Virtual Machine/System Product
High Performance Option

SC23-0189-4

Release 6 Guide

Release 6

IBM

Virtual Machine/System Product High Performance Option

SC23-0189-4

Release 6 Guide

Release 6

The term "VM/SP High Performance Option" applies to the VM/SP High Performance Option Licensed Program when used in conjunction with the VM/System Product Licensed Program.

Fifth Edition (June 1989)

This edition applies to Release 6 of IBM Virtual Machine/System Product High Performance Option (Program Number 5664-173). Changes are periodically made to the information contained herein; before using this publication to operate IBM systems, consult the latest *IBM/370, 30xx, 4300, and 9370 Processors Bibliography*, GC20-0001, for the editions that are applicable and current.

To order previous editions of this manual, use the following temporary order numbers:

Release 4.2	ST00-1903
Release 5	ST00-2750

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any functionally equivalent program may be used instead.

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

A form for readers' comments is at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, Department 52Q/MS 511, Neighborhood Rd., Kingston, N.Y., U.S.A. 12401. IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Preface

Purpose of This Book

This publication provides an overview of the IBM Virtual Machine/System Product High Performance Option (VM/SP HPO), Release 6. It will help you decide whether this release is suitable for your installation. This book also contains information that will help you begin planning for migration to VM/SP HPO Release 6.

Who Should Read This Book?

This book is intended for users of VM/SP HPO who plan to migrate to Release 6. It is for data processing managers, system programmers, system analysts, and other programming personnel as well as IBM service personnel responsible for migrating an installation to VM/SP HPO Release 6. Other users, such as application programmers, can also learn about the new release.

What You Should Know Before Reading This Book

This book assumes that you understand System/370 hardware, virtual storage concepts and terminology. It also assumes that you are familiar with operating system concepts in general and with any operating system that you will run under control of VM/SP HPO.

What This Book Contains

- Part 1 — provides an overview of between-release support, summarizes the system requirements and planning information, describes the changes in the VM/SP HPO library, and discusses program distribution.
- Part 2 — provides an overview of VM/SP Release 6
- Part 3 — discusses migration and performance considerations for Release 6.
- Part 4 — discusses changes to the internal design of Release 6.
- Appendix A — provides an overview of VM/SP HPO prior to this release.
- Glossary — defines terms used in this publication.

Where to Find More Information

See Chapter 4, "VM/SP HPO Publications" on page 29 for the library diagram and a discussion of the new and changed VM/SP HPO publications.

Contents

Part 1. Overview of VM/SP HPO Release 6

VM/SP HPO Release 6 has no unique functions. VM/SP HPO Release 6 is a retrofit of code in VM/SP Release 6. For a discussion of the new functions added for VM/SP Release 6, see Part 2 of this manual.

This part of the manual discusses:

- Support shipped between the general availability of VM/SP HPO Release 5 and VM/SP HPO Release 6 and contained in VM/SP HPO Release 6
- Unique VM/SP HPO Release 6 installation and service updates
- System requirements and planning information for VM/SP HPO Release 6
- The VM/SP HPO Release 6 library
- Program distribution.

Chapter 1. Between-Release Support

The following support was shipped since VM/SP HPO Release 5 and is integrated in VM/SP HPO Release 6:

- DIAGNOSE code X'64' Subcode X'0014'
- DIAGNOSE code X'E4'
- SPOOL enhancement accommodation
- 3380 Models AJ4/BJ4 and AK4/BK4
- 3990 Storage Control Unit Models 1 and 2 and 3380 Model CJ2
- Enhanced connectivity facilities
- DASD dump restore streaming support improvements
- Terminal usability enhancements
- National language support enhancement
- National language support file naming conventions enhancements
- Support for the IBM 3990 Storage Control Model 3
- Support for mixed directory
- Free storage enhancement
- Paging storage support and Vector Facility changes
- VM/Directory Maintenance Licensed Program
- Security and integrity enhancements
- Alternate VSAM
- PR/SM™ support
- Enhanced VSE support
- Enhanced dispatcher request queue management
- Reduced checkpoint I/O operations.

DIAGNOSE Code X'64' Subcode X'0014'

Subfunction code X'0014' (ADDRQRY) is added to DIAGNOSE code X'64' so you can request information about a virtual storage address. With ADDRQRY, you can determine whether a page is shared as well as the following:

- Segment name if the address is within a saved segment
- Number of users having this segment loaded if the virtual address is within a shared segment
- Whether the virtual address is within a protected area of a shared segment
- Whether the storage is discontiguous from the invoking user's address space.

PR/SM is a trademark of the International Business Machines Corporation.

Like the other function codes of DIAGNOSE code X'64', ADDRQRY is available to all privilege classes (except privilege class ANY).

Refer to the *VM System Facilities for Programming* or *VM/SP HPO Administration* for more information.

DIAGNOSE Code X'E4'

A new diagnose code, X'E4', gives authorized users these capabilities:

- DIAGNOSE code X'E4' can be used with subcodes X'0000' and X'0001' to return relocation and real device information about a minidisk. Specifically, X'0000' is used to reference the device to which a logged-on user is linked; X'0001' is used to reference the device as defined in the directory entry of the user ID specified.
- DIAGNOSE code X'E4' can be used with subcode X'0002' to dynamically create a full-pack overlay read/write minidisk of the volume on which a specified user's minidisk resides, provided the user's minidisk extents are allocated as PERM space (that is, they do not overlap PAGE, SPOOL, DIRECTORY, or DUMP space).
- DIAGNOSE cod X'E4' can be used with subcode X'0003', with the real device address and the cylinder/block number specified as input, to get a full-pack minidisk overlay, provided the cylinder is allocated as PERM space (that is, there is no overlap of PAGE, SPOOL, DIRECTORY, or DUMP space.

Also, some CCW operation codes have been extended so that they can be issued for a full-pack overlay minidisk that was defined by subcodes X'0002' or X'0003' of DIAGNOSE code X'E4'. The Write Home Address (WHA), Diagnostic Write Home Address (DWHA), Write Record Zero (WRO) and DIAGNOSTIC CNTL (DIAGCNTL) functions are correctly simulated for all cylinders of a full-pack overlay minidisk instead of being translated to an equivalent READ operation. Access to the CE cylinders and to the DEFINE EXTENT CCW is valid for the full-pack overlay user. The information from the Read Device Characteristic (RDC) CCW is returned untranslated to the full-pack overlay user.

SPOOL Enhancement Accommodation

The IUCV SPOOL system service uses the spool file ID and the spool file userid to identify a spool file to the printer subsystem. This interface does not change the way VM generates spool files. It acts as a common interface to a printer subsystem.

Refer to *VM System Facilities for Programming* for more information about SPOOL enhancement accommodation.

3380 Models AJ4/BJ4 and AK4/BK4

The 3380 Models AJ4/BJ4 and AK4/BK4 are enhanced subsystem units that offer improved data availability and error recovery. The support for these models is functionally equivalent to that for currently supported 3380 models, with additional support to recognize the increased number of cylinders in the 3380 AK4/BK4.

Each 3380 AJ4, BJ4, AK4, or BK4 unit contains four storage devices. Each device represents an addressable data volume. Models AJ4 and BJ4 have 885 cylinders per volume, providing a capacity of 2.52 billion bytes per unit. Models AK4 and BK4 have 2655 cylinders per volume, providing a capacity of 7.56 billion bytes per unit. Data is stored in the count-key-data (CKD) record format.

A-units (AJ4 and AK4) contain two controllers to control data transfer between the 3380 and the IBM 3880 Storage Control. B-units (BJ4 and BK4) do not contain controllers, and must be connected to A-units in string configuration. Up to three B-units can be attached to an A-unit.

The Device Level Selection (DLS) function of the 3380 AJ4/BJ4 and AK4/BK4 provides two independent data paths in the DASD string, one from each of the two controllers in the A-unit to each device (actuator) in the string (four to sixteen addresses). This allows any two devices in the string to read or write data at the same time through different controllers.

CP Support

The 3380 Models AK4/BK4 use the CP support previously introduced with the 3380 Models AE4/BE4 that lets you enter 4-digit cylinder numbers in commands and control statements.

Format and Allocation

The FORMAT/ALLOCATE program (DMKFMT) has been modified to allocate the 2655 cylinders of the 3380 AK4/BK4. The maximum size of the DASD allocation map has been increased to 4K (4096 bytes).

When you use the FORMAT, LABEL, and ALLOCATE functions of DMKFMT, specify *3380* as the device type in the control statements; no explicit device model is accepted. Because the numbering of cylinders begins with cylinder 0, the highest allocatable ending cylinder is *2654*.

CP functions that read the DASD allocation map have also been modified. These CP functions include:

- CP Initialization Routine
- Paging
- CP directory
- VARY, ATTACH, and OVERRIDE command processors
- Directory program (DMKDIR)
- DASD Dump/Restore Program (DMKDDR)
- Stand-alone dump program (DMKSAD)

New 3380 Migration Aid

The 3380 Migration Aid (MIG3380) is a new tool that you can use to adjust the allocation map after you migrate data between different 3380 models.

A 3380 DASD volume is formatted with a 1K, 2K, or 4K allocation map according to the number of cylinders contained in the volume, which varies from model to model. When you back up a 3380 CP volume using DASD Dump Restore (DDR) and restore the data to another 3380 volume that has a different number of cylinders, the original CP allocation map is carried over, and does not match the new volume. The MIG3380 program rearranges the contents of cylinder 0 on the new volume to correct the size of the allocation map and filler records, while preserving CP-specific information.

Error Recovery and Recording

When VM/SP HPO records hardware error conditions for 3380 devices, the following records now provide additional information:

- CCHREC (Channel Check Handler) identifies the device class, the device type, and the control unit ID.
- MDRREC (Miscellaneous Data Recording) identifies the control unit ID and the secondary device address, and includes up to 32 bytes of sense data.
- OBRRECN (Outboard, or Unit Check) identifies the device class, the device type, and the control unit ID, and includes up to 32 bytes of sense data.
- TNSREC ("T" Type Format) identifies the control unit ID and the secondary device, and includes up to 32 bytes of sense data.

VM/SP HPO modules that process sense data have been modified to handle 32-byte input.

Refer to the *VM/SP HPO Planning Guide and Reference* for more information.

3990 Storage Control Unit Models 1 and 2 and 3380 Direct Access Storage Direct Channel Attach Model CJ2

The 3990 model 1 is a new storage control providing two internal storage paths. The 3990 model 2 storage control provides four storage paths. Both 3990 models 1 and 2 storage controls provide for attachment of all models of 3380 direct access storage (except A04). Both models attach to block multiplexer channels capable of data transfer rates of 3.0 megabytes or greater. Both support the extended count key data (ECKD) architecture and provide other enhancements to surpass the capabilities of the 3880 family of storage controls.

The 3380 Direct Access Storage Direct Channel Attach Model CJ2 provides both the direct access storage of a 3380 and the storage control functions of a 3990 Model 1 (the two storage path model) in a single unit. The DASD in the 3380 Model CJ2 is equivalent to a 3380 model AJ4 except that it provides a single Head Disk Assembly (HDA). Up to three 3380 'B' units (models BJ4 or BK4 only) can be attached to the 3380 Model CJ2. The 3380 Model CJ2 attaches directly to block multiplexer channels capable of data transfer rates of 3.0 megabytes or greater and provides the head-of-storage-facility function for up to 14 devices. The 3380 Model CJ2 combines the improved seek characteristics of the 3380 enhanced subsystem models with the improved storage control capabilities of the 3990 family.

Refer to the *VM/SP HPO Planning Guide and Reference* and *VM System Facilities for Programming* for more information.

Enhanced Connectivity Facilities

The initial package for enhanced connectivity facilities now supports control unit (CUT) communication as well as distributed function terminals (DFTs).

As part of this support, the user can specify either CUT or DFT on the CMSSERV command. Also, the user can set the CMSSERV command default mode to CUT or DFT via the DEFAULTS command.

DASD Dump Restore Streaming Support Improvements

This enhancement increases I/O transfer rates during DDR DUMPs and RESTOREs that involve FBA DASD and streaming tapes. It is specifically for FBA and streaming tapes and has no effect on I/O that involves CKD DASD or Start/Stop tapes.

DDR DUMP and RESTORE transfer rates are improved because now the internal DDR buffer is larger than the buffer previously used. Because of the increased buffer size, a larger number of records can be read/written at one time, thereby, requiring fewer Start I/Os (SIO) and improving data transfer rates (especially for streaming tape drives such as the 8809).

Reducing the number if SIO's results in decreased *backhitching*. Backhitching occurs when a streaming tape drive stops and repositions itself. This could take more than one second to complete. When an inter-record gap is reached, the tape continues in anticipation of the next SIO. If this SIO does not arrive in time, the tape proceeds past the next record, backs up and then performs the SIO.

Because of the larger amount of data now transferred per SIO, DDR's internal headers for FBA DASD are written to the tape differently. A tape produced in the new DDR format cannot be restored using an older version of DDR, unless you specify the OLDFORM option on the DUMP.

This support does the following:

- Reduces the time required to transfer data from an FBA DASD to a streaming tape device, and vice versa
- Supports all the existing functions of DDR (DUMP, RESTORE, COPY, TYPE and PRINT)
- Supports all input and output DASD and tape devices that are currently supported
- RESTORES backup tapes produced by versions of DDR that do not have this support
- Uses the DUMP function to produce output tapes in the old format so it can be used as input to previous levels of DDR
- Supports both compact and standard formats
- Uses less space on the system minidisk.

For the complete format of the DDR command, see *VM System Facilities for Programming*.

Terminal Usability Enhancements

The CP SET RETRIEVE Command Has a FORWARD Option

Any user of a local, remote, or VM/VTAM terminal that supports PF keys can set the PF keys to retrieve commands in both a forward and backward direction, using the enhanced RETRIEVE function. This enhanced command can save time and keystrokes when searching for a command you want to reenter.

Queued WRITEs in VM/VTAM

During a full-screen application, the queueing of messages after the execution of a write-structured field now occurs automatically.

You, as a VM/VTAM user, can also control when such messages appear on the screen, whether or not your application issues write-structured field commands. Using the TERMINAL BREAKIN options (IMmed and GUESTctl), you can let messages interrupt your full-screen application immediately, or you can place incoming messages in a queue until you release them.

Data Length and Screen Depth Checking

Now, CP checks the length and depth of the data going to all display devices, including VM/VTAM devices. If the data will fit on the screen, CP sends the data to the VSM. If the data will not fit, CP sends a CHANNEL PROGRAM CHECK (INCORRECT LENGTH) to your virtual machine and keeps your terminal from disconnecting.

Improved Full-Screen Error Handling

Now, VM/VTAM full-screen error checking includes the same tests as those made for native devices. Following are the conditions:

- Normal completion of the READ/WRITE function
- Line break, intervention required
- FSSM reject, screen in use by CP
- Operation check
- Equipment check and unit specification
- Channel data check and unit specification
- Equipment check
- Command reject
- Channel data check
- Channel control check
- Attention pending, CC1.

Because the virtual machine has more specific error information, it recovers from errors more efficiently.

VM/VTAM and Native Console Mode Read Lengths are Now Compatible

Before this support, on a VM/VTAM device, data entered via a console mode read was truncated at 240 bytes before processing. With the addition of this support, the RDTERM macro will be able to read and process up to 2030 bytes, if EDIT=PHYS, thus attaining compatibility with native devices.

If the user requests data to be edited (via the SET LINEDIT ON command) and the input is more than 240 bytes, the data will be truncated to 240 bytes for editing purposes.

National Language Support Enhancement

This enhancement allows proper National Language Support (NLS) translation of messages issued by the PEEK, RECEIVE, and RDR commands. Two new options are added to the RDR command to support this enhancement:

- MSGSUBS
- MSGALL.

MSGUBS specifies that the RDR command returns only the available substitution information for the file.

MSGALL specifies that the RDR command returns the normal message and all available substitution information for the file.

Refer to the *VM/SP CMS Command Reference* for more information on the RDR command.

National Language Support File Naming Conventions Enhancements

This support enhances the VM language source file installation maintenance procedures. It allows the conventional text naming of VM language source file output. The functions introduced with this are compatible with existing functions, allowing *traditional* and *conventional* language object files to coexist. Conventional text naming for *non-language* source files allows eight character file names and file types of format TEXT or TXTptfno. The file type extension *ptfno* is the Program Temporary Fix (PTF) number that is assigned by the national Service Division (NSD) when a problem is opened. The object file type follows the naming conventions for the new service process. The file naming of *language* source files also files for consistency. The processing of VM *language* source files is enhanced to respect the source file level identifiers it the VM control files. This enhancement affects the file identifiers of VM language output files and the file handling in several language-related commands. The commands affected by this enhancement are:

CONVERT COMMANDS
VMFAPPLY/VMFAPTXT
GENMSG
VMFLDS
LANGGEN
VMFLOAD
LANGMERG
VMFNLS
SET LANGUAGE

Support for the IBM 3990 Storage Control Model 3

The 3990 Model 3 provides all of the performance, reliability, availability, and serviceability characteristics of the 3990 family. In addition to features provided by the 3990 Model 2, the model 3 provides the following:

- A choice of cache sizes: 32, 64, 128, or 256 megabytes.

- A data transfer rate between the cache and the channel of 4.5 megabytes per second when attached to a 3090 processor with a channel capable of transferring 4.5 megabytes per second.
- Improved internal cache algorithms resulting in greater efficiency.
- Two new fast write capabilities, DASD fast write and cache fast write. Both fast write capabilities permit write operations to be performed at cache access speed, eliminating the need to write data to DASD immediately.
- A dual copy capability that greatly increases the availability and accessibility of data. The dual copy capability writes all selected data to two different DASD, independent of host processing.
- A 4 megabyte nonvolatile storage (NVS) providing random-access storage with its own separate power region for data protection. The NVS is used by the DASD fast write and dual copy functions. If power is lost to the 3990 before DASD fast write or dual copy operations have completed to DASD, a battery-backup system maintains power in the nonvolatile storage for up to 48 hours to prevent data loss. When power is restored, the Model 3 destages any data in nonvolatile storage to DASD and completes any operations in progress at the time of the power failure.

Caching Operations

VM/SP HPO Release 6 CP support for the 3990 Model 3 includes the support of basic caching operations equivalent to that previously provided in VM/SP HPO for the 3880 Models 13 and 23 Storage Controls. Both CP and guest virtual machine use of the cache is supported.

