Utilities               |
  |  gram, SC33-6101 (S370/4300-34)  |   |   SC33-6100      (S370/4300-32)  |
  |  ------------------------------  |   |  ------------------------------  |
  |                |                 |   |                |                 |
  └----------------|-----------------┘   |                |                 |
                   |                     |  --------------V---------------  |
                   |                     |   Device Support Facilities      |
                   |                     |   GC35-0033      (S370/4300-32)  |
                   |                     |  ------------------------------  |
                   V                     |                |                 |
                   O<-------------------------------------┘                 |
                   |                     └----------------------------------┘
                   |
 System planning,  | system generation, system management
└------------------|---------------------------------------------------------------┘
                   |
                 ┌───┐
                 | 5 |
                  *.*
```

Figure C-1. VSE library of publications – overview and reading guide (part 1 of 7)

```
       ┌───┐
       | 7 |
       *.*
        |
┌───────|─────────────────────────────────────────────────────────────────────┐
        |      Error diagnosis (cont'd)
        |
        |
        V      ....................................................
        O---> VSE/Advanced Functions Diagnosis Reference:
        |      Maintain System History Program
        |      SY33-9089                             (S370/4300-34)
        |      ....................................................
        O---> Device Support Facilities Logic
        |      SY35-0030                             (S370/4300-32)
               ....................................................
        O---> DOS/VSE Assembler Logic
        |      SY33-8567                             (S370/4300-21)
        |      ....................................................
        O---> VSE/Advanced Functions Diagnosis Reference:
        |      LIOCS Volume 1 General Information and Imperative
        |      Macros
        |      LY24-5209                             (S370/4300-30)
        |      ....................................................
        O---> VSE/Advanced Functions Diagnosis Reference:
        |      LIOCS Volume 2 SAM Logic
        |      LY24-5210                             (S370/4300-30)
        |      ....................................................
        O---> VSE/Advanced Functions Diagnosis Reference:
        |      LIOCS Volume 3 DAM and ISAM Logic
        |      LY24-5211                             (S370/4300-30)
        |      ....................................................
        O---> VSE/Advanced Functions Diagnosis Reference:
        |      LIOCS Volume 4 SAM for DASD
        |      LY24-5212                             (S370/4300-30)
        |      ....................................................
        O---> OS/VS, DOS/VSE, VM/370 EREP Logic
        |      SY28-0773                             (S370/4300-31)
        |      ....................................................
        O---> DOS/VSE OLTEP  Logic
               SY33-8568                             (S370/4300-37)
               ....................................................

               Error diagnosis
└─────────────────────────────────────────────────────────────────────────────┘
```

Figure C-1. VSE library of publications – overview and reading guide (part 5 of 7)

```
  ┌───┐
  | 8 |
  *.*
    |
┌───|──────────────────────────────────────────────────────────────────────┐
    |                   Program product information
    |
    |      ..............................................
    O----> VSE/POWER, General Information
    |      GH12-5131                       (S370/4300-36)
    |      ..............................................
    O----> VSE/VSAM, General Information
    |      GC24-5143                       (S370/4300-30)
    |      ..............................................
    O----> Data Language/I Disk Operating System/Virtual
    |      Storage (DL/I DOS/VS) General Information
    |      GH20-1246                       (S370/4300-50)
    |      ..............................................
    O----> Customer Information Control System/VS
    |      (CICS/VS) General Information
    |      GC33-0066                       (S370/4300-50)
    |      ..............................................
    O----> VSE/Interactive Computing and Control Facility
    |      General Information
    |      GC33-6066                       (S370/4300-39)
    |      ..............................................
    O----> VSE/Access Control Logging and Reporting
    |      General Information
    |      GH12-5130                       (S370/4300-36)
    |      ..............................................
    O----> VSE/Operator Communication Control Facility
    |      General Information
    |      GC33-6113                       (S370/4300-36)
    |      ..............................................
    O----> Advanced Communications Function for VTAM
    |      Entry, General Information:  Introduction
    |      GC27-0438                       (S370/4300-30)
    |      ..............................................