In addition, cache fast write operations issued by guest virtual machines are supported. Cache fast write is designed for use with special kinds of data, such as temporary data created as sort work files, that can easily be reconstructed if lost. When sort work files are allocated on a 3990 Model 3, the intermediate work files can be kept temporarily in the cache. The data might not be written to the DASD. The channel program is complete when the data is written to cache. Cache fast write does not use nonvolatile storage. CP does not use the cache fast write function for CP system data. For more information on cache fast write, see the *IBM 3990 Storage Control Planning, Installation, and Storage Administration Guide.*

The DASD fast write and dual copy functions of the 3990 Model 3 are not supported for either CP or guest virtual machine use. All I/O operations to temporary minidisk space (T-DISK) bypass the cache. This includes cache fast write operations by guest virtual machines.

Support for Mixed Directory

In addition to adding several new CP messages, this enhancement changes:

- The CMS DIRECT command.

  The CMS DIRECT command creates an object user directory from a source directory file. This enhancement adds a new operand, MIXED, to the DIRECT command.

  The MIXED operand lets you use a source directory file that contains both VM/SP HPO and VM/XA SP directory control statements and options to create a VM/SP HPO directory. Rather than flagging VM/XA SP-unique statements and options as errors, the directory program checks them for syntax errors.

However, the VM/XA SP-unique statements and options have no effect on the VM/SP HPO object directory.

- The DIRMAINT command

  The mixed directory support enhancement changes the way the DIRMAINT command operates with the DIRECT and DIREDIT operands. When you enter either of these operands with the DIRMAINT command, DIRMAINT now checks the MIXED= statement in the DIRMAINT DATA file to see whether the MIXED operand is to be used when invoking the VM/SP HPO or VM/XA SP directory program. If it does, DIRMAINT uses the MIXED operand on the VM/SP HPO DIRECT command or the VM/XA SP DIRECTXA command, as appropriate.

- The MIXED= statement in the DIRMAINT DATA file.

  The MIXED= statement in the DIRMAINT DATA file determines the use of the MIXED operand when the DIRMAINT command invokes the directory program on a VM/SP HPO or VM/XA SP system.

Free Storage Enhancements

This programming enhancement changes the way that CP manages free storage. In particular, this support:

- Provides recovery from many free storage errors that cause abends
- Reduces the overhead associated with free storage management
- Reduces the probability that SEL007 abends will occur
- Provides an external control for swap table migration.

Paging Storage Support and Vector Facility Changes

Paging Storage (also known as Expanded Storage) is a high-speed paging medium for processors such as the 3090. This support changes the Paging Storage increment size to 64 megabytes on the 3090 Models E and S processors. The 4381 Models 91 and 92 have Paging Storage, but are not affected by this enhancement.

Specifying Paging Storage

You can specify Paging Storage in:

1. Megabytes, using the SYSXSTOR macro
2. *n*-megabyte increments, using the SYSPAG macro, where n is the increment size of Paging Storage.

It is strongly recommended that you use the SYSXSTOR macro instead of the SYSPAG macro to allocate Paging Storage. When you use the SYSPAG macro, Paging Storage is allocated in units of Paging Storage increments. You must specify the first and last increment numbers. This creates three problems:

- The smallest unit of Paging Storage that you can allocate for swapping or paging on the 3090 E and S models is 64 megabytes, which is the increment size on these processors
- The same SYSGEN will not give you the same allocation on all processors, since the Paging Storage increment size is not the same on all processors.

- The Paging Storage increment numbers for each physical or logical partition are not the same when you use a physically partitioned or a logically partitioned processor. Each partition requires a SYSGEN with a different SYSPAG macro.

The SYSXSTOR macro solves all of these problems by allowing you to specify the number of megabytes of Paging Storage that you wish to allocate for swapping and/or paging.

VM/Directory Maintenance Licensed Program (DIRMAINT) Release 4

This support improves performance:

- Since DIRMAINT does not need to process the entire source directory each time it makes an update
- By reducing I/O contention to the minidisk that contains the real object directory.

An object directory is the working directory that CP uses to get information about a particular user. For example, CP uses the object directory to find the user's password at logon time.

With this support, DIRMAINT migrates the entire CP user source file (or source directory) from its current format to cluster format. In cluster format, user definitions reside in cluster files. Each cluster file may contain several user definitions. The number of users in one cluster file depends on the size of the individual user definitions.

User definitions may also exist in stand-alone DIRMPART files. Profile definitions exist *only* in stand-alone DIRMPART files. Unlike a cluster file, a DIRMPART file contains only one definition.

The support for DIRMAINT Release 4 changes the DIRECTORY control statement and introduces the LOAD control statement. Also, for VM/SP HPO installations migrating to VM/XA SP, DIAGNOSE code X'84' processing will now accept and ignore any VM/XA SP-unique operations and parameters.

Note: The stand-alone module IPL DIR provided with this support does *not* support cluster format.

Security and Integrity Enhancements

1. You can now audit CP commands, DIAGNOSE functions, and virtual machine communication more than ever before.

   Note: This enhancement requires RACF 1.8.2 or the equivalent.

2. Using an external security manager, like RACF/VM, you can now prevent the DIAL command from being used at all and the MESSAGE command from being issued before LOGON.

   Note: This enhancement requires RACF 1.8.2 or the equivalent.

3. You are now assured that any data you leave behind on a T-disk is never seen by the next user of that disk.

4. CP now tests to see if you are authorized to access the storage space involved in three particular functions.

Alternate VSAM

This enhancement provides a means for OS VSAM request processing through an alternate VSAM emulator.

The alternate emulator is selected using the FILEDEF command. VSE/VSAM runs in the CMS/DOS environment and requires the use of the DLBL command instead of FILEDEF.

When a VSAM file is defined, CMS establishes a connection to the selected alternate VSAM emulator. This connection remains while there are filedefs for VSAM files using the emulator. When these filedefs are cleared, the connection to the alternate emulator is dropped and subsequent VSAM functions are again directed to VSE/VSAM.

Alternate VSAM emulators are supported as nucleus extensions. Also, only one VSAM emulator can be active at a time.

For more information, refer to the *VM/SP Application Development Guide.*

PR/SM Support

PR/SM supports logically partitioned mode (LPAR) which allows the operator to define the resources that are to be allocated to each logical partition of a 3090 Enhanced Processor complex. LPAR allows VM/SP HPO to run in two logical partitions using PR/SM.

Please see page 18 for a list of processors and configurations supported.

Enhanced VSE Support

Preferred machine assist allows a VSE/SP or VSE/AF guest to run under VM/SP HPO in supervisor state with direct control of its own I/O operations.

Preferred machine assist on VM/SP HPO, VSE/SP 2.1 or later releases can now operate with the preferred machine assist hardware feature. With this support it is possible to run a VSE/SP guest machine with a mode=370 supervisor supporting up to 40MB of virtual storage in a preferred mode together with V=R. Preferred machine assist provides performance improvements when running VSE/SP 2.1.7 or later releases as a preferred machine assist-supported guest on VM/SP HPO.

Enhanced Dispatcher Request Queue Management

This enhancement reduces system lock utilization by making the scheduler run without the system lock. This reduces the chance that a processor will hold the system lock when it becomes non-runnable. This enhancement is particularly important in a PR/SM environment where CP can run in two logical partitions.

Reduced Checkpoint I/O Operations

The number of checkpoint operations has been reduced.

Chapter 2. Installation and Service

VM/SP HPO enhancements to installation and service include:

- HUTILITY EXEC
- HITASK EXEC
- HPOINST EXEC
- The VM/SP HPO product parameter file
- The VM/SP HPO product parameter override file for the NLS feature.

HUTILITY

The HUTILITY EXEC performs the same function as UTILITY EXEC, but recognizes the VM/SP HPO product parameter file (5664-173E).

HITASK

HITASK EXEC performs the same function as the ITASK EXEC. In addition, it recognizes 4-digit real addresses.

HPOINST EXEC

The VM/SP HPO INSTALL EXEC loads down the VM/SP HPO product tape. The prompts have been reworded and you are not prompted for the minidisk addresses.

Additional updates to the VM/SP HPO installation procedure are discussed in Chapter 9.

VM/SP HPO Product Parameter File for VM/SP HPO

The product parameter file for VM/SP HPO (5664173H $PPF) is similar to the VM/SP product parameter file (5664167E $PPF), but only provides data for the CP component.

VM/SP HPO Product Parameter Override File for the NLS Feature

The VM/SP HPO product parameter override file for the NLS feature is similar to the VM/SP file, but only provides updates for the CP component. A sample file for the Kanji product parameter override file is included in the *VM/SP HPO Service Guide*.

Chapter 3. System Requirements and Planning Information

This chapter briefly describes the following system requirements for VM/SP HPO:

- Storage requirements
- Processors supported
- Processor requirements
- Devices supported
- Configuration information
- Program requirements.

Storage Requirements

VM/SP HPO requires a minimum of two megabytes of available processor storage. Mixed-mode environments – for example, CMS running concurrently with another virtual machine operating system — require an additional 512KB of available processor storage. The resident nucleus size for VM/SP HPO Release 6 is 370KB (an additional 40KB of real storage is allocated in AP or MP mode).

Note: The nucleus size approximation for VM/SP HPO is for planning purposes only and may vary depending upon your installation's configuration.

Processors Supported by VM/SP HPO

VM/SP High Performance Option operates on the following processors:

Processor	Model No.	Configuration
303x	3031	UP/AP
	3032	UP
	3033	UP/AP/MP
3042	3042-2	AP
308x	3081	Dyadic
	3083	UP
	3084	Physically Partitioned
3090	150	UP
	180	UP
	200	Dyadic
	400	Physically Partitioned
	120E	UP, LPAR
	150E	UP, LPAR
	180E	UP, LPAR
	200E	Dyadic, LPAR
	280E	Physically Partitioned, LPAR
	300E	LPAR
	400E	LPAR
	500E	LPAR
	600E	LPAR
	100S	UP, LPAR
	120S	UP, LPAR
	150S	UP, LPAR
	170S	UP, LPAR
	180S	Dual
	200S	Dyadic, LPAR
	250S	Physically Partitioned, LPAR
	280S	Physically Partitioned, LPAR
	300S	LPAR
	380S	Physically Partitioned, LPAR
	400S	Physically Partitioned, LPAR
	500S	LPAR
	600S	LPAR

Processor	Model No.	Configuration
43xx	4341 (All models)	UP
	4381-1	UP
	4381-2	UP
	4381-3	Dual
	4381-11	UP
	4381-12	UP
	4381-13	UP
	4381-14	Dual
	4381-21	UP
	4381-22	UP
	4381-23	UP
	4381-24	Dual
	4381-90E	UP
	4381-91E	UP
	4381-92E	Dual

Notes:

1. See the glossary in the back of this book for descriptions of UP (uniprocessor), AP (attached processor), MP (multiprocessor), dyadic, dual, partitioned processing modes, logically partitioned processing (LPAR), and PR/SM.
2. The references to the 3081, 3083, and 3084 include all models of these processors.

Processor Requirements

Certain processors require additional hardware in order to support specific VM/SP HPO functions.

1. On the 3031 UP/AP processors, the EC level for shadow table bypass assist (a feature of the shadow table maintenance facility) is 388860 and later.
2. For the IBM 3033 processor:
   - Preferred machine assist requires the following:
     - Preferred Machine Assist RPQ (#8P0913).
     - Virtual Machine Assist RPQ (#EJ1156).
     - 3033 Extension feature (#6850) with Extended Control Storage (#3868) on each channel group.
     - Extended Addressing feature (#3832).

     Note: These features are required for the IPL processor of an AP or MP configured system.
   - Extended storage support requires the Extended Addressing feature (#3832). Also, the virtual machine assist feature must be at EC level 209811 or later.
   - MVS/SP microcode assist cross memory services requires the following:
     - Virtual Machine Assist RPQ (#EJ1156)
     - 3033 Extension feature (#6850) with Extended Control Storage (#3868) on each channel group.
   - 3033 Extension Feature Enhancement to Virtual Machine Assist RPQ (#EJ1156) requires the 3033 Extension feature (#6850). Note that these features are required for the IPL processor of an AP or MP configured system.
   - The EC level for shadow table bypass assist is 212205.
3. For the IBM 308x, the control switch assist for preferred machine assist requires the following engineering change (EC) levels:

Processor	EC Level
3081D	209917
3081K	207224
3081KX	209956C
3081G	212012
3081GX	209996B
3083B	212046
3083BX	209986A
3083E	214336
3083EX	209966A
3083J	208301
3083JX	209976B
3084Q	216745
3084QX	207231A/B
4381-1	993079
4381-2	866905
4381-3	329510
4381-11	A32141
4381-12	A35026
4381-13	A09749
4381-14	A09805

Full support is provided without additional engineering change for 3083CX, 4381 models (21, 22, 23, 24, 90E, 91E, 92E), and for all 3090s.

Devices Supported by VM/SP HPO

VM/SP HPO supports most of the currently available direct access storage devices, magnetic tapes, unit record devices, terminals, and transmission control units. The following sections list the devices supported by VM/SP HPO. The devices are listed by device type:

- Direct access storage devices
- Magnetic tapes
- Unit record devices (printers, readers, and punches)
- Terminals
- Transmission control units and communication controllers
- Multisystem communication units
- Remote Spooling Devices
- Other Devices (your IBM marketing representative can provide you with a complete list of devices supported by VM/SP HPO)

Warning: Certain devices no longer supported are still mentioned elsewhere in the VM/SP HPO library because the code still refers to them. Use any devices not listed here or in Chapter 2 of the *VM/SP HPO Planning Guide and Reference* at your own risk.

Direct Access Storage Devices

The direct access storage devices supported by VM/SP HPO are:

Storage Device	Model No.
2305 Fixed Head	1 and 2
3310 Direct Access	–
3330 Disk	1, 2, and 11
3333 Disk and Control	1 and 11
3340 Direct Access[1]	A2, B1, and B2
3350 Direct Access	A2 and B2
3370 Direct Access	A1, A2, B1, and B2
3375 Direct Access	–
3380 Direct Access	AA4, AD4, AE4, B4, and BE4

Notes:

1. All of these direct access devices are supported as dedicated devices.
2. The 2305 is not supported by CMS or on a 3090.
3. The 2305, 3310, 3330, 3340, 3350, 3370, 3375, 3380, and FB-512 devices are supported as system residence devices. (However, VM/SP HPO Release 6 only provides starter systems on 3350s, 3375s, and 3380s.)

For more information on devices that support paging, swapping, and spooling, see the *VM/SP HPO Planning Guide and Reference*.

[1] 3348 Data Modules Models 35, 70, and 70F; and the 3344 Direct Access Storage Model B2.

Direct Access Storage Control Units

The following direct access control units are supported by VM/SP HPO:

Control Unit	Model No.	Storage Device	Model No.
2835	1 and 2	2305	1 and 2
3830	1	3330	1 and 2
3830	2	3333 3340 3350	1 and 11 A2 A2
3830	3	3330 3333 3350	1 and 11 1 and 11 A2 and B2
3880	1	3330 3333 3340 3350 3370 3375	1, 2, and 11 1 and 11 A2 A2 and A2F A1, A2, B1, and B2 —
3880	2,11,21	3330 3333 3340 3350 3370 3375	1, 2, and 11 1 and 11 A2 A2 and A2F A1, A2, B1, and B2 All except A04
	13,23	3380[2]	—
3990	1,2,3	3380	—

Magnetic Tapes

VM/SP HPO supports the following magnetic tape devices:

Device	Model No.
2401,2402,2403,and 2404	1,2,3,4, and 5
3410/3411	1, 2, and 3
3411 Tape Unit and Control	1, 2, and 3
3420	3, 4, 5, 6, 7, and 8
3422 Tape Unit and Control	–
3424 Tape Unit and Control	–
3430	–
3430 Tape Unit and Control	–
3480	–
8809 Tape Unit and Control	–

[2] For the 303x processor, you must have the Data Streaming Feature (No. 4850).

Magnetic Tape Control Units

The magnetic tape control units supported by VM/SP HPO are:

Tape Control Unit
2803
2804
3411 Unit and Control
3422 Unit and Control
3424 Unit and Control
3430 Unit and Control
3480
3803
8809 Unit and Control

Unit Record Devices (Printers, Readers, and Punches)

VM/SP HPO supports the following printers, readers, punches, and unit record control units as system spool devices.

Printers

VM/SP HPO supports the following printers:

Printer	Model No.
1403	2, 3, 7, and N1
1443	N1 (with 144 print positions)
3203	5
3211	(Right indexing only)
3213	(in 3215 emulator mode)
3287	1, 1C, 2, and 2C
3289	4
3800	1, 3, and 8
3812	—
3820	—
4245	—
4248	1
4250	(dedicated only)
5210	—
6262	—

Card Readers and Card Punches

VM/SP HPO supports the following readers and punches:

Device	Model No.
2501 Card Reader	B1 and B2
2520 Card Punch	B2 and B3
2540 Card Read Punch	1
3505 Card Reader	B1 and B2
3525 Card Punch	P1, P2, and P3

Unit Record Control Units

VM/SP HPO supports the following unit record control units:

- IBM 2821 Control Unit
- IBM 3811 Printer Control Unit

High-Function Graphic Devices

VM/SP HPO supports the following high-function graphic devices:

- IBM 3250 Models 1 and 2

Terminals and Display Stations

VM/SP HPO supports the following system consoles and terminals: (in 3215 emulation mode or in 3270 mode only)

Device	Model
PC/AT/370	All models
PC/XT/370	All models
3275	2
3276	2, 3, and 4
3180	via 3276 2,3,and 4
3277	2,3 and 4
3278	2, 3, 4, and 5
3279 (color)	2A, 2B, 3A, and 3B
3290 Info Panel	via 3274 (all models)

Transmission Control Units

VM/SP HPO supports the following transmission control units:

Device	Unit Type or Model
2701	Data type
2702	Transmission control
2703	Transmission control
Integrated Communications Attachment (ICA)	No. 4060
3704, 3705-I, 3705-II, 3725	Communication controllers

Multisystem Communication Units

VM/SP HPO supports all models of the 3088 Multisystem Communications Unit. The 3088 Model 1 interconnects up to four systems; the 3088 Model 2 interconnects up to eight systems.

Minimum Configuration

VM/SP HPO requires:

- One of the processors previously listed with at least 2 MB of available processor storage.
- One system control device.
- One printer
- One card reader.[3]
- One card punch.[3]
- Two spindles of direct access storage of the same type
- One 9-track magnetic tape unit or one 3480 cartridge tape unit.
- One multiplexer channel.
- One selector or block multiplexer channel.

Note: The configuration shown is an absolute minimum. Support of large numbers of CMS users requires larger configurations. Consult your marketing team for the appropriate configuration for your environment.