    O----> Advanced Communications Function for VTAM
    |      General Information:  Introduction
    |      GC27-0462                       (S370/4300-30)
    |      ..............................................
    O----> BTAM-ES, General Information
    |      GC38-0292                       (S370/4300-30)
    |      ..............................................
    O----> VSE/Interactive Problem Control System
    |      General Information
    |      GC34-2017                       (S370/4300-31)
    |      ..............................................
    |
    |          Program product information (to be cont'd)
└───|──────────────────────────────────────────────────────────────────────┘
    |
    V
  ┌───┐
  | 9 |
  *.*
```

Figure C-1. VSE library of publications – overview and reading guide (part 6 of 7)

```
       ┌───┐
       | 9 |
       *.*
        |
┌────|────────────────────────────────────────────────────────────────────────┐
        |              Program product information (cont'd)
        |
        |      ..............................................
        O----> IBM System 1401/1440/1460 Emulator Program
        |      Licensed Program Specifications
        |      GC33-6071
        |      ..............................................
        O----> VSE/IBM System/3--3340 Data Import
        |      Licensed Program Specifications
        |      GC33-6062
        |      ..............................................
        O----> DOS/VS Sort/Merge, General Information
        |      GC33-4030                      (S370/4300-33)
        |      ..............................................
        O----> VSE/Interfile Transfer Testing and Operations
        |      Utility, General Information
        |      GH19-6072                      (S370/4300-32)
        |      ..............................................
        O----> VSE/Fast Copy Data Set Program, General
        |      Information
        |      GC33-6081                      (S370/4300-32)
        |      ..............................................
        O----> Subsystem Support Services User's Guide
        |      GC30-3022                      (S370/4300-31)
        |      ..............................................
        O----> DOS/VS COBOL, General Information
        |      GC28-6473                      (S370/4300-24)
        |      ..............................................
        O----> PL/I Optimizing Compiler: General Information
        |      GC33-0004                      (S370/4300-29)
        |      ..............................................
        O----> DOS/VS RPG II General Information
        |      GC33-6030                      (S370/4300-28)
        |      ..............................................
        O----> FORTRAN IV Library Option 1, Program Product
               Specifications
               GC28-6473
               ..............................................

                          Program product information
└─────────────────────────────────────────────────────────────────────────────┘
```

Figure C-1. VSE library of publications – overview and reading guide (part 7 of 7)

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 indicated by an asterisk (*).

The glossary defines terms as they are used in this book. If you do not find the term you are looking for, refer to *IBM Data Processing Glossary,* GC20-1699.

access control: In VSE, a facility that provides for programmed control of a user's authorization to access protected data.

Access Control – Logging and Reporting: A licensed IBM program used for the logging of access to protected data and for the printing of selected and formatted reports on such access.

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

ACF/VTAM: see Virtual Telecommunication Access Method.

ACF/VTAME: see Virtual Telecommunication Access Method.

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.

APAR: see authorized program analysis report.

application programs: A program written for or by a user; a program that applies to the user's 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: In VSE, a facility that allows the operator to defer the reply to a system message which requires a response.

authorized program analysis report (APAR): A report of a problem caused by a suspected defect in a current release of a program.

auto-polling: A facility that handles negative responses to polling without interrupting the central processor.

auxiliary storage: A storage device that is other than processor storage; for example storage on direct access devices.

Basic Telecommunication Access Method (BTAM): An IBM-supplied licensed access method known as BTAM-ES. It permits read and write communications with remote devices.

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 source statement library.

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

BTAM: See Basic Telecommunication Access Method.

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

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

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

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

***catalog:** (noun) (1) A directory of locations of files and libraries. (2) An ordered compilation of item descriptions and sufficient information to afford access to the items.

***central processor:** Synonym for central processing unit.

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

chaining: (1) A method for storing records. When this method is used, each stored record has a link field which points to the next record in the processing chain. Physically, the records may be stored anywhere on the storage media. (2) Synonymous with concatenation.

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.

channel command word (CCW): A doubleword that directs a channel, control unit, or device to perform an operation or a set of operations.

channel program: One or more channel command words that control a specific sequence of channel operations.

CICS: See Customer Information Control System.

CKD: See count-key-data device.

COBOL: Common business-oriented language; a programming language designed for business data processing applications.

communication line: Any physical link, such as a wire or a telephone circuit, that connects one or more remote terminals to a communication control unit, or connects one communication control unit with another.

***compile:** (1) To translate a computer program expressed in a problem-oriented language into a computer-oriented language. (2) 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 used to compile.

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

concatenation: In VSE, a logical connection of libraries of the same type such that these libraries are treated as one library.

configuration: The group of machines, devices, etc. which make up a data processing system.

control program: A program that is designed to schedule and supervise the performance of data processing work by a computing system.

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

core image library: A library of phases that have been produced as output from link-editing. The phases in the core image library are in a format that is executable either directly or after processing by the relocating loader in the supervisor.

CPU: See central processing unit.

count-key-data (CKD) device: A disk storage device for storing data in the format count field normally followed by a key field followed by the actual data of a record. The count field contains, besides other information, the address of the record in the format CCHHR (CC = cylinder number, HH = head number, R = record number) and the length of the data; the key field contains the record's key (search argument).

cross-domain link: A telecommunication line physically connecting two domains. See also local-to-local link.

Customer Information Control System (CICS): An IBM program product that enables transactions entered at remote terminals to be processed concurrently by user-written application programs. The product includes facilities for building, using, and servicing a data base.

data base: A set of data that is sufficient for one or more purposes at one or more data processing installations.

data import: The process of reformatting data that was used under one operating system (for example, IBM System/3) such that it can subsequently be used under a different operating system (for example, the VSE system).

Data Interfile Transfer, Testing and Operations utility (DITTO): An IBM program product that provides file to file services for card I/O, magnetic tape, and disk devices.

Data Language/I (DL/I): A data base access method available from IBM as a program product for users of a VSE controlled computing system.

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

data security: See access control.

data set: A collection of data in one of several prescribed arrangements and described by control information to which the operating system has access. Synonymous with file.

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

***debug:** To detect, trace, and eliminate mistakes in computer programs or in other software. Synonymous with checkout.

default value: The choice among exclusive alternatives made by the system when no explicit choice is specified by the user.

Device Support Facilities: A program of VSE/Advanced Functions; it is used for executing operations on disk volumes so that these volumes can be used in a VSE system. Examples of these operations are initializing a disk volume and assigning an alternate track.

direct access: (1) Retrieval or storage of data by a reference to its location on a volume, other than relative to the previously retrieved or stored data. (2)* Pertaining to the process of obtaining data from, or placing data into, storage where the time required for such access is independent of the location of the data most recently obtained or placed in storage. (3)* Pertaining to a storage device in which the access time is effectively independent of the location of the data. Synonymous with random access.

direct file organization: Implies that, for the purpose of storage and retrieval, there is a direct relationship between the contents of the records and their addresses on disk storage.

directory: An index that is used by the system control programs 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.

DITTO: See Data Interfile Transfer, Testing and Operations utility.

DL/I: See Data Language/I.

domain: In a telecommunication system, the portion of the total network that is controlled by the system services control point in one data communication access method.

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

dynamic partition balancing: See partition balancing.

emulation: The use of programming techniques and special machine features that permit a computing system to execute programs written for another system or for the use of I/O devices different from those that are available.

entry-sequenced file: A VSAM file whose records are loaded without respect to their contents, and whose relative byte addresses cannot change. Records are retrieved and stored by addressed access, and new records are added to the end of the file.

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.

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

FBA: See fixed-block-architecture device.

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

hard copy: A printed copy of machine output in a visually readable form, for example, printed reports, lists, documents, and summaries.

***hardware:** Physical equipment, as opposed to the computer program or method of use, for example, mechanical, magnetic, electrical, or electronic devices. Contrast with software.