Configurations Supported by CMS

VM/SP HPO does not extend the capabilities of the CMS component of VM/SP. The following CMS configurations apply only to the CMS component of VM/SP.

- The minimum virtual storage size for CMS is 1 MB. Virtual storage can be as large as 16 MB added in multiples of 4K.
- CMS can use as a virtual operator's console any terminal supported by VM/SP HPO.
- CMS can use as a spooling device any virtual (nondedicated) card reader, card punch (except the 2520), and printer supported by VM/SP HPO.
- CMS supports up to 26 logical 3330, 3340, Model 1,2, or 11, 3333 Model 1 or 11, 3350, 3375, 3380 or FB-512 direct access storage devices (DASDs).
- CMS supports up to 16 2400, 2415, 3410 (9-track only), or 3420 (7- or 9-track), 3422, 3430, 3480 (18 track), or 8809 (9 track only) magnetic tape units. In addition to the preceding information, CMS users may not be authorized to use DASD space within CMS Shared File System (SFS) file pools.

Program Requirements

VM/SP HPO executes with VM/SP. VM/SP HPO Release 6 requires installation of VM/SP Release 6 for additional non-CP components. When you merge the prerequisite VM/SP licensed program with the appropriate VM/SP HPO program, you obtain a functional operating system that integrates the functions of the VM/SP base product and the added features of the VM/SP HPO.

Your VM/SP system must be at a certain service level prior to your installing VM/SP HPO, or at a lower service level than your VM/SP HPO 6 system. Refer to the Program Directory for information describing the required VM/SP service level.

[3] This device is not required for a cardless system.

Language

VM/SP HPO source code is written in basic assembler language.

Licensed Programs That Operate with VM/SP HPO Release 6

The following are licensed programs that you may require or find useful for your VM/SP HPO system. Contact your IBM marketing representative for the specific release levels of these and other programs from which your installation may benefit.

- To operate VM/SP HPO Release 6, you must have:
  - VM/System Product (VM/SP) Release 6, Program Number 5664-167
- For Vector Facility, you may require:
  - Engineering and Scientific Subroutine Library (ESSL), Program Number 5668-863
  - VS FORTRAN Version 2 Compiler, Library and Interactive Debug, Program Number 5668-806
  - Assembler H Vector Facility Enhancement, Program Number 5668-962
- To use the transparent services access facility component of VM, you may need:
  - SQL/DS Release 3.5 with Remote Relational Access Support, Program Number 5748-XXJ
- For a VM/SNA network, you need:
  - Advanced Communications Function/Virtual Telecommunications Access Method (ACF/VTAM), Program Number 5664-280
  - Advanced Communications Function/Network Control Program (ACF/NCP), Program Number 5668-854
  - Advanced Communications Function/System Support Programs (ACF/SSP), Program Number 5664-289
- In addition, the following products are recommended for a communication network:
  - NetView™, Program Number 5664-204 (this requires ACF/VTAM Version 3 Release 1.1)
  - RSCS Networking Version 2, Program Number 5664-188
  - VM/Pass-Through Facility (PVM), Program Number 5748-RC1
- For system support, you may find the following helpful:
  - VMBACKUP Management System, Program Number 5664-291
  - VMTAPE Management System, Program Number 5664-292
  - Resource Access Control Facility/VM (RACF/VM), Program Number 5740-XXH
  - Virtual Machine/Directory Maintenance Licensed Product (DIRMAINT), Program Number 5748-XE4
  - VM Batch Facility, Program Number 5664-364
  - VM/Interactive Productivity Facility (VM/IPF), Program Number 5664-318
  - Environmental Recording, Editing, and Printing (EREP), Program Number 5654-260

Netview is a trademark of the International Business Machines Corporation.

- To aid in evaluating performance, you may use:
  - Virtual Machine Monitor Analysis Program (VMMAP), Program Number 5664-191
  - Virtual Machine Real Time Monitor (VM/RTM), Program Number 5796-PNA
  - Virtual Machine Performance Planning Facility (VMPPF), Program Number 5664-179
- For business and text office support, you might consider:
  - Professional Office System (PROFS), Program Number 5664-309
  - Document Composition Facility, Program Number 5748-XX9
  - Display Write/370, Program Number 5664-370
- For printer support, you may use:
  - Print Services Facility/Virtual Machine (PSF/VM), Program Number 5664-198
  - Print Services Access Facility/Virtual Machine (PSAF/VM), Program Number 5664-312.

Compatibility

Application programs that currently execute using VM/SP that do not depend on internal CP structure and/or control blocks should continue to execute using VM/SP HPO.

Chapter 4. VM/SP HPO Publications

This chapter:

- Discusses ordering procedures
- Shows the changes to the library for Release 6
- Lists the books you receive with the product tape
- Gives instructions for ordering additional copies of publications
- Discusses the documentation for between-release support
- Provides abstracts of the manuals.

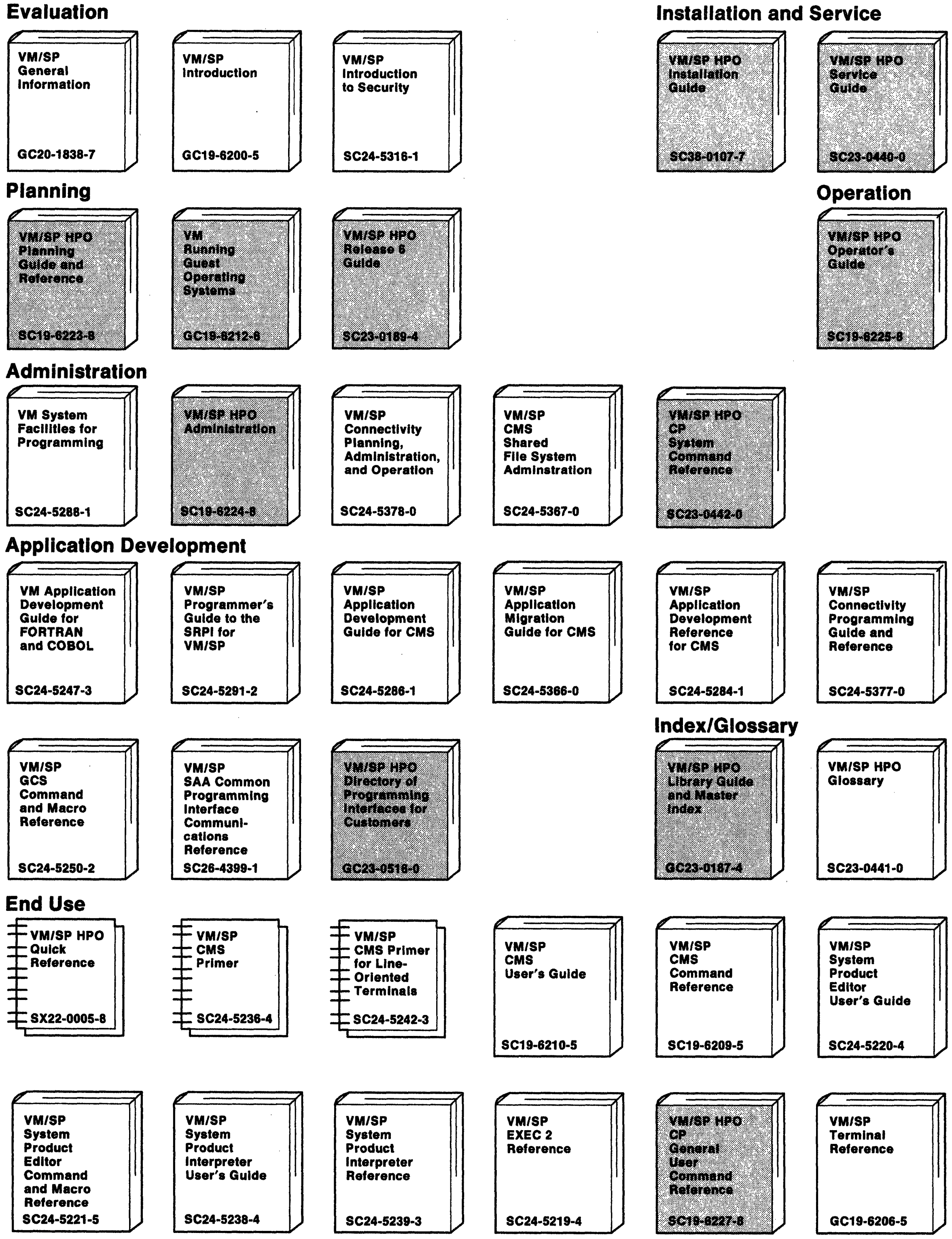
Ordering Release 6 Publications

Continuing with Release 6 of VM/SP HPO, certain manuals (entitled manuals) in the library will be shipped with the product tape, and automatic updates (new releases and technical newsletters) will be sent automatically. These VM/SP HPO entitled manuals are shaded on the two-page library organization chart beginning on page 30.

The manuals in the *Diagnosis* category of the library diagram whose order numbers begin with an L use-key are shipped automatically. Licensed users can also order additional manuals by specifying feature numbers. There will be a charge for these. You can also order individual use-key L manuals to be shipped with the product by ordering their individual feature codes.

The set which you will receive automatically does not include all the books that are available. You can order the other manuals in the Release 6 VM/SP HPO library, or multiple copies of any manual, through your IBM marketing representative. Use the order number, *without* the suffix, of the book(s) you need. This will ensure that you get the most current version.

You can also *subscribe* to these other manuals, or to multiple copies of any manual, by using the System Library Subscription Service (SLSS). See "System Library Subscription Service (SLSS)" on page 52 for more information about SLSS.


Evaluation
VM/SP General Information GC20-1838-7
VM/SP Introduction GC19-6200-5
VM/SP Introduction to Security SC24-5316-1
Installation and Service
VM/SP HPO Installation Guide SC38-0107-7
VM/SP HPO Service Guide SC23-0440-0
Planning
VM/SP HPO Planning Guide and Reference SC19-6223-8
VM Running Guest Operating Systems GC19-6212-8
VM/SP HPO Release 6 Guide SC23-0189-4
Operation
VM/SP HPO Operator's Guide SC19-6225-8
Administration
VM System Facilities for Programming SC24-5288-1
VM/SP HPO Administration SC19-6224-8
VM/SP Connectivity Planning, Administration, and Operation SC24-5378-0
VM/SP CMS Shared File System Administration SC24-5367-0
VM/SP HPO CP System Command Reference SC23-0442-0
Application Development
VM Application Development Guide for FORTRAN and COBOL SC24-5247-3
VM/SP Programmer's Guide to the SRPI for VM/SP SC24-5291-2
VM/SP Application Development Guide for CMS SC24-5286-1
VM/SP Application Migration Guide for CMS SC24-5366-0
VM/SP Application Development Reference for CMS SC24-5284-1
VM/SP Connectivity Programming Guide and Reference SC24-5377-0
VM/SP GCS Command and Macro Reference SC24-5250-2
VM/SP SAA Common Programming Interface Communi-cations Reference SC26-4399-1
VM/SP HPO Directory of Programming Interfaces for Customers GC23-0516-0
Index/Glossary
VM/SP HPO Library Guide and Master Index GC23-0187-4
VM/SP HPO Glossary SC23-0441-0
End Use
VM/SP HPO Quick Reference SX22-0005-8
VM/SP CMS Primer SC24-5236-4
VM/SP CMS Primer for Line-Oriented Terminals SC24-5242-3
VM/SP CMS User's Guide SC19-6210-5
VM/SP CMS Command Reference SC19-6209-5
VM/SP System Product Editor User's Guide SC24-5220-4
VM/SP System Product Editor Command and Macro Reference SC24-5221-5
VM/SP System Product Interpreter User's Guide SC24-5238-4
VM/SP System Product Interpreter Reference SC24-5239-3
VM/SP EXEC 2 Reference SC24-5219-4
VM/SP HPO CP General User Command Reference SC19-6227-8
VM/SP Terminal Reference GC19-6206-5

▨ one copy of each shaded manual automatically received with the VM/SP HPO product tape

Figure 1 (Part 1 of 2). The VM/SP HPO and Related Libraries

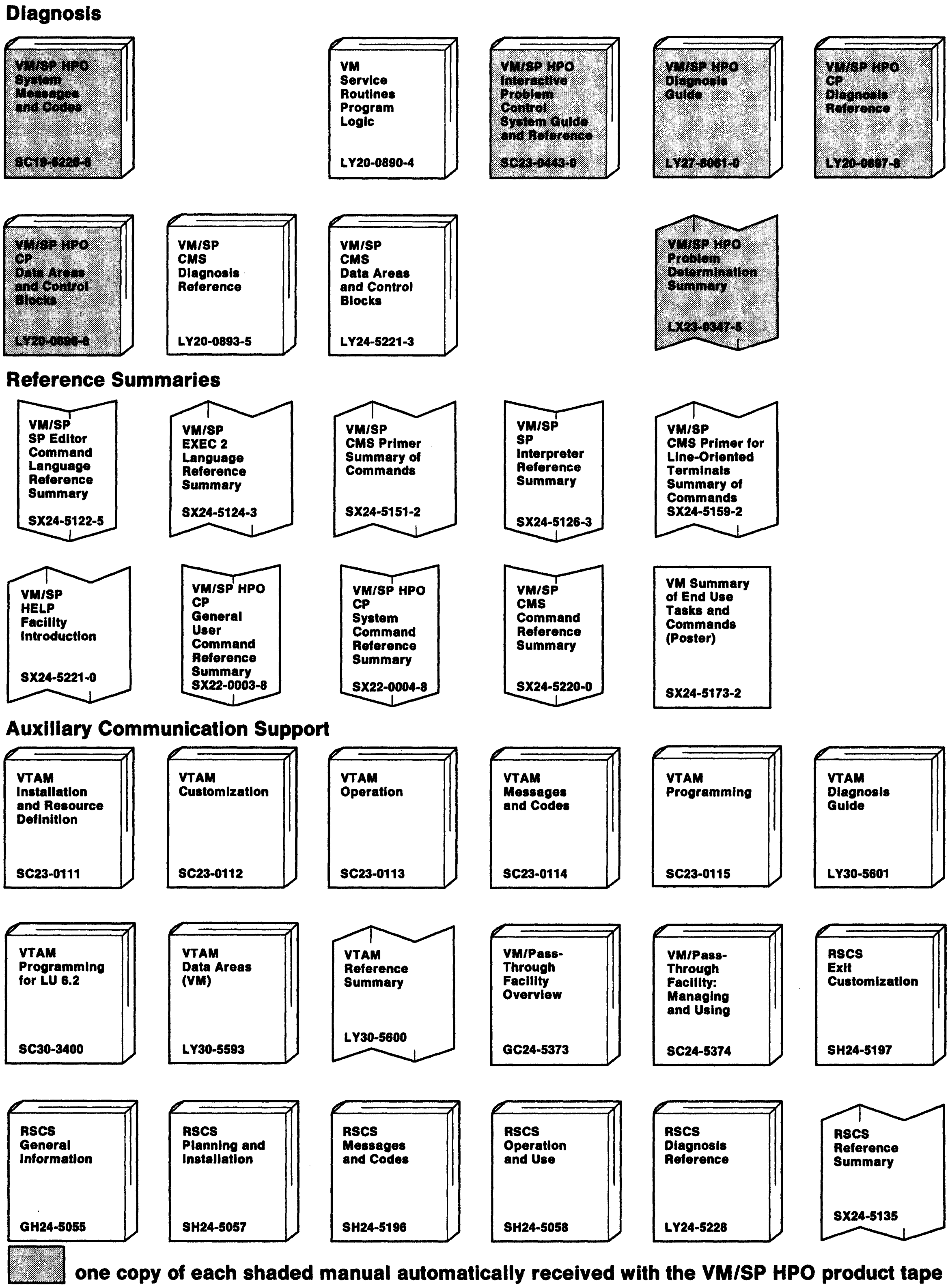

Diagnosis
VM/SP HPO System Messages and Codes SC19-6226-6
VM Service Routines Program Logic LY20-0890-4
VM/SP HPO Interactive Problem Control System Guide and Reference SC23-0443-0
VM/SP HPO Diagnosis Guide LY27-8061-0
VM/SP HPO CP Diagnosis Reference LY20-0897-8
VM/SP HPO CP Data Areas and Control Blocks LY20-0896-8
VM/SP CMS Diagnosis Reference LY20-0893-5
VM/SP CMS Data Areas and Control Blocks LY24-5221-3
VM/SP HPO Problem Determination Summary LX23-0347-5
Reference Summaries
VM/SP SP Editor Command Language Reference Summary SX24-5122-5
VM/SP EXEC 2 Language Reference Summary SX24-5124-3
VM/SP CMS Primer Summary of Commands SX24-5151-2
VM/SP SP Interpreter Reference Summary SX24-5126-3
VM/SP CMS Primer for Line-Oriented Terminals Summary of Commands SX24-5159-2
VM/SP HELP Facility Introduction SX24-5221-0
VM/SP HPO CP General User Command Reference Summary SX22-0003-8
VM/SP HPO CP System Command Reference Summary SX22-0004-8
VM/SP CMS Command Reference Summary SX24-5220-0
VM Summary of End Use Tasks and Commands (Poster) SX24-5173-2
Auxiliary Communication Support
VTAM Installation and Resource Definition SC23-0111
VTAM Customization SC23-0112
VTAM Operation SC23-0113
VTAM Messages and Codes SC23-0114
VTAM Programming SC23-0115
VTAM Diagnosis Guide LY30-5601
VTAM Programming for LU 6.2 SC30-3400
VTAM Data Areas (VM) LY30-5593
VTAM Reference Summary LY30-5600
VM/Pass-Through Facility Overview GC24-5373
VM/Pass-Through Facility: Managing and Using SC24-5374
RSCS Exit Customization SH24-5197
RSCS General Information GH24-5055
RSCS Planning and Installation SH24-5057
RSCS Messages and Codes SH24-5196
RSCS Operation and Use SH24-5058
RSCS Diagnosis Reference LY24-5228
RSCS Reference Summary SX24-5135
one copy of each shaded manual automatically received with the VM/SP HPO product tape

Figure 1 (Part 2 of 2). The VM/SP HPO and Related Libraries

What Are Some of the Changes in the Release 6 Library?

The structure of the VM Library is changed for VM/SP Release 6 and VM/SP HPO Release 6. The information is reorganized so that each book relates more closely to a specific task. Some publications have been eliminated, and the information is provided in new or reorganized publications. In addition, information is added for the new support in Release 6.

What Publications Have Split?