ICCF: See Interactive Computing and Control Facility.

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.

indexed-sequential access method (ISAM): An access method used to retrieve data from a file whose records are stored sequentially by key.

indexed-sequential organization: The records of an indexed sequential file are arranged in logical sequence by key. Indexes to these keys permit direct access to individual records. All or part of the file can be processed sequentially.

initial program load (IPL): The initialization procedure that causes VSE/Advanced Functions to commence operation.

Interactive Computing and Control Facility (ICCF): An IBM program product for use at VSE installations; it is referred to in this publication as VSE/ICCF. Through VSE/ICCF, the services of a VSE controlled computing system become available, on a time-sliced basis, to authorized users of terminals that are linked to the system's central processor.

Interactive Problem Control System (IPCS): An IBM program product for use at VSE installations; it is referred to in this publication as VSE/IPCS. A programming tool that assists in software problem determination and in submitting operating system problems to IBM.

***interface:** A shared boundary. An interface might be a hardware component to link two devices or it might be a portion of storage or registers accessed by two or more computer programs.

***I/O:** An abbreviation for input/output.

IPCS: See Interactive Problem Control System.

IPL: See initial program load.

irrecoverable error: A hardware error which cannot be recovered from by the normal retry procedures.

ISAM: See indexed-sequential access method.

ISAM interface program: A set of routines that allow a processing program coded to use ISAM to access a key-sequenced VSAM file.

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: 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 a job or part of a job to be run. Some of its functions are to assign I/O devices to logical unit names, set switches for program use, log (or print) job control statements, and fetch the first program phase of a job.

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.

key sequence: The collating sequence of data records, determined by the value of the key field in each of the data records. May be the same as, or different from, the entry sequence of the records.

key-sequenced file: A file whose records are loaded in key sequence and controlled by an index. Records are retrieved and stored by keyed access or by addressed access, and new records are inserted in the file in key sequence.

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.

language translator: A general term for any assembler, compiler, or other routine that accepts statements in one language and produces equivalent statements in another language.

librarian: The set of programs that maintains, services, and organizes the system and private libraries.

library: A collection of files or programs that are related by some common characteristic. For example, all phases in the core image library have been processed by the linkage editor.

linkage editor: A 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) For a VSE system, to bring a program phase from a core image library into virtual storage for execution.

local-to-local link: A telecommunication link between two local communications controllers controlled by the same system services control point.

logical unit (logical device) name: A name used in programming to represent an I/O device address.

macro: In assembler language programming, a statement that causes the assembler to process a predefined set of statements called macro definition. The result of this processing, a sequence of machine instructions, replaces the macro in the source program.

***magnetic ink character recognition (MICR):** The recognition of characters printed with ink that contains particles of a magnetic material.

Maintain System History program: A program used for automating and controlling various installation, tailoring, and service activities for a VSE system.

main task: The program (or phase) loaded into a partition for execution as a result of a load request to job control (see also subtask).

message: See error message, operator message.

MICR: See magnetic ink character recognition.

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

multitasking: The concurrent execution of one main task and one or more subtasks in the same partition.

object code: Output from a compiler or assembler which is suitable for processing 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.

OCCF: See Operator Communication Control Facility.

***online:** Pertaining to the operation of a functional unit that is under the continual control of a computer. The term is also used to describe a user's access to a computer via a terminal.

***operating system:** Software that controls the execution of computer programs and that may provide scheduling, debugging, input/output control, accounting, compilation, storage assignment, data management, and related services.

operator command: A statement to the control program, issued via a console device; it causes the control program to provide requested information, alter normal operations, initate new operations, or terminate existing operations.

Operator Communication Control Facility (OCCF): An IBM program product that is designed to reduce operator interaction in the operation of a VSE controlled installation and to centralize data processing skills.

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 fixed-length block of instructions, of data, or of both that can be transferred between processor storage and the page data set. In VSE, a page is 2K bytes in length. (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: A 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.

partition: A division of the address area of that part of virtual storage which is available for the execution of problem programs.

partition balancing: A facility which allows the user of a VSE system to specify two, more, or all of the system's partitions to have their processing priorities changed dynamically such that the system privileges that partition which has the lowest record of central processor usage for a specific time interval.

phase: The smallest complete unit that can be referred to in the core image library.

***PL/I:** A programming language designed for use in a wide range of commercial and scientific computer applications.

polling: In telecommunication, the process of inviting remote stations to transmit, one at a time. Polling usually involves the sequential interrogation of several remote stations.

POWER: See VSE/POWER.

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

priority: A rank assigned to a partition; it determines that partition's 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.

processing program: (1) A general term for any program that is not a control program. (2) Synonymous with problem program.

processor storage: The storage contained in a central processor and available for the execution of machine instructions. Synonymous with real storage.

program temporary fix (PTF): A temporary solution or by-pass of a problem diagnosed by IBM field engineering as a result of a defect in a current unaltered release of a program.

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.

random access: See direct access.

random processing: The treatment of data without respect to its location on direct access storage, and in an arbitrary sequence governed by the input against which it is to be processed.

real address: The address of a location in real (processor) storage.

real address area: The area of virtual storage where virtual addresses are equal to real (processor storage) addresses.

real mode: In VSE, the execution mode of a program that may not be paged.

real storage: See processor storage.

recovery management support (RMS): Facilities of VSE/Advanced Functions; these facilities gather information about hardware reliability and allow retry of operations which failed because of central processor, I/O device, or channel errors.

relocatable library: A library of relocatable object modules and IOCS modules required by various compilers. It allows the user to keep frequently used modules available for combination with other modules without recompilation.

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

RMS: See recovery management support.

RPG II: A commercially oriented programming language designed for writing application programs that meet common business data processing requirements.

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

SCP: See system control program.

SDL: See system directory list.

second level directory (SDL): An index in the supervisor. For each library directory track (on a CKD disk) or a 2K group of library directory blocks (on an FBA disk), this index contains the phase name highest in collating sequence on the track or in the group of blocks.

security: See access control.

sequential organization: Records of a sequential file are arranged in the order in which they will be processed.

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 high address range of virtual storage. It can contain a system directory list (SDL) of frequently-used phases, resident programs that can be shared between partitions, and an area for dynamic allocation to components of the VSE system.

SNA: See system network architecture.

software: A set of programs concerned with the operation of the hardware in a data processing system.

***source program:** A computer program written in a source language. Contrast with object program.

source statement library: A collection of books (such as macro definitions) cataloged in the system by the librarian program.

spanned records: Records of varying length that may be longer than the currently used blocksize, and which may therefore be written in one or more continuous blocks. A spanned record may occupy more than one track of a CKD disk device or more than one physical block of data of an FBA device.

spooling: The reading and writing of input and output streams on auxiliary storage devices, concurrently with job execution, in a format convenient for later processing or output operations.

SSS: See Subsystem Support Services.

stand-alone program: A program that runs independently of (not controlled by) the VSE system.

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

storage dump: See dump.

Subsystem Support Services (SSS): A set of IBM-supplied programs executed in a VSE host system; they provide services for subsystems of this host system. The services include such things as creating and updating sybsystem libraries (at the host system), processing programs and controller data to be used by the subsystems, and transmitting the programs and controller data to communication controllers and program controlled terminals.

subtask: A task that is initiated by the main task or by another subtask; a unit of work which competes for processing resources similarly to a program that is being executed in a partition.

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.

system control program (SCP): IBM-supplied, nonlicensed 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 library: A library defined to the operating system to contain programs or parts of programs required for system operation; a system library is stored on the system residence volume.

system network architecture (SNA): The total description of the logical structure, formats, protocols, and operational sequences for transmitting information units through a communication system. The structure of SNA allows the ultimate origins and destinations of information – that is, the end users – to be independent of and unaffected by the specific communication-system services and facilities used for information exchange.