The following publications split into two books:

- *VM/SP HPO CP Command Reference* is now:
  - *VM/SP HPO CP General User Command Reference*
  - *VM/SP HPO CP System Command Reference*
- *VM/SP HPO Library Guide, Glossary, and Master Index* is now:
  - *VM/SP HPO Library Guide and Master Index*
  - *VM/SP HPO Glossary*
- *VM/SP HPO Installation Guide* is now:
  - *VM/SP HPO Installation Guide*
  - *VM/SP HPO Service Guide*

What Publications Are Renamed?

The titles of the following books have changed:

- *VM/SP HPO CP for System Programming* is now *VM/SP HPO Administration*
- *VM/SP HPO CP Command Reference* is now *VM/SP HPO CP General User Command Reference*
- *VM/SP HPO System Logic and Problem Determination Guide—CP* is now *VM/SP HPO CP Diagnosis Reference*
- *VM/SP HPO Data Areas and Control Block Logic—CP* is now *VM/SP HPO CP Data Areas and Control Blocks*
- *VM/SP HPO Service Routines Program Logic* is now *VM Service Routines Program Logic.*

What Publications Are New?

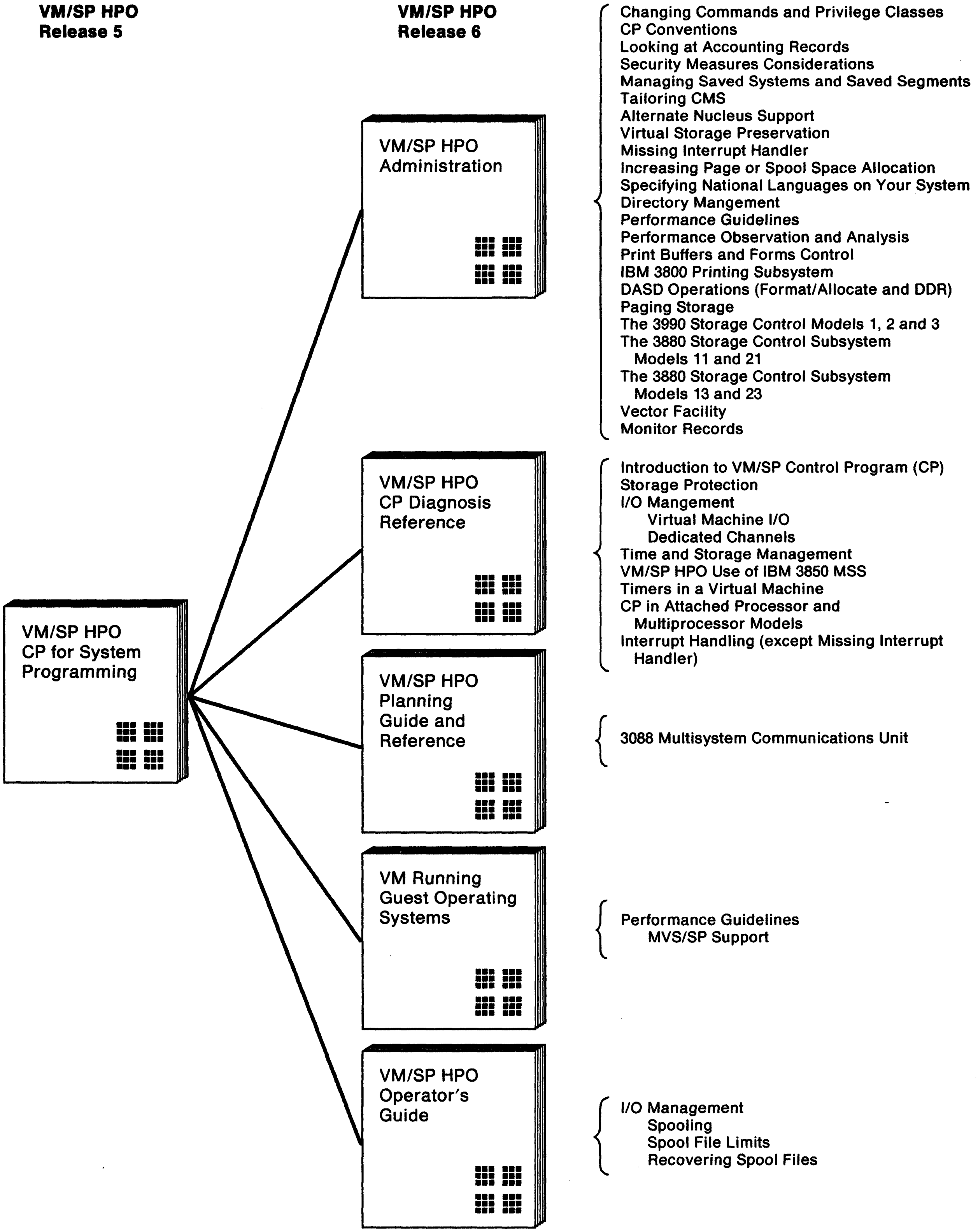
Some new titles that you will notice in this release are:

- *VM/SP HPO CP System Command Reference*
- *VM/SP HPO Diagnosis Guide*
- *VM/SP HPO Interactive Problem Control System Guide and Reference*
- *VM/SP HPO Glossary*
- *VM/SP HPO Service Guide*
- *VM/SP HPO Directory of Programming Interfaces for Customers.*

Refer to the following pages to find out how these publications fit into the structure of the library and to the "Abstracts of Manuals" on page 55 for descriptions of their contents.

CP for System Programming Is Renamed and Reorganized

You can now find the information that was in the *CP for System Programming* in the following Release 6 publications:



VM/SP HPO Release 5

VM/SP HPO CP for System Programming

VM/SP HPO Release 6

VM/SP HPO Administration

- Changing Commands and Privilege Classes
- CP Conventions
- Looking at Accounting Records
- Security Measures Considerations
- Managing Saved Systems and Saved Segments
- Tailoring CMS
- Alternate Nucleus Support
- Virtual Storage Preservation
- Missing Interrupt Handler
- Increasing Page or Spool Space Allocation
- Specifying National Languages on Your System
- Directory Mangement
- Performance Guidelines
- Performance Observation and Analysis
- Print Buffers and Forms Control
- IBM 3800 Printing Subsystem
- DASD Operations (Format/Allocate and DDR)
- Paging Storage
- The 3990 Storage Control Models 1, 2 and 3
- The 3880 Storage Control Subsystem Models 11 and 21
- The 3880 Storage Control Subsystem Models 13 and 23
- Vector Facility
- Monitor Records

VM/SP HPO CP Diagnosis Reference

- Introduction to VM/SP Control Program (CP)
- Storage Protection
- I/O Mangement
  - Virtual Machine I/O
  - Dedicated Channels
- Time and Storage Management
- VM/SP HPO Use of IBM 3850 MSS
- Timers in a Virtual Machine
- CP in Attached Processor and Multiprocessor Models
- Interrupt Handling (except Missing Interrupt Handler)

VM/SP HPO Planning Guide and Reference

- 3088 Multisystem Communications Unit

VM Running Guest Operating Systems

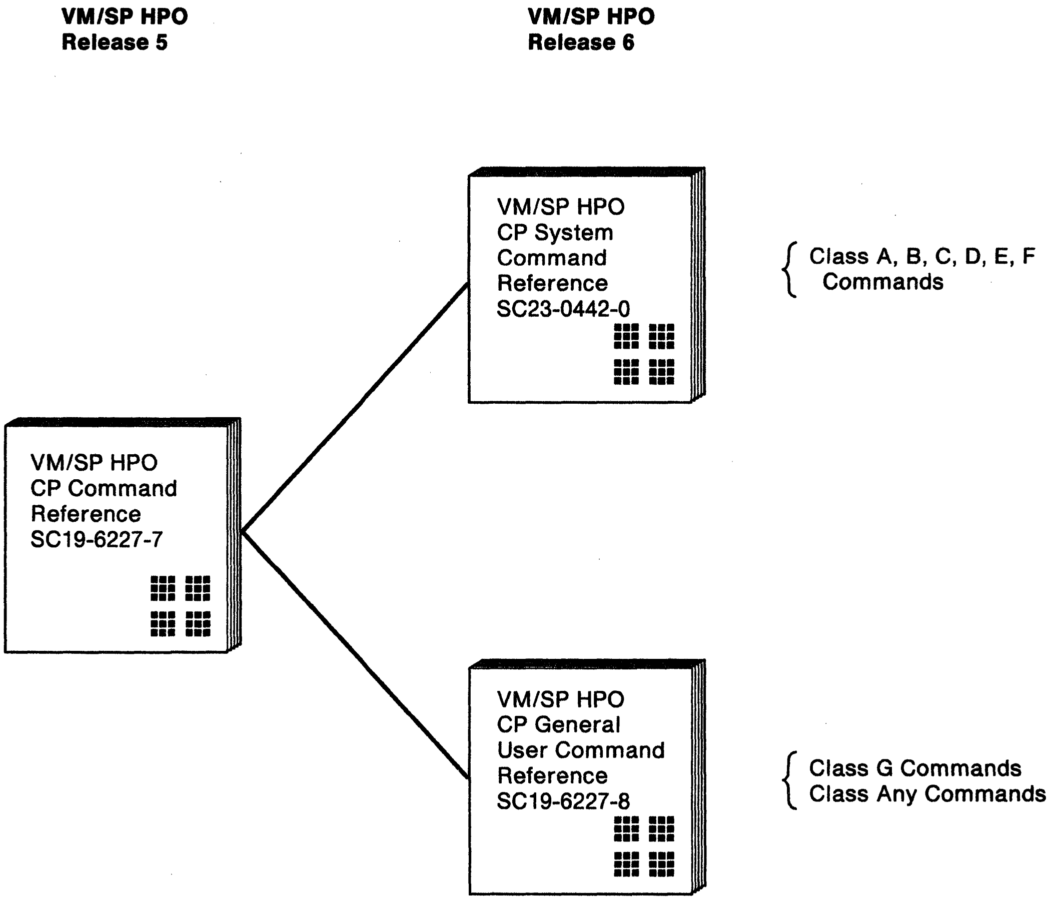
- Performance Guidelines
  - MVS/SP Support

VM/SP HPO Operator's Guide

- I/O Management
  - Spooling
  - Spool File Limits
  - Recovering Spool Files

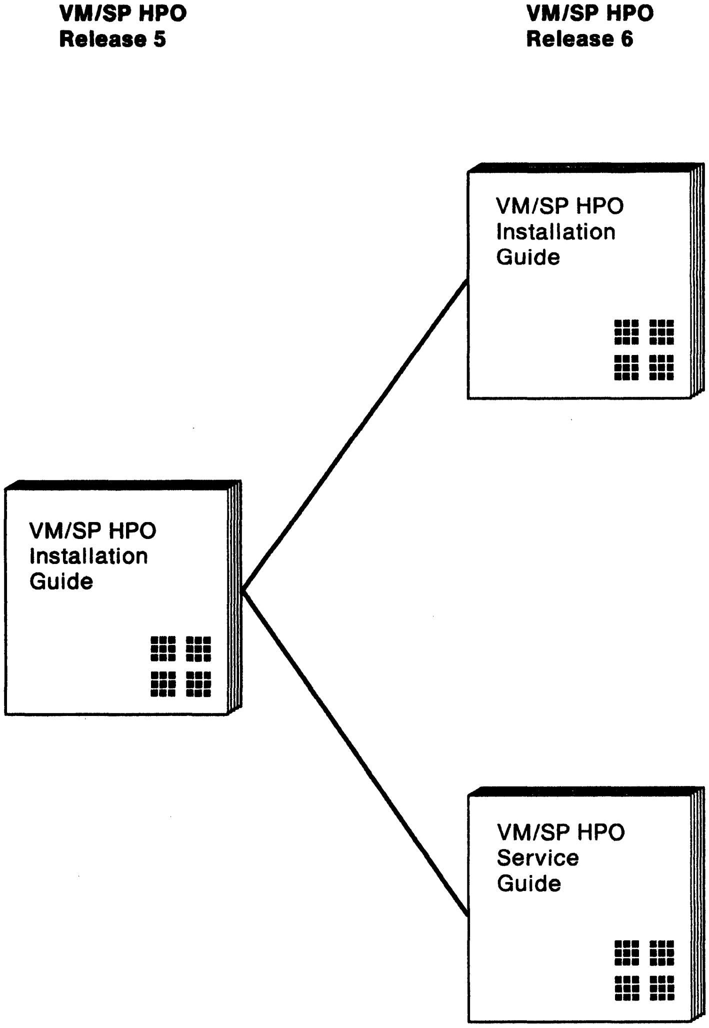
CP Command Reference is Split

You can now find the information that was in the *CP Command Reference* in the following Release 6 publications:



Installation Guide is Split

You can now find the information that was in the *Installation Guide* in the following Release 6 publications:

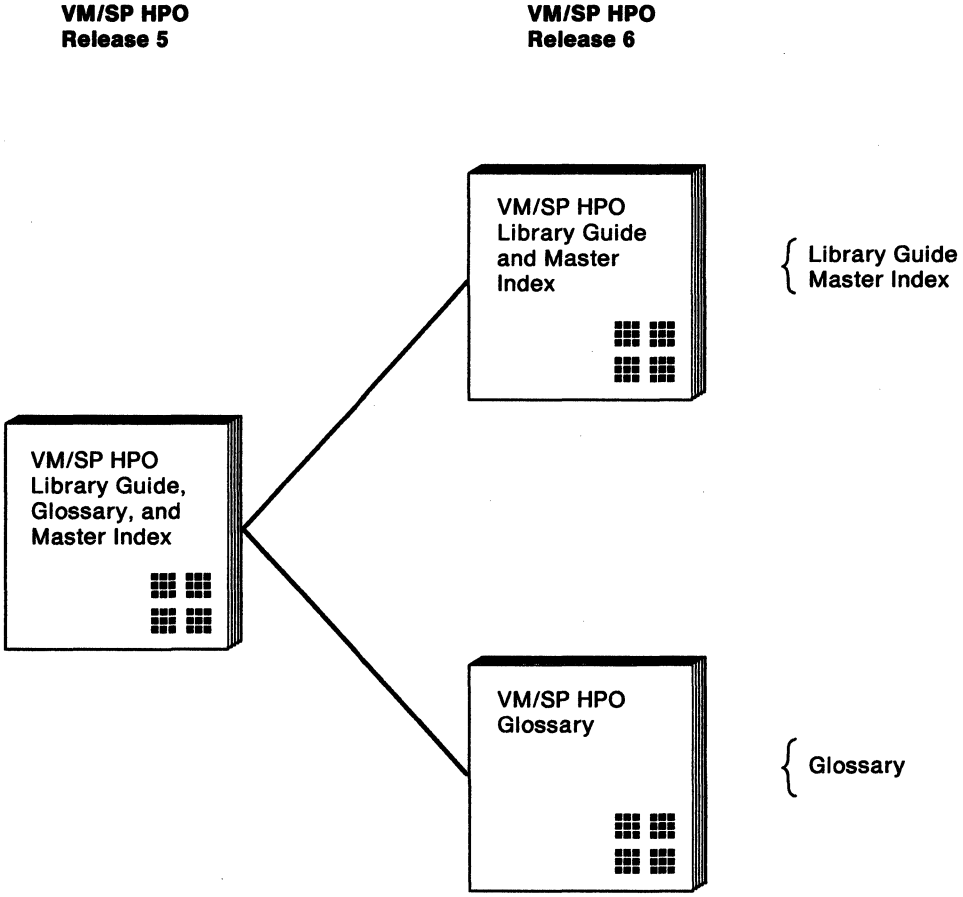


Introduction
Planning Your VM/SP HPO Installation
Installing VM/SP HPO Using a Starter System
Installing VM/SP HPO Using an Existing System
Verifying Your Installed VM/SP HPO System
Installing Saved Segments
Installing a New System National Language
Tools
Minidisks and SFS Directories Reserved for the MAINT Userid
Restricted Logon Passwords
Sample SPLOAD PROFILE
Special Options for CP
Migrating Spool Files Using SHUTDOWN/WARM IPL
Enlarging the CMS Nucleus to Contain the Y-Disk Directory (Y-STAT)
Example of Alternate CMS Nucleus Placement
Example of Alternate GCS Nucleus Placement
Considerations for the Placement of the the DOS-Related Saved Segments
Sample EXEC Procedure for Copying VSE Macros into a CMS MACLIB

Introduction to VM/SP HPO Service
Service Files
Service Control Structures
Product Parameter File
PUT/COR Service Procedure
Local Service
Tools
Sample VM/SP Product Parameter File
IBM-Supplied Control Files
Servicing SNA Products

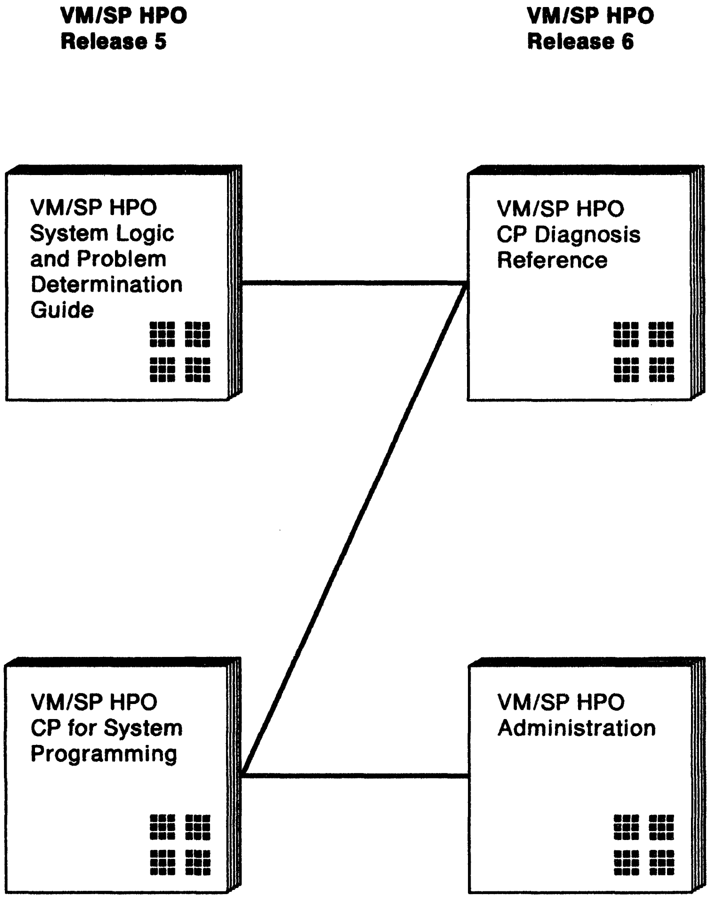
Library Guide, Glossary, and Master Index is Split