system residence device: The direct access device on which the system residence volume is mounted.

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

system service program: see service program.

system utility program: see utility program.

task: a unit of work for the central processing unit from the standpoint of the control program.

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.

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. The area is a serially reusable system resource.

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 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 a VSE operating in System/370 mode, the area of virtual storage whose addresses are greater than the highest address of the real address area.

virtual mode: In VSE, 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: A data communications access method available from IBM as the Advanced Communications Function product for the Virtual Telecommunications Access Method (ACF/VTAM) or for the Virtual Telecommunications Access Method Entry (ACF/VTAME). Either product provides single-domain and multiple-domain data communication capability. ACF/VTAM is available for systems that use communications controllers, ACF/VTAME for systems that have a communications adapter.

volume: (1) 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.

VSAM: See Virtual Storage Access Method.

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

VSE/POWER: The name of an IBM program product primarily used for the spooling of unit record input and output.

VTAM: See Virtual Telecommunications Access Method.

work file: A file on a secondary storage medium reserved for intermediate results during execution of a particular program.

Index

GC33-6108-1

Introduction to the VSE System (File No. S370/4300-20) Printed in U.S.A. GC33-6108-1

IBM®

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, N.Y. 10604

IBM World Trade Americas/Far East Corporation
Town of Mount Pleasant, Route 9, North Tarrytown, N.Y., U.S.A 10591

IBM World Trade Europe/Middle East/Africa Corporation
360 Hamilton Avenue, White Plains, N.Y., U.S.A. 10601

READER'S
COMMENT
FORM

Introduction to the VSE System
Order No. GC33-6108-1

This manual is part of a library that serves as a reference source for system analysts, programmers, and operators of IBM systems. This form may be used to communicate your views about this publication. They will be sent to the author's department for whatever review and action, if any, is deemed appropriate. 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 comment 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.)

GC33 6108 1

Cut or Fold Along Line

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 Corporation
Department 812 BP
1133 Westchester Avenue
White Plains, New York 10604

Fold and tape | Please Do Not Staple | Fold and tape

IBM®

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, N.Y. 10604

IBM World Trade Americas/Far East Corporation
Town of Mount Pleasant, Route 9, North Tarrytown, N.Y., U.S.A 10591

IBM World Trade Europe/Middle East/Africa Corporation
360 Hamilton Avenue, White Plains, N.Y., U.S.A. 10601

Introduction to the VSE System (File No. S370/4300-20) Printed in U.S.A. GC33-6108-1

READER'S
COMMENT
FORM

Introduction to the VSE System
Order No. GC33-6108-1

This manual is part of a library that serves as a reference source for system analysts, programmers, and operators of IBM systems. This form may be used to communicate your views about this publication. They will be sent to the author's department for whatever review and action, if any, is deemed appropriate. 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 comment 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.)

GC33-6108-1

Cut or Fold Along Line

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 Corporation
Department 812 BP
1133 Westchester Avenue
White Plains, New York 10604

Fold and tape Please Do Not Staple Fold and tape

Introduction to the VSE System (File No. S370/4300-20) Printed in U.S.A. GC33-6108-1

IBM®

**International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, N.Y. 10604**

**IBM World Trade Americas/Far East Corporation
Town of Mount Pleasant, Route 9, North Tarrytown, N.Y., U.S.A 10591**

**IBM World Trade Europe/Middle East/Africa Corporation
360 Hamilton Avenue, White Plains, N.Y., U.S.A. 10601**

GC33-6108-1

Introduction to the VSE System (File No. S370/4300-20) Printed in U.S.A. GC33-6108-1

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, N.Y. 10604

IBM World Trade Americas/Far East Corporation
Town of Mount Pleasant, Route 9, North Tarrytown, N.Y., U.S.A 10591

IBM World Trade Europe/Middle East/Africa Corporation
360 Hamilton Avenue, White Plains, N.Y., U.S.A. 10601