You can now find the information that was in the *Library Guide, Glossary, and Master Index* in the following Release 6 publications:


VM/SP HPO
Release 5
VM/SP HPO
Release 6
VM/SP HPO Library Guide and Master Index
Library Guide
Master Index
VM/SP HPO Library Guide, Glossary, and Master Index
VM/SP HPO Glossary
Glossary

System Logic and Problem Determination Guide Is Renamed

The *VM/SP HPO System Logic and Problem Determination Guide* is renamed for Release 6 as the *VM/SP HPO CP Diagnosis Reference* and has incorporated some material from *VM/SP HPO CP for System Programming.*

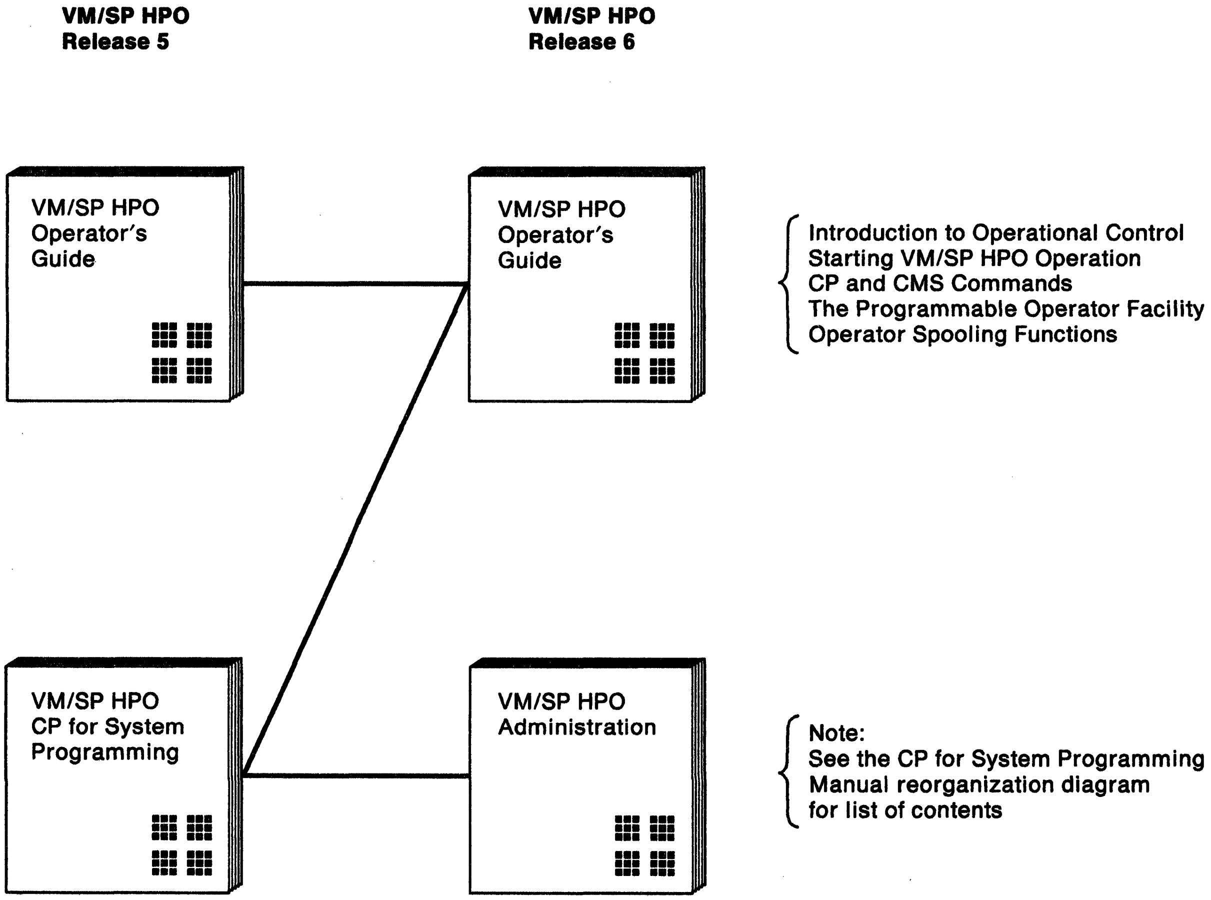

VM/SP HPO
Release 5
VM/SP HPO
Release 6
VM/SP HPO System Logic and Problem Determination Guide
VM/SP HPO CP Diagnosis Reference
VM/SP HPO CP for System Programming
VM/SP HPO Administration

Introduction to the VM/SP HPO Control Program
CP Commands and Messages
System Initialization
Timer Management
Storage Management
Free Storage Management
I/O Management
Interrupt Handling
Communication Facilities
Spool Management
Error Management
System Failure Recovery
The Attached Processor and Multiprocessor Environments
CP Annotated Flow Diagrams
CP Module Entry Point Directory
Entry Points for CP Commands
Function Codes for DIAGNOSE Instruction
Access Verification Routines
Extended Control-Program Support
VM/SP HPO Use of the IBM 3850 Mass Storage System
MVS/System Extensions/Product Support

Note:
See the CP for System Programming manual reorganization diagram for list of contents

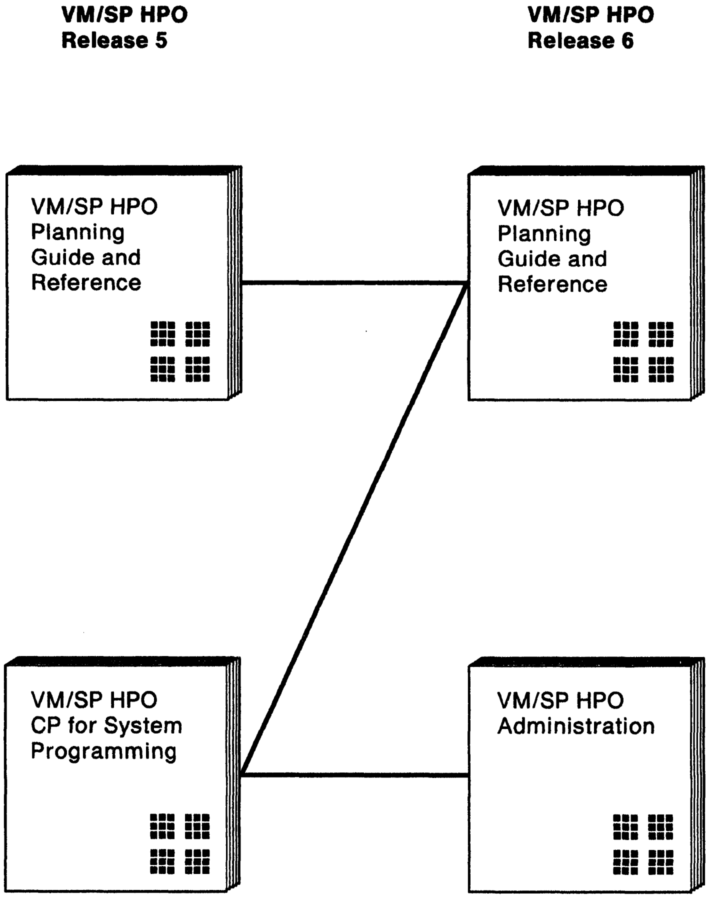
Operator's Guide is Reorganized

The *Operator's Guide* has incorporated additional information from *VM/SP HPO CP for System Programming* for Release 6.


VM/SP HPO
Release 5
VM/SP HPO
Release 6
VM/SP HPO Operator's Guide
VM/SP HPO CP for System Programming
VM/SP HPO Operator's Guide
VM/SP HPO Administration
Introduction to Operational Control
Starting VM/SP HPO Operation
CP and CMS Commands
The Programmable Operator Facility
Operator Spooling Functions
Note:
See the CP for System Programming Manual reorganization diagram for list of contents

Planning Guide and Reference is Reorganized

The *Planning Guide and Reference* has incorporated additional information from *VM/SP HPO CP for System Programming* for Release 6.

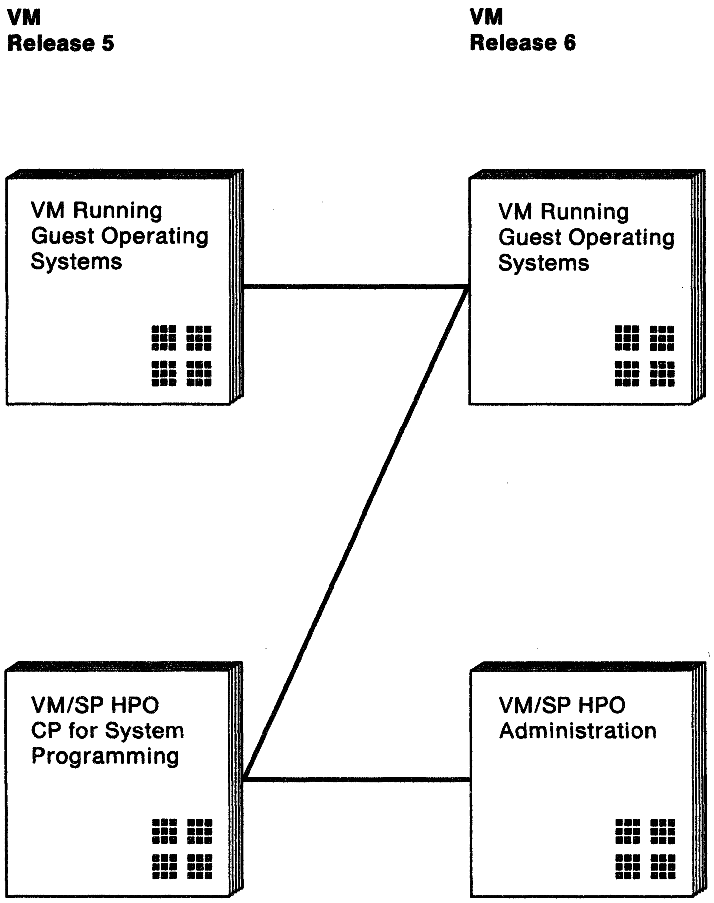

VM/SP HPO
Release 5
VM/SP HPO
Release 6
VM/SP HPO Planning Guide and Reference
VM/SP HPO CP for System Programming
VM/SP HPO Planning Guide and Reference
VM/SP HPO Administration

Introduction to Planning and VM/SP HPO
Planning Device Configurations for VM/SP HPO
Estimating CP Storage Requirements
Planning for Other VM/SP HPO Areas
Extending VM/SP HPO
Planning for I/O
Planning for Hardware Devices
Creating Your Directory
Preparing the Real I/O Configuration File (DMKRIO)
Preparing the CP System Control File (DMKSYS)
Preparing the System Name Table File (DMKSNT)
Additional System Definition Files
Configuration Aid
VM/SP HPO Restrictions
Worksheet to Aid in Coding the NAMESYS Macro
Sample SNTMAP Output

Note:
See the CP for System Programming manual reorganization diagram for list of contents

Running Guest Operating Systems is Reorganized

Running Guest Operating Systems has incorporated additional information from *VM/SP HPO CP for System Programming* for Release 6.


VM
Release 5
VM
Release 6
VM Running Guest Operating Systems
VM Running Guest Operating Systems
VM/SP HPO CP for System Programming
VM/SP HPO Administration

Introduction to Running VSE/SP Version 2 or 3 Under VM
Defining a Single VSE/SP Virtual Machine
Operating a VSE/SP Virtual Machine
VSE/SP Virtual Machines Sharing DASD
Introduction to Running MVS/SP Under VM/SP HPO
Defining a Basic MVS/SP Virtual Machine
Enhancing MVS/SP Performance Under VM/SP HPO
More About Operating an MVS/SP Virtual Machine
MVS Virtual Machines Sharing DASD
Introduction to Running VM Under VM
Defining the First and Second-Level VM Systems
Operating VM Under VM

Note:
See the CP for System Programming manual reorganization diagram for list of contents

What Publications Will I Receive with the Product Tape?

For VM/SP HPO Release 6, you will receive the following publications with the product tape:

1. *VM/SP HPO Release 6 Guide*
2. *VM/SP HPO Licensed Program Specifications*
3. *VM/SP HPO Directory of Programming Interfaces for Customers*
4. *VM/SP HPO Library Guide and Master Index*
5. *VM/SP HPO Administration*
6. *VM/SP HPO Planning Guide and Reference*
7. *VM/SP HPO Operator's Guide*
8. *VM/SP HPO Installation Guide*
9. *VM/SP HPO CP General User Command Reference*
10. *VM/SP HPO CP System Command Reference*
11. *VM/SP HPO System Messages and Codes*
12. *VM/SP HPO Service Guide*
13. *VM Running Guest Operating Systems*
14. *VM/SP HPO Interactive Problem Control System Guide and Reference*
15. *VM/SP HPO CP Diagnosis Reference*
16. *VM/SP HPO CP Data Areas and Control Blocks*
17. *VM/SP HPO Diagnosis Guide*
18. *VM/SP HPO Problem Determination Summary*

You will not receive the binders or labels. You must order them separately.

Ordering Additional Copies of Manuals

You may use SLSS (see "System Library Subscription Service (SLSS)" on page 52) to order additional copies of VM/SP HPO manuals individually by using the individual order numbers. Or, you may use bill of form numbers (BOFs) to order entire sets of manuals. Three BOFs exist so you can easily order:

- The VM/SP HPO library as a set
- The combined VM/SP HPO and VM/SP libraries as a set
- The VM/SP library as a set.

Note: For Release 6, the VM/SP HPO BOFs do not include binders. You must order them separately.

Ordering the VM/SP HPO Library as a Set

To order the VM/SP HPO library as a set, use bill of form number SBOF-2020. You receive labels and 16 publications:

1. *VM/SP HPO Directory of Programming Interfaces for Customers*
2. *VM/SP HPO Licensed Program Specifications*
3. *VM/SP HPO Library Guide and Master Index*
4. *VM/SP HPO Glossary*
5. *VM/SP HPO Release 6 Guide*
6. *VM/SP HPO Planning Guide and Reference*
7. *VM Running Guest Operating Systems*
8. *VM/SP HPO CP General User Command Reference*
9. *VM/SP HPO CP System Command Reference*
10. *VM/SP HPO Installation Guide*
11. *VM/SP HPO Service Guide*
12. *VM/SP HPO Administration*
13. *VM/SP HPO Interactive Problem Control System*
14. *VM/SP HPO Operator's Guide*
15. *VM/SP HPO System Messages and Codes*
16. *VM/SP HPO Quick Reference.*

Binders do not come with SBOF-2020. You must order them separately. The order number for one binder is SX23-0399. Order six binders for the unlicensed publications. If you order the licensed publications, you will need two additional binders. In addition, you may want to order an extra binder to hold manuals that IBM issues between releases.

Figure 2 shows the arrangement of the VM/SP HPO manuals in binders with labels.

VM/SP HPO Library in Binders

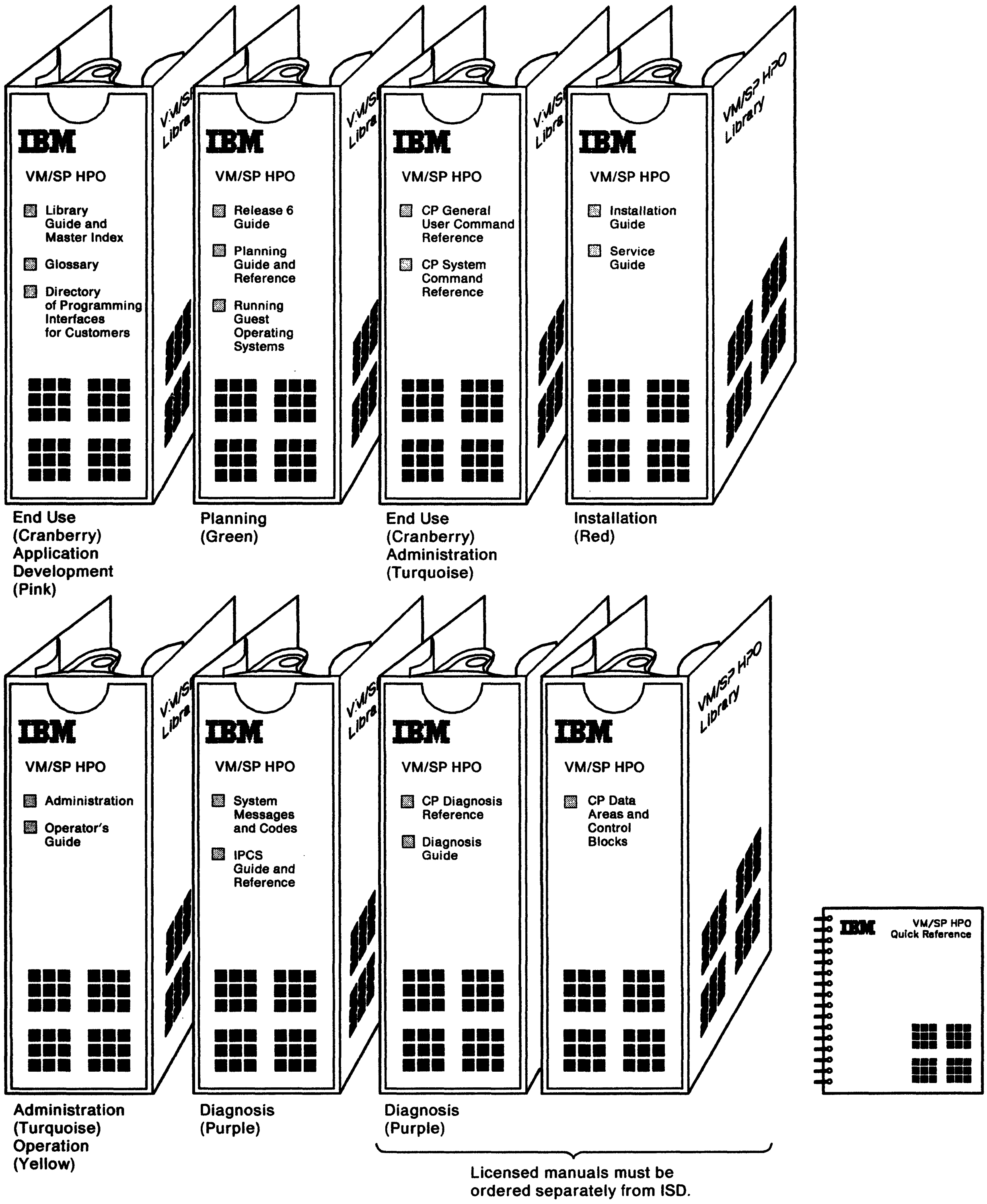

IBM
VM/SP HPO
Library Guide and Master Index
Glossary
Directory of Programming Interfaces for Customers
End Use (Cranberry)
Application Development (Pink)
IBM
VM/SP HPO
Release 6 Guide
Planning Guide and Reference
Running Guest Operating Systems
Planning (Green)
IBM
VM/SP HPO
CP General User Command Reference
CP System Command Reference
End Use (Cranberry)
Administration (Turquoise)
IBM
VM/SP HPO
Installation Guide
Service Guide
Installation (Red)
VM/SP HPO Library
IBM
VM/SP HPO
Administration
Operator's Guide
Administration (Turquoise)
Operation (Yellow)
IBM
VM/SP HPO
System Messages and Codes
IPCS Guide and Reference
Diagnosis (Purple)
IBM
VM/SP HPO
CP Diagnosis Reference
Diagnosis Guide
Diagnosis (Purple)
IBM
VM/SP HPO
CP Data Areas and Control Blocks
VM/SP HPO Library
Licensed manuals must be ordered separately from ISD.
IBM
VM/SP HPO Quick Reference

Figure 2. Ordering and Organizing the VM/SP HPO Library

Ordering the Combined VM/SP HPO and VM/SP Libraries as a Set

To order the combined VM/SP HPO and VM/SP libraries as a set, use bill of form number SBOF-2021. You receive the labels and 40 publications:

1. *VM/SP HPO Licensed Program Specifications*
2. *VM/SP General Information*
3. *VM/SP Introduction*
4. *VM/SP Introduction to Security*
5. *VM/SP HPO Operator's Guide*
6. *VM/SP HPO Release 6 Guide*
7. *VM/SP HPO Planning Guide and Reference*
8. *Virtual Machine Running Guest Operating Systems*
9. *VM/SP HPO Installation Guide*
10. *VM/SP HPO Service Guide*
11. *VM/SP CMS Shared File System Administration*
12. *VM/SP HPO Administration*
13. *VM/SP HPO CP System Command Reference*
14. *Virtual Machine System Facilities for Programming*
15. *VM/SP Connectivity Planning, Administration, and Operation*
16. *VM/SP Application Development Reference for CMS*
17. *VM/SP HPO Directory of Programming Interfaces for Customers*
18. *VM/SP Application Development Guide for CMS*
19. *VM/SP Application Migration Guide for CMS*
20. *VM/SP Connectivity Programming Guide and Reference*
21. *VM/SP SAA CPI Communications Reference*
22. *VM/SP Application Development Guide for FORTRAN and COBOL*
23. *VM/SP Programmer's Guide to the SRPI for VM/SP*
24. *VM/SP Group Control System Macro Reference*
25. *VM/SP HPO Library Guide and Master Index*
26. *VM/SP HPO Glossary*
27. *VM/SP Terminal Reference*
28. *VM/SP CMS Command Reference*
29. *VM/SP CMS User's Guide*
30. *VM/SP HPO CP General User Command Reference*
31. *VM/SP System Product Interpreter User's Guide*
32. *VM/SP System Product Interpreter Reference*
33. *VM/SP EXEC 2 Reference*
34. *VM/SP System Product Editor User's Guide*
35. *VM/SP System Product Editor Command and Macro Reference*
36. *VM/SP HPO System Messages and Codes*
37. *VM/SP HPO IPCS Guide and Reference*
38. *VM/SP HPO Quick Reference*
39. *VM/SP CMS Primer*
40. *VM/SP CMS Primer for Line-Oriented Terminals*

Binders do not come with SBOF-2021. You must order them separately. The order number for one binder is SX23-0399. Order 15 binders for the unlicensed publications. If you order the licensed publications, you will need three additional binders. In addition, you may want to order an extra binder to hold manuals that IBM issues between releases.

Figure 3 on page 45 shows the arrangement of the VM/SP HPO and VM/SP manuals in binders with labels.

VM/SP HPO and VM/SP Library in Binders

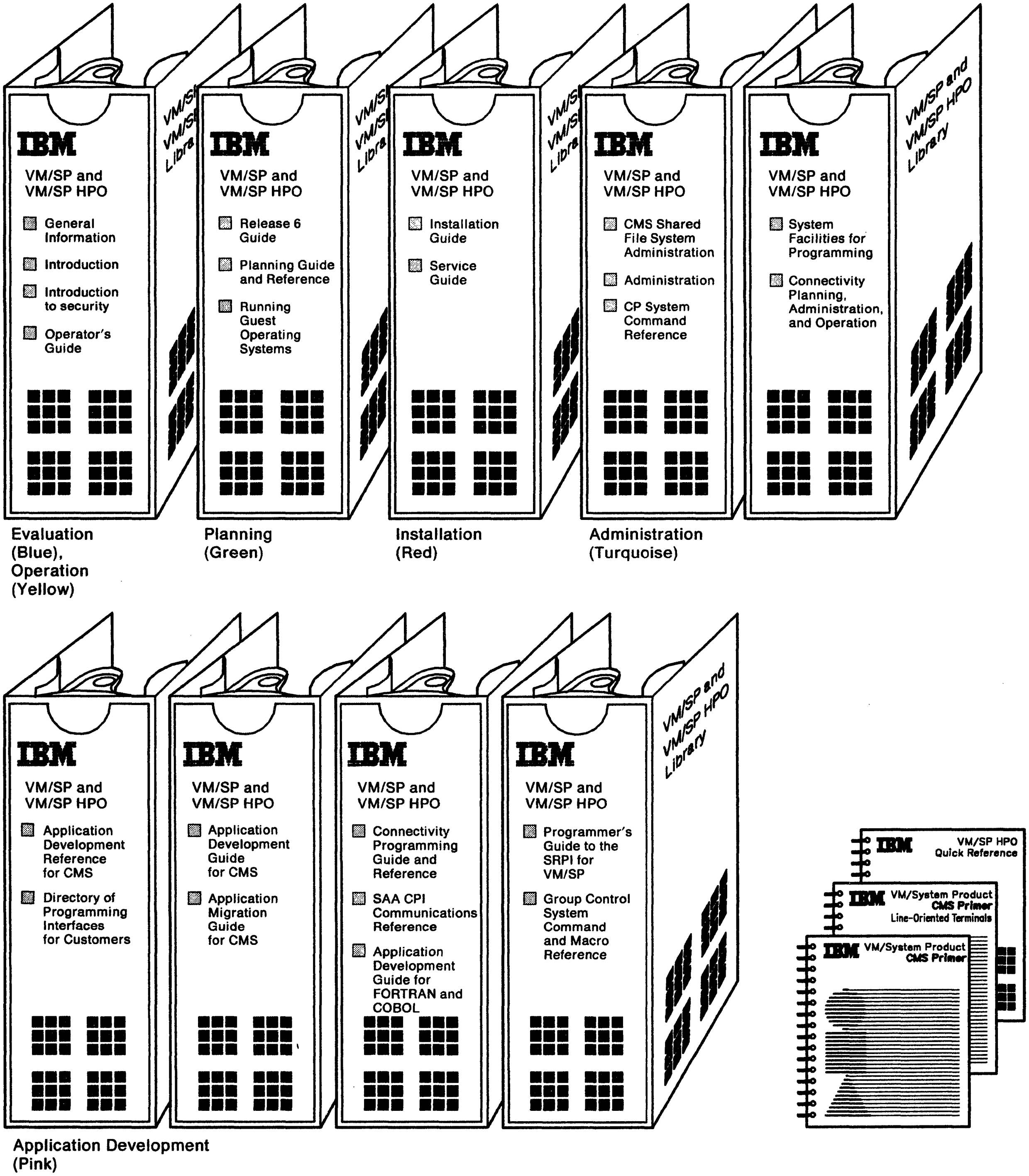

IBM
VM/SP and VM/SP HPO
▨ General Information
▨ Introduction
▨ Introduction to security
▨ Operator's Guide
Evaluation (Blue), Operation (Yellow)
IBM
VM/SP and VM/SP HPO
▨ Release 6 Guide
▨ Planning Guide and Reference
▨ Running Guest Operating Systems
Planning (Green)
IBM
VM/SP and VM/SP HPO
▨ Installation Guide
▨ Service Guide
Installation (Red)
IBM
VM/SP and VM/SP HPO
▨ CMS Shared File System Administration
▨ Administration
▨ CP System Command Reference
Administration (Turquoise)
IBM
VM/SP and VM/SP HPO
▨ System Facilities for Programming
▨ Connectivity Planning, Administration, and Operation
VM/SP and VM/SP HPO Library
IBM
VM/SP and VM/SP HPO
▨ Application Development Reference for CMS
▨ Directory of Programming Interfaces for Customers
IBM
VM/SP and VM/SP HPO
▨ Application Development Guide for CMS
▨ Application Migration Guide for CMS
IBM
VM/SP and VM/SP HPO
▨ Connectivity Programming Guide and Reference
▨ SAA CPI Communications Reference
▨ Application Development Guide for FORTRAN and COBOL
IBM
VM/SP and VM/SP HPO
▨ Programmer's Guide to the SRPI for VM/SP
▨ Group Control System Command and Macro Reference
VM/SP and VM/SP HPO Library
Application Development (Pink)
IBM VM/SP HPO Quick Reference
IBM VM/System Product CMS Primer Line-Oriented Terminals
IBM VM/System Product CMS Primer

Figure 3 (Part 1 of 2). Ordering and Organizing the Combined VM/SP HPO and VM/SP Libraries

VM/SP HPO and VM/SP Library in Binders

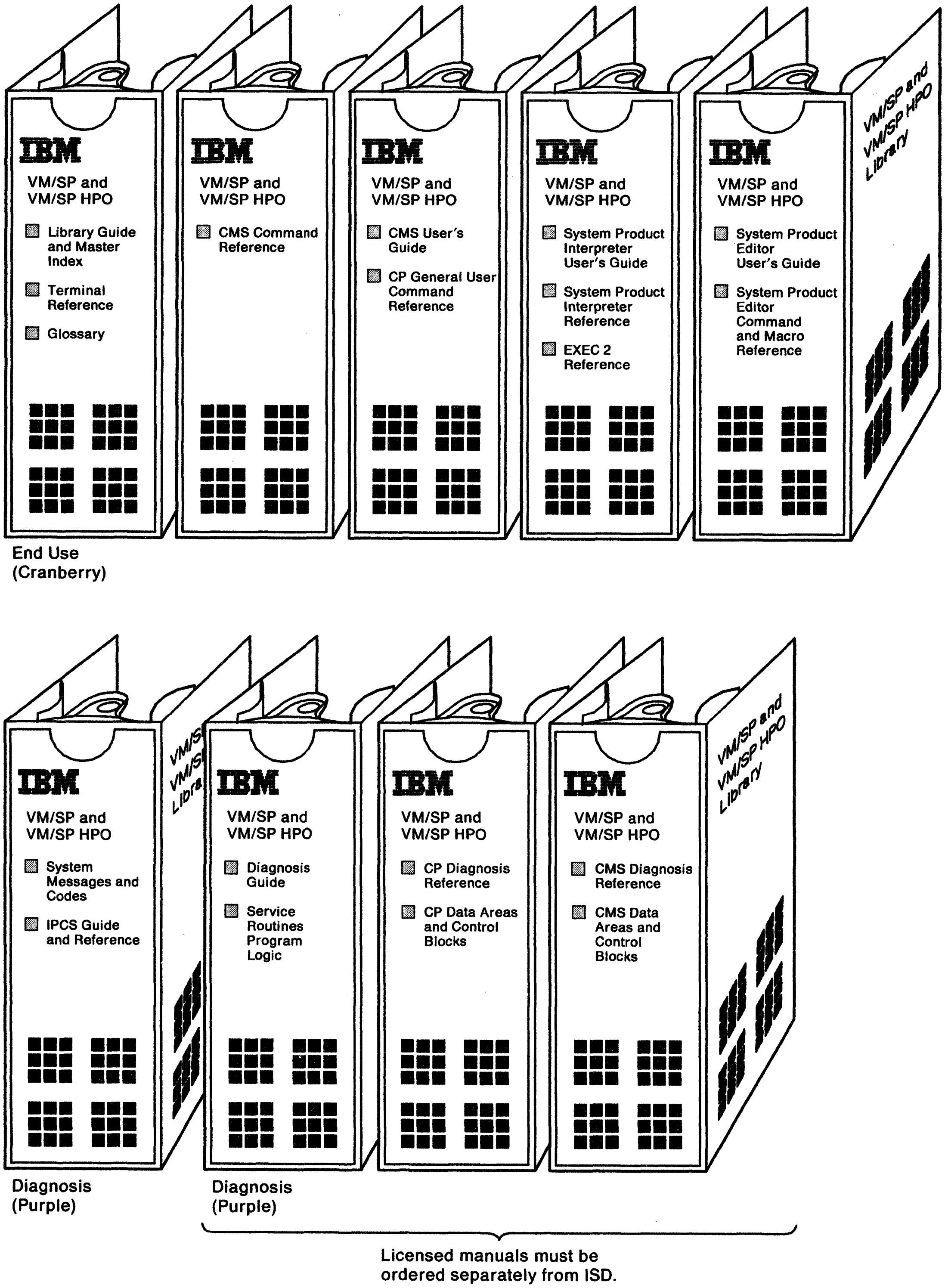

IBM
VM/SP and VM/SP HPO
▨ Library Guide and Master Index
▨ Terminal Reference
▨ Glossary
IBM
VM/SP and VM/SP HPO
▨ CMS Command Reference
IBM
VM/SP and VM/SP HPO
▨ CMS User's Guide
▨ CP General User Command Reference
IBM
VM/SP and VM/SP HPO
▨ System Product Interpreter User's Guide
▨ System Product Interpreter Reference
▨ EXEC 2 Reference
IBM
VM/SP and VM/SP HPO
▨ System Product Editor User's Guide
▨ System Product Editor Command and Macro Reference
VM/SP and VM/SP HPO Library
End Use (Cranberry)
IBM
VM/SP and VM/SP HPO
▨ System Messages and Codes
▨ IPCS Guide and Reference
IBM
VM/SP and VM/SP HPO
▨ Diagnosis Guide
▨ Service Routines Program Logic
IBM
VM/SP and VM/SP HPO
▨ CP Diagnosis Reference
▨ CP Data Areas and Control Blocks
IBM
VM/SP and VM/SP HPO
▨ CMS Diagnosis Reference
▨ CMS Data Areas and Control Blocks
VM/SP and VM/SP HPO Library
Diagnosis (Purple)
Diagnosis (Purple)
Licensed manuals must be ordered separately from ISD.

Figure 3 (Part 2 of 2). Ordering and Organizing the Combined VM/SP HPO and VM/SP Libraries

Ordering the VM/SP Library

You can order the VM/SP library as a set using SBOF-3254 (this includes binders, labels, and manuals) or order the binders, labels, and VM/SP manuals separately. For this ordering information, refer to *VM/SP Release 6 Library Guide and Master Index*, GC19-6207.

Ordering Individual VM/SP or VM/SP HPO Publications

You can order individual VM/SP or VM/SP HPO publications. The order numbers (for the latest release) and pseudonumbers (for previous but still current releases) appear in Table 1 and Table 2 on page 50.

Ordering Publications for VM/SP HPO Releases 4.2, 5, and 6

Table 1 lists the VM/SP HPO publications and order numbers.

Table 1 (Page 1 of 2). Pseudonumbers and Order Numbers for VM/SP HPO Manuals

VM/SP HPO Publication	Release 4.2 Pseudonumber	Release 5 Pseudonumber	Release 6 Order Number
Labels for VM/SP HPO Library	ST00-1909	ST00-3304-00	SX23-0289
Labels for Combined VM/SP HPO and VM/SP Libraries	ST00-1908	ST00-3305-00	SX23-0282
Announcement Brochure	GT00-1895	GC19-6221	NA
What's In VM/SP HPO Release 5?	NA	GC23-0384	NA
Release Guide	ST00-1903	ST00-2750-00	SC23-0189-4
Directory of Programming Interfaces For Customers	NA	NA	GC23-0516
Library Guide, Glossary, and Master Index	GT00-1914	GT00-2749-00	NA
Library Guide and Master Index	NA	NA	GC23-0187-4
Glossary	NA	NA	SC23-0441-0
Licensed Program Specifications	GT00-1902	GT00-2751-00	GC20-1844-9
Planning Guide and Reference	ST00-1896	ST00-2752-00	SC19-6223-8
Installation and Service Guide	ST00-1904	ST00-2719-00	NA
Installation Guide	NA	NA	SC38-0107-7
Service Guide	NA	NA	SC23-0440-0
System Definition Files	SC23-0186	NA	NA
Service Routines Program Logic	LT00-1913	LY20-0898	NA
System Programmer's Guide	ST00-1897	NA	NA
CP for System Programming	NA	ST00-2753-00	NA
Administration	NA	NA	SC19-6224-8
Problem Determination Reference Information	LT00-1910	LT00-2375-00	NA
Problem Determination Summary	NA	NA	LX23-0347-5

Table 1 (Page 2 of 2). Pseudonumbers and Order Numbers for VM/SP HPO Manuals

VM/SP HPO Publication	Release 4.2 Pseudonumber	Release 5 Pseudonumber	Release 6 Order Number
Operator's Guide	ST00-1898	ST00-2754-00	SC19-6225-8
System Messages and Codes	ST00-1899	ST00-2755-00	SC19-6226-8
System Messages Cross-Reference	SQ23-0190	NA	NA
CP Command Reference	ST00-1900	ST00-2756-00	NA
CP General User Command Reference	NA	NA	SC19-6227-8
CP System Command Reference	NA	NA	SC23-0442-0
Interactive Problem Control System Guide and Reference	NA	NA	SC23-0443-0
Virtual Machine Running Guest Operating Systems	GT00-1894	GT00-3300-00	GC19-6212-6
OLTSEP and Error Recording Guide	ST00-1901	NA	NA
Data Areas and Control Block Logic – CP	LT00-1911	LT00-2757-00	NA
CP Data Areas and Control Blocks	NA	NA	LY20-0896-8
System Logic and Problem Determination Guide – CP	LT00-1912	LT00-2758-00	NA
CP Diagnosis Reference	NA	NA	LY20-0897-8
Diagnosis Guide	NA	NA	LY27-8061-0
Command Summary (General Users)	ST00-1905	ST00-3302-00	NA
CP General User Command Reference Summary	NA	NA	SX22-0003-8
Command Summary (Other than General Users)	ST00-1906	ST00-3303-00	NA
CP System Command Reference Summary	NA	NA	SX22-0004-8
Quick Reference	ST00-1907	ST00-3301-00	SX22-0005-8

Notes:

1. *VM Service Routines Program Logic* contains both VM/SP and VM/SP HPO information.

2. *System Programmer's Guide* was split into four publications for Release 5. Order *VM/SP HPO CP for System Programming*, *VM/SP CMS for System Programming*, *Virtual Machine System Facilities for Programming*, and *Virtual Machine Diagnosis Guide* (in Table 2 on page 50).

3. *VM/SP HPO Library Guide, Glossary, and Master Index* split into two publications for Release 6. Order the *VM/SP HPO Library Guide and Master Index* and the *VM/SP HPO Glossary.*

4. *VM/SP HPO Installation and Service Guide* split into two publications for Release 6. Order the *VM/SP HPO Installation Guide* and the *VM/SP HPO Service Guide.*

5. *VM/SP HPO CP for System Programming* was renamed and reorganized for Release 6. Most of the material is now found in *VM/SP HPO Administration.* Other material is now in *VM/SP HPO CP Diagnosis Reference, VM/SP HPO Planning Guide and Reference, VM Running Guest Operating Systems*, and *VM/SP HPO Operator's Guide.*

6. *VM/SP HPO CP Command Reference* split into two publications for Release 6. Order the *VM/SP HPO CP General User Command Reference* and the *VM/SP HPO System Command Reference.*

7. *VM/SP HPO Interactive Problem Control System Guide and Reference* previously was named *VM/SP Interactive Problem Control System Guide and Reference.*

8. *VM/SP HPO Data Areas and Control Block Logic—CP* is now named *VM/SP HPO CP Data Areas and Control Block Logic.*

9. *VM/SP HPO System Logic and Problem Determination Guide* is now named *VM/SP HPO CP Diagnosis Reference.*

10. *VM/SP HPO Diagnosis Guide* previously was named *VM Diagnosis Guide.*

11. *VM/SP HPO General User Command Summary* previously was named *VM/SP HPO Command Summary (General User).*

12. *VM/SP HPO System Command Summary* previously was named *VM/SP HPO Command Summary (Other Than General User).*

Ordering Publications for VM/SP Releases 4, 5, and 6

VM/SP HPO Release 4.2 operates in conjunction with VM/SP Release 4. VM/SP HPO Release 5 operates in conjunction with VM/SP Release 5. VM/SP HPO Release 6 operates in conjunction with VM/SP Release 6. Table 2 lists pseudonumbers and order numbers for applicable VM/SP Release 4, 5, and 6 publications.

Table 2 (Page 1 of 2). Pseudonumbers and Order Numbers for VM/SP Manuals

VM/SP Publication	Release 4 Pseudonumber	Release 5 Pseudonumber	Release 6 Order Number
Introduction	GT00-1575	GT00-1977	GC19-6200-05
General Information	GT00-1587	GT00-1976	GC20-1838-07
Introduction to Security	NA	ST-24-5316	SC24-5316-01
Terminal Reference	GT00-1581	GT00-1979	GC19-6206-05
CMS Command Reference	ST00-1583	ST00-1981	SC19-6209-05
CMS User's Guide	ST00-1584	ST00-1980	SC19-6210-05
System Product Editor User's Guide	ST00-1589	ST00-1985	SC24-5220-04
System Product Editor Command and Macro Reference	ST00-1590	ST00-1986	SC24-5221-05
System Product Interpreter User's Guide	ST00-1593	ST00-1987	SC24-5238-04
System Product Interpreter Reference	SQ24-5239	ST00-1988	SC24-5239-03
EXEC 2 Reference	ST00-1368	ST00-1984	SC24-5219-04
Application Development Guide for FORTAN and COBOL	SQ24-5247	ST00-2086	SC24-5247-03
CMS Macros and Functions Reference	NA	ST24-5284	NA
CMS Application Development Reference for CMS	NA	NA	SC24-5284-01
Programmer's Guide to the Server-Requester Programming Interface for VM/SP	NA	ST24-5291	ST24-5291-02
CMS for System Programming	NA	ST24-5286	NA
Application Development Guide for CMS	NA	NA	ST24-5286-01
TSAF Reference	NA	ST24-5287	NA
Connectivity Planning, Administration, and Operation	NA	NA	SC24-5378-0
Connectivity Programming Guide and Reference	NA	NA	SC24-5377-0
Group Control System Command and Macro Reference	ST24-5250	SQ24-5250	SC24-5250-02
Virtual Machine System Facilities for Programming	NA	ST24-5288	SC24-5288-01
SAA Common Programming Interface Communications Reference	NA	NA	SC26-4399-00

Table 2 (Page 2 of 2). Pseudonumbers and Order Numbers for VM/SP Manuals

VM/SP Publication	Release 4 Pseudonumber	Release 5 Pseudonumber	Release 6 Order Number
Distributed Data Processing Guide	ST24-5241	SQ84-5241	NA
System Logic and Problem Determination Guide Volume 2 (CMS)	LT00-1819	LT00-2007	NA
CMS Diagnosis Reference	NA	NA	LY20-0893-00
VM Application Migration Guide for CMS	NA	NA	SC24-5366-0
Data Areas and Control Block Logic – CMS	LQ64-5221	LT00-2009	NA
CMS Data Areas and Control Blocks	NA	NA	LY24-5221-00
Virtual Machine Diagnosis Guide	NA	LT00-2010	NA
Group Control System Diagnosis Reference	LT24-5239	LT00-2012	NA
CMS Primer	ST00-1591	ST00-1992	SC24-5236-04
CMS Primer for Line-Oriented Terminals	SQ24-5242	ST00-1993	SC24-5242-03
CMS Shared File System Administration	NA	NA	SC24-5367-0
HELP Facility Introduction	NA	NA	SX224-5221-0
VM/SP Diagnosis Guide	NA	NA	LY24-5241

Library-Related Services

System Bibliography

The *IBM System/370, 30xx, 4300, and 9370 Processors Bibliography*, GC20-0001, lists by subject code and provides abstracts (by order number) of all the publications available which pertain to VM/SP HPO. These abstracts are brief descriptions of the purpose, scope, and intended audience of each publication.

System Library Subscription Service (SLSS)

The System Library Subscription Service (SLSS) is an automatic updating service for technical documentation. If IBM releases a new or revised document which matches your SLSS subscription, you automatically receive the latest version or edition of the publication, and any applicable Technical Newsletters.

You can enter a subscription in two different ways: by creating an *interest profile,* or by listing the *order numbers* of the manuals you want. Both methods use the SLSS order form, G120-1816.

Order by interest profile when you need all the documents relating to your installation. Profiles are composed of specific information about your processor, program, communications needs and your special subjects of interest about your system.

You should subscribe by order number when you need multiple copies, when the publication is not part of the set automatically shipped with the software, or when it is not available through the profile service. For example, when you order VM/SP HPO Release 6, an initial library is shipped with it, and you will automatically receive updates to these books without an SLSS subscription. Use SLSS when you want updating service for multiple copies of any non-licensed books, or for any additional or optional books you want to order. You will be billed for any use-key S manuals which you receive from SLSS.

You can update your subscription at any time using the SLSS order form. For more information on SLSS see *Entering an SLSS Subscription*, G320-1561 or contact your IBM marketing representative.

Technical Newsletters (TNLs)

TNLs are issued to update and correct existing books. They consist of a cover letter listing the changes, and either pages to add to the existing manual or pages to substitute for the original pages.

A book is revised when the current stock to which TNLs have been issued is exhausted or when major changes are made in its content.

TNLs and major revisions are automatically sent to users who subscribe to the base document using the System Library Subscription Service.

Between-Release Publications for PTFs

From time to time, IBM provides new VM/SP HPO support that is announced between releases. Such support is available as program temporary fixes (PTFs) on a Program Update Tape (PUT) and is documented by a separate publication.

These publications, called stand-alone guides, describe the new support, hardware device, new or changed commands and messages, system generation considerations, and new and changed modules. The publication contains only new information, so you should use it along with the regular library.

The distribution of stand-alone guides differs slightly from the distribution of entitled manuals. All new licenses receive stand-alone guides automatically. Existing licenses receive a letter describing the support and a post card asking them if they would like to receive the stand-alone guide.

We suggest that you order a binder, SX23-0399, to hold the publications. We are providing a label that you can use to keep track of the books in the binder.

The booklets listed in Table 3 on page 54 have been issued for PTFs on releases other than VM/SP HPO 6. Because the information is included in the Release 6 manuals, these booklets are no longer necessary.

Table 3. Current Program Enhancement Publications

Title	Order Number	VM/SP R4	VM/SP HPO R4.2	VM/SP R5	VM/SP HPO R5	VM/SP R6
DASD DUMP/Restore Streaming Support Improvements	GC24-5359	X		X		
SPOOL Enhancement Accommodation	GC24-5361			X		
Enhancements to the IBM Enhanced Connectivity Facilities for VM/System Product	GC24-5295			X		
IBM 3380 Direct Access Storage Models AJ4/BJ4 and AK4/BK4	GC24-5371	X	X	X	X	
IBM 3990 Storage Control Models 1 and 2 and IBM Direct Access Storage Direct Channel Attach Model CJ2	GC24-5372	X	X	X	X	
Terminal Usability Enhancements	GC24-5309			X	X	
VM/VTAM and NetView Enhancements	GC24-5310			X		
DIAGNOSE Code X'64' Enhancement	GC24-5311	X	X	X	X	
DIAGNOSE Code X'E4' Enhancement	GC24-5376			X	X	
Support for the IBM 3990 Storage Control Model 3	GC23-0430				X	
Support for Mixed Directory	GC23-0436		X		X	VM33114
Productivity Aids National Language Support Enhancement	GC24-5400			X		
National Language Support File Naming Conventions Enhancements	GC24-5418			X	X	X
Enhancements to the LINERD and LINEWRT Macros	GC24-5313					X
Alternate-VSAM	GC24-5399					X
Security and Integrity Enhancements	GC24-5312					X
Support for VM/Directory Maintenance Licensed Program Release 4	GC23-0459		X		X	
Free Storage Enhancements	LY27-8062				X	
Paging Storage and Vector Facility Changes	GC23-0463		X		X	

Online HELP Facility

Online HELP information, intended for both new and experienced users, contains information about commands, instructions, and messages. Online HELP provides menus of commands and tasks on which information is available, as well as several ways to display this information. HELP information is available for:

- Commands
  - APPC/VM VTAM Support (AVS), CMS, all CP, IPCS and RSCS
- Subcommands
  - EDIT, XEDIT, Servor-Requestor Programming Interface (SRPI), IPCSSCAN and IPCSPRT
- Macros
  - CMS assembler language
- Options
  - For the QUERY and SET commands of both CMS and CP
- Routines
  - From the VMLIB callable services library (CSL)
- Messages
  - CMS, CP, RSCS, TSAF, AVS, GCS, IPCS
- Statements
  - EXEC and EXEC2, REXX

New users can read the *VM/SP HELP Facility Introduction* and the HELP chapter in the *VM/SP CMS Primer* for information on how to use online HELP.

Abstracts of Manuals

The following *abstracts* present a description of each manual and, consequently, a better idea about which manual is best suited for your task. The VM/SP HPO manuals are listed first, followed by the VM/SP manuals.

VM/SP HPO Publications

Library Guide and Master Index, GC23-0187

This manual describes the documentation available for VM/SP HPO and includes a compilation of all the index entries for the library. The master index section is particularly helpful if you are not familiar with the VM/SP HPO library as it directs you to the correct manual for a particular topic.

Glossary, GC23-0441

This manual defines terms used throughout the VM/SP HPO manuals.

Planning Guide and Reference, SC19-6223

This manual contains planning and installation information for installing a VM/SP HPO system.

This publication includes information about planning for system generation and defining your VM/SP HPO operating system. Appendixes provide:

- Information about various configurations
- Restrictions on VM/SP HPO
- A worksheet for coding the NAMESYS macro
- Sample SNTMAP output.

The intended audience is system programmers and anyone responsible for the planning, installation, and updating of a VM/SP HPO system. The reader is expected to have a general understanding of data processing and teleprocessing techniques.

Installation Guide, SC38-0107

This manual is for system programmers and anyone responsible for installing VM/SP HPO. It contains step-by-step procedures for using the Starter System or an existing VM/SP HPO system to install VM/SP HPO. In addition, there are procedures for verifying an installed system, installing saved segments, and installing a new system national language.

Service Guide, SC23-0440

This manual is for system programmers and anyone responsible for servicing VM/SP HPO. It contains step-by-step procedures for program update (preventative) service, corrective service, and local service. It also contains descriptions of the various service tools and service files, including the VM/SP HPO and VM/SP product parameter files.

Some of the material in this manual previously was found in *VM/SP HPO Installation and Service Guide*, SC38-0107.

Administration, SC19-6224

This is a reference manual for the system administrator and others who carry out and extend the functions of IBM's Virtual Machine/System Product High Performance Option (VM/SP HPO). It assumes some experience with programming concepts and techniques. This reference consists of three parts and an appendix:

> Part 1: *Tailoring Your VM/SP HPO System* describes some functions of VM/SP HPO and provides guidance to the system administrator in tailoring those functions.
> Part 2: *Performance* describes options available in VM/SP HPO to analyze and improve the performance of virtual machines and operating systems.
> Part 3: *Hardware Considerations* provides information about various types of printers, DASD, Paging Storage, storage control subsystems, and Vector Facilities.
> Appendix, "VM/SP HPO Monitor Tape Format and Content" describes the format and contents of data records for classes and codes of MONITOR CALL.

This manual contains information previously found in the Release 5 book *CP for System Programming*, SC19-6224.

CP General User Command Reference, SC19-6227

This manual is a reference for class G and class Any users who are running systems such as OS, OS/VS, DOS, DOS/VS, and VSE systems, CMS, and networking systems in a virtual machine under VM/SP HPO. Control Program (CP) commands available to privilege classes G and Any are listed alphabetically. Each command description contains general usage information, the command line format, descriptions of all allowable operands, default values for operands, usage notes, and messages generated by this command.

Material in this manual previously was found in the *VM/SP HPO CP Command Reference*, SC19-6227.

CP System Command Reference, SC23-0442

This manual is a reference for class A, B, C, D, E, and F users who are running systems such as OS, OS/VS, DOS, DOS/VS, and VSE systems, CMS, and networking systems in a virtual machine under VM/SP HPO. Control Program (CP) commands available to privilege classes A, B, C, D, E, and F are listed alphabetically. Each command description contains general usage notes, the command line format, descriptions of all allowable operands, and default values for operands, and messages generated by the command.

Material in this book previously was found in the *VM/SP HPO CP Command Reference*, SC19-6227.

Directory of Programming Interfaces for Customers, GC23-0516

This publication lists all the programming interfaces that are available in VM/SP HPO for programmers to use in writing programs. These interfaces are documented in the rest of the VM/SP HPO library. It describes each interface by name, book and chapter location in the library. This publication also specifies which interfaces IBM prefers programmers to use, and which should be used with caution.

Interactive Problem Control System Guide and Reference, SC23-0443

This manual contains an overview of IPCS functions, data flow, and files. It describes preliminary and repetitive IPCS operations, and gives insight into using IPCS in different programming environments. It is also a reference manual that contains a detailed description of all the IPCS commands and subcommands.

This manual contains information previously found in the Release 5 book *VM Diagnosis Guide, LY24-5241.*

Operator's Guide, SC19-6225

This manual is for those responsible for the operation and administration of a VM/SP HPO system. This book describes the recovery features and how to initialize and terminate the system. It briefly introduces the various commands, CP and CMS, that an operator may find useful. It also includes some information about hardware considerations and spooling and Batch commands.

System Messages and Codes, SC19-6226

This manual contains messages and codes, as well as restrictions, that the user may encounter when using CMS, CP, IPCS, GCS, AVS, and TSAF. Conditions that generate these messages and codes are explained, the resulting action is described, and appropriate responses are suggested.

Running Guest Operating Systems, GC19-6212

This manual has been restructured to include updated and practical examples, and expanded information about running guests. This publication is for people who are interested in running their operating system (VSE, MVS, and VM) under the VM host system. The book presents the basic processes so that the working system can be set up quickly. Experienced persons can improve the efficiency of their installation, running a guest system under VM, with the included recommendations. The book assumes the audience is knowledgeable about their own system but new to VM.

Some of the material in this book previously was found in *VM/SP HPO CP for System Programming*, SC19-6224.

Diagnosis Guide, LY27-8061

This manual is a guide to identifying and collecting information, and reporting and solving problems in the VM/SP HPO licensed program.

Material in this book previously was found in *VM Diagnosis Guide*, LY24-5241.

CP Diagnosis Reference, LY20-0897

This manual provides general information needed to analyze problems that occur when running VM/SP HPO.

This manual previously was named *VM/SP HPO System Logic and Problem Determination Guide—CP*, LY20-0897.

CP Data Areas and Control Blocks, LY20-0896

This manual describes the CP data areas and control blocks used for communication between components and modules of VM/SP HPO. The data areas and control blocks are arranged in alphabetical order by their DSECT name. A descriptive statement is followed by a graphic representation of the arrangement of the fields. Also included is a cross reference containing all the labels defined in the data area as well as any general equates that the copy file may contain.

This manual previously was named *VM/SP HPO Data Areas and Control Block Logic—CP*, LY20-0896.

Abstracts of Reference Summaries

CP General User Command Reference Summary, SX22-0003

This reference card provides the syntax for all class G commands.

This information previously was found in the *VM/SP HPO CP Command Summary (General User)*, SX22-0003.

CP System Command Reference Summary, SX22-0004

This reference card provides the syntax for all non-class G commands.

This information previously was found in the *VM/SP HPO CP Command Summary (Other Than General User)*, SX22-0004.

Quick Reference, SX22-0005

The first section of this spiral-bound book provides a summary of system reference information about communicating with the system and interpreting responses, and working with two of its components, CP and CMS. The second section consists of syntax diagrams and brief descriptions of commands and service aids.

Problem Determination Summary, LX23-0347

This reference card summarizes techniques for finding the cause of a problem, shows the CP trace table, and the PSA (prefix storage area).

VM/SP Publications

VM Application Migration Guide for CMS, SC24-5366

The *VM/SP Application Migration Guide for CMS* provides application program migration information for application programmers and system programmers who are migrating programs to VM/SP HPO Release 6. It summarizes some of the new and changed features of CMS, including Shared File System compatibility and exploitation, Callable Services Library features, the CMS formalized programming interface, Extended Architecture support in CMS, and CMS functional changes for calling and handling programs, handling interrupts, managing storage and saved segments, and using OS/MVS simulation.

Application Development Guide for FORTRAN and COBOL, SC24-5247

The *Application Development Guide for FORTRAN AND COBOL* tells FORTRAN and COBOL application programmers how to compile, link, load, run, test, and debug programs using CMS. It also contains information on using the Interactive System Productivity Facility (ISPF) for dialog management and the Structured Query Language/Data System (SQL/DS) for data base management.

Application Development Guide for CMS, SC24-5286

The *VM/SP Application Development Guide for CMS* provides application programmers and system programmers with information about CMS. This information includes details on the CMS programming interface and CMS architecture, using CMS native services (to handle interrupts, obtain free storage, perform I/O, and process abends), and managing CMS programs (using CMS services to build, load, execute, and update programs and program packages). It also includes information on developing OS/MVS and VSE programs under CMS, and using Access Method Services and VSAM under CMS and CMS/DOS. Much of the information in this book was formerly found in *CMS for System Programming*, ST24-5286.

Application Development Reference for CMS, SC24-5284

All of the information contained in the Release 5 CMS Macros and Functions Reference is now in this manual. New CMS macros are also incorporated. In addition to detailed reference information about CMS assembler language macro instructions and CMS functions, a callable services library (CSL) is described. The routines in this library can be called from assembler, COBOL, VS Fortran, PL/I, VS Pascal, and REXX languages. Many of these routines facilitate the use of the CMS Shared File System.

The manual is intended for application programmers, system programmers, and IBM system support personnel.

CMS Command Reference, SC19-6209

This publication provides users of the Conversational Monitor System (CMS) component of VM/SP with detailed reference information about command syntax and usage for: CMS commands, EDIT subcommands, HELP format words, DEBUG commands, EXEC control statements, special variables, and built-in functions.

CMS Primer, SC24-5236

This manual teaches you, as a new user, how to do your work using VM/SP and a video display terminal. The primer presents only a subset of all the functions and commands available with VM/SP. The material is presented in an interesting manner with suggested exercises included in the text.

CMS Primer for Line-Oriented Terminals, SC24-5242

This manual is an interactive tutorial for VM/SP users of line-oriented (line mode) video display terminals. The manual, which is similar in scope and content to the *VM/SP CMS Primer*, is designed to quickly give the reader a working knowledge of VM/SP. Topics include logging on, editing, managing, and printing files, using the Document Composition Facility (SCRIPT/VS) to format files, and writing EXECs.

CMS Shared File System Administration, SC24-5367

This book helps system administrators and programmers administrate file sharing in CMS. It includes information about managing users and storage, protecting and securing data, creating and managing file pools and file pool catalogs, monitoring performance, accounting and tracing the system.

CMS User's Guide, SC19-6210

This manual contains general information on the Conversational Monitor System (CMS) component of VM/SP. It provides information and examples regarding the CMS Shared File System, the CMS batch facility, the HELP facility, and windowing support. Also included are examples on using the System Product Editor to create and edit CMS files, and to create and use EXECs which can be executed by the System Product Interpreter.

Connectivity Planning, Administration, and Operation, SC24-5378

This manual provides an overview of the products and the VM/SP components related to connectivity and an overview of the programming interface, Advanced Program-to-Program/Virtual Machine (APPC/VM) and the Common Programming Interface (CPI) for Communications of System Application Architecture (SAA). This manual contains all of the reference material needed to use the Transparent Services Access Facility (TSAF) and APPC/VM VTAM Support (AVS). It includes information for the system programmer and system operator to run the TSAF virtual machine, to run the AVS virtual machine, to use the CP System Services, and to use the service tools that are provided with TSAF and AVS. Contains information formerly found in the *TSAF Reference*, SC24-5287.

Connectivity Programming Guide and Reference, SC24-5377

This book is intended for someone writing an application program to communicate with another application program using APPC protocol. The communication can occur within a single TSAF collection, across multiple TSAF collections, or between a TSAF collection and an SNA network.

This book introduces the two kinds of APPC/VM communications programming interfaces:

1. Systems Application Architecture™ (SAA) Common Programming Interface (CPI) Communications, intended for programs written in high-level languages (such as COBOL) and procedures languages (REXX).
2. The APPC/VM macro interface, for programs written in assembler language.

This book provides complete details on the APPC/VM assembler interface; readers are referred to the *SAA Common Programming Interface Communications Reference*, SC26-4399, for details on CPI Communications.

EXEC 2 Reference, SC24-5219

This is a reference manual that defines the EXEC 2 language. It contains all the formats, syntax rules, and descriptions of the arguments for EXEC 2 statements. An EXEC 2 primer for new users is included. It summarizes the language and its capabilities. A detailed discussion of the different types of EXEC 2 statements is followed by examples. It lists the error messages and return codes issued by the EXEC 2 interpreter.

General Information, GC20-1838

This manual introduces and describes the features and facilities of VM/SP and provides customer management and technical staffs with information needed to evaluate the applicability of VM/SP to their installations. It contains information needed for a basic understanding of the environments and functions that are supported by VM/SP.

Group Control System Command and Macro Reference, SC24-5250

This manual provides detailed reference material that describes the functions and use of all macros supported in the Group Control System. Each macro description contains information on general usage, format, all available parameters, messages, and return codes. The manual is intended for personnel who are developing application programs to run on the Group Control System.

HELP Facility Introduction, SX24-5221

This reference card shows the menu structure of the online HELP facility, the format of HELP panels and the syntax of the command formats in HELP. It also shows how to get HELP for system messages, and lists the filetypes reserved by the system for use by the HELP facility.

Programmer's Guide to the Server-Requester Programming Interface for VM/SP, SC24-5291

This manual provides an application programmer with information on how to write and install IBM Enhanced Connectivity Facilities servers in a VM/SP system. For the work station (for example, IBM Personal Computer or Personal System/2) user, the manual also gives information on how to start IBM Enhanced Connectivity Facilities communications on VM/SP.

Systems Application Architecture is a trademark of the IBM Corporation.

Introduction, GC19-6200

This manual is an extension of the *VM/SP General Information* manual. It describes, at an introductory level, what VM/SP is and what it can do for you. It is intended for anyone who wants basic information about VM/SP facilities and capabilities. This manual is useful for those who plan a VM/SP installation, use program products with VM/SP, or migrate to VM/SP from another operating system. This manual is suggested as a prerequisite before reading the more detailed reference manuals in the VM/SP library.

Introduction to Security, SC24-5316-01

This book is intended to introduce VM/SP system and application programmers to the topic of security. It acquaints the reader with the security challenges that might confront his or her organization, and the security tools and facilities provided by VM/SP to meet those challenges. It includes information on maintaining physical security and system integrity, controlling access to the VM/SP system, protecting data, and performing a risk analysis.

SAA Common Programming Interface Communications Reference, SC26-4399

This book describes SAA Common Programming Interface (CPI) Communications in detail, including scenarios and individual routines. It is intended for someone writing an application program that is to communicate with another application program using APPC protocol. The communications can occur within a single TSAF collection, across multiple TSAF collections, or between a TSAF collection and an SNA network.

VM System Facilities for Programming, SC24-5288

This manual provides a system programmer with detailed information on facilities available in VM/SP and VM/SP HPO. Included are the DIAGNOSE instruction, the Inter-User Communications Vehicle (IUCV) for CP and CMS, CP System Services, the Virtual Machine Communication Facility (VMCF), the programmable operator facility, and information on getting national languages on your system. Information on DASD Operations (including the Format/Allocate and DASD Dump Restore (DDR) programs), the CMS Batch Facility, the Collection Resource Management System Service, and the Identify System Service are now also contained in this book.

This information was extracted from the Release 5 books *CP for System Programming*, SC24-5285, *CMS for System Programming*, SC24-5286, and the *Transparent Services Access Facility (TSAF) Reference*, SC24-5287, respectively.

System Messages Cross Reference, SC24-5264

This manual contains cross-reference information for the messages in the manual, *VM/SP Messages and Codes*, SC19-6204. The cross-reference information relates messages to the commands that caused them to be issued, lists messages alphanumerically by message identifier, relates messages to the module that issued them, and lists the messages in alphanumeric order by message text. The manual consists of five appendices, one for each of the VM/SP components: Conversational Monitor System (CMS), Control Program (CP), Interactive Problem Control System (IPCS), Group Control System (GCS), and Transparent Services Access Facility (TSAF).

System Product Editor Command and Macro Reference, SC24-5221

This publication is a reference manual that contains all the command formats, syntax rules, and operand and option descriptions for the XEDIT subcommands and macros. It tells how to enter XEDIT commands, subcommands and macros. It contains the format description and operand and option list for the XEDIT command, which is used to invoke the editor. It lists EDIT subcommands and their XEDIT counterparts, and describes the relationship between XEDIT and the CMS Session Services. You should be familiar with the information in the *System Product Editor User's Guide*, SC24-5220, before attempting to use this manual.

System Product Editor User's Guide, SC24-5220

This manual is intended for the individual who has limited data processing experience. It is designed to give the user a working knowledge of the System Product Editor (called by the XEDIT command). XEDIT provides a wide range of functions for text processing and programming development. Both a full screen and a line mode editor, it can be used on display and typewriter terminals.

System Product Interpreter Reference, SC24-5239

This manual provides reference information about System Product Interpreter statements and their use. It is suitable for experienced programmers, particularly those who have used another high-level language (e.g., PL/I, Algol, or Pascal). It includes error messages and describes syntax, instructions, functions, debugging aids, and parsing.

System Product Interpreter User's Guide, SC24-5238

This manual is a step-by-step guide to using the System Product Interpreter, the powerful interpretive command and macro language. It is intended for a user with some knowledge of terminals, editors, and VM/SP, but the user need not have any previous programming experience.

The novice should read the System Product Interpreter chapter in the *VM/SP CMS Primer*, SC24-5236, for prerequisite knowledge.

Terminal Reference, GC19-6206

This manual is intended for those users who plan to use terminals with VM/SP in their operations. It discusses the characteristics of terminals in general and the physical characteristics of some terminals and consoles supported by VM/SP. It discusses how to use terminals to access and control VM/SP, and contains an example of a typical session from logon to logoff.

Abstract of Reference Poster

Summary of End Use Tasks and Commands, SX24-5173

The *VM Summary of End Use Tasks and Commands*, SX24-5173, is a poster listing the nine major tasks and the associated commands most frequently used by end users. This poster contains excerpts of command information found in various CP, CMS, and XEDIT books. The *Summary* poster is helpful for general users who need a reminder list of the commands used to perform routine daily tasks.

VM CP Trace Table (Poster), SC24-5225

This table shows the format (type of event, module, and hex value) of the entries in the CP internal trace table. This table records events that occur in the real machine (such as interrupts, communications, and instructions).

Chapter 5. Program Distribution

This chapter briefly describes the following system requirements for VM/SP HPO:

- Product tape
- Feature numbers
- System maintenance
- Testing period
- License
- Program services
- Warranty.

VM/SP HPO Product Tape

IBM distributes VM/SP HPO on product tapes that you can load onto your system's direct access storage devices. The product tapes are distributed in the following formats:

- 9-track 6250 BPI reel
- 18-track 38K BPI 3480 cartridge.

When you merge the product tape with the prerequisite VM/SP, you obtain an operational system that combines the features of VM/SP with the functional extensions available in VM/SP HPO. Refer to the following page for an illustration of the layout of the product tape.

If you are installing an IBM virtual machine facility licensed program for the first time, you must order a starter system. The starter system is a basic system with control program and conversational monitor system functions. These functions provide the system support necessary for generating VM/SP HPO.

Starter systems must be restored to DASD. IBM makes available starter systems that can be restored to 3350, 3375, and 3380 DASD. You should obtain the starter system that conforms to your system DASD configuration. See Table 5 for the layout of the starter system.

If you have an existing system that is running VM/SP HPO Releases 4.2 or 5, it is not necessary to use a starter system; you can generate VM/SP HPO from your existing system.

See "Feature Numbers" on page 69 for feature number information.

Product Tape Layout for VM/SP HPO Release 6

Table 4 lists the files that are written on the product tape for VM/SP HPO Release 6.

Table 4. Product Tape Layout

File Number	Contents
File 1	Header File
File 2	HPOINST EXEC, HUTILITY EXEC, HITASK EXEC
File 3	VM/SP HPO 6 Sample Files, Product Parameter File (5664173H $PPF)
File 4	VM/SP HPO 6 CNTRL files, EXECs, and loaderfiles
File 5	VM/SP HPO 6 Textfiles
File 6	VM/SP HPO 6 MACLIB
File 7	Miscellaneous system files
File 8	VM/SP HPO 6 HELP files
File 9	AUX files for VM/SP 6, VM/SP HPO APARs, VM/SP HPO 6 updates (for ASSEMBLE files), Update shells for OCO modules
File 10	PTF source for VM/SP 6 APARs, VM/SP HPO 6 updates (for ASSEMBLE files)
File 11	AUX files for VM/SP APARs, VM/SP HPO 6 updates (for COPY/MACRO files)
File 12	PTF source for VM/SP 6 APARs, VM/SP HPO 6 updates (for COPY/MACRO files)
File 13	VM/SP HPO unique CP source
File 14	VM/SP HPO CP Delta code
File 15	VM/SP HPO CP Apply code
File 16	Spool file migration update files for UPDMIG42, Migration aid for users going from VM/SP HPO 4.2 to VM/SP HPO 6

In the following procedure, it is assumed that:

- The minidisk for VM/SP HPO sample files is 095
- The minidisk for VM/SP HPO CP Text is 196
- The minidisk for CMS is MAINT's 190
- The minidisk VM/SP HPO CP AUX and apply code is 096
- The minidisk VM/SP HPO CP source is 396
- The minidisk VM/SP HPO delta code is 094
- The minidisk HELP in mixed case English is 19D
- The minidisk VM/SP HPO MACLIB is 196
- The minidisk for VM/SP HPO 6 updates and VM/SP HPO 6 APARs is 196.

Do not change the minidisk address unless you have a thorough working knowledge of the system and are aware of the implications. There are no safeguards in the system to check the validity of your change.

Starter System Tape Layout

Table 5 shows the layout of the starter system tape.

Table 5. Starter System Tape Layout

Contents
Device Support Facility Program DMKFMT – Format/Allocate Service Program
DMKDDR – DASD Dump Restore Service Program
Starter System

Tape layouts are subject to change. You should always refer to the Program Directory (available from the HONE system through your branch office) for the most current format.

Feature Numbers

When you order VM/SP HPO starter systems, you must specify tape density and DASD type. For example, if your DASD configuration consists of 3380 direct access storage devices and 3420 tape drives, you should order a starter system tape with 6250 user bytes per inch (BPI) for restoration to the 3380. Feature numbers are available to assist you in ordering the starter system tapes for your system.

Starter System Feature Numbers

Table 6 lists valid starter system feature numbers. All starter systems include the VM/SP HPO product tape. As mentioned previously, if you have an operational system, a starter system is not necessary.

Table 6. Starter System Feature Numbers

Release 6	Description
5847	3350 starter system 6250 BPI
5848	3350 starter system 3480 Cartridge
5850	3375 starter system 6250 BPI
5021	3375 starter system 3480 Cartridge
5853	3380 starter system 6250 BPI
5854	3380 starter system 3480 Cartridge

Feature numbers for the 6250 BPI starter system include the 6250 BPI product tapes. Feature numbers for the 3480 cartridge starter system include the 3480 cartridge product tapes.

Starter system feature numbers are shipped in two tape formats.

Product Feature Numbers

Because the product tape does not have to be restored to a DASD unit, you do not have to be concerned about DASD types when ordering; however, you should order the correct tape density. To order the VM/SP HPO Release 6 product tape and control program listings, use the features numbers listed in Table 7.

Table 7. VM/SP HPO Release 6 Product Feature Numbers

Release 6	Description
5844	Product Tape 6250 BPI
5845	Product Tape 3480 Cartridge
8488	Control Program Listings – UP (microfiche)
8489	Control Program Listings – AP/MP (microfiche)

Table 8. National Language Support Feature Numbers

Release 6	Description
5305	Spanish 6250 BPI
5306	Spanish 3480 Cartridge
5331	Kanji 6250 BPI
5332	Kanji 3480 Cartridge
5335	French 6250 BPI
5336	French 3480 Cartridge
5339	German 6250 BPI
5340	German 3480 Cartridge

System Maintenance

IBM periodically distributes a Program Update Tape (PUT) to aid you in maintaining your system. The PUT you receive contains cumulative service for all licensed programs and system control programs for which your installation is licensed. A cover letter accompanying the PUT describes the procedure necessary for printing the machine-readable Memo to Users. This document describes what the PUT contains and how to apply service to your system.

There are instances, however, when your installation may need to apply an update to an individual module or apply a fix that is not included in the PUT. In addition, you may wish to modify your system to suit a special need. The *VM/SP HPO Installation Guide* provides procedures you can follow to manually update your system using established service routines.

Testing Period

The testing period is 2 months for a basic license.

License

A separate license is required for each designated machine on which the licensed program materials will be used except as otherwise provided by IBM.

Program Services

Central service, including the IBM Support Center, for the basic license will be available until discontinued by IBM upon 12 months′ written notice under the terms and conditions of the Agreement for IBM Licensed Programs. Central service, including the IBM Support Center, for DSLO licenses will be provided only through the customer location designated for the basic license.

Local licensed program support is available under the terms and conditions of the Agreement for Local Licensed Program Support for IBM Licensed Programs at the monthly licensed program support charge, or monthly multiple licensed program support charge, or will be provided at the applicable IBM hourly service rate.

With the availability of National Language Features worldwide, any problem that a customer believes is related to the use of a supported IBM program must be reported in the language normally used by the IBM Support Center to ensure the proper level of product service and support.

Warranty

VM/SP HPO is warranted to conform to its Licensed Program Specifications when shipped to the customer if properly used in accordance with the "Specified Operating Environment" section of the Licensed Program Specifications.

Licensed Program Specifications may be updated from time to time, and such updates may constitute a change in the specifications.

Following the discontinuance of all program services, this program will be distributed on an "As Is" basis without warranty of any kind either express or implied.

Part 2. Overview of VM/SP Release 6

VM/SP HPO Release 6 requires installation of VM/SP Release 6 or an equivalent licensed program. When you merge VM/SP HPO with the prerequisite VM/SP release, you obtain a system that incorporates all features from VM/SP and additional features from VM/SP HPO.

Note: VM/SP HPO has changed some the VM/SP Release 6 installation procedures. See part 1 for a discussion of these changes.

This part of the manual introduces you to VM/SP Release 6. It discusses the base VM/SP Release 6 including:

- Shared file system
- Extended architecture support for CMS
- Connectivity
- Callable services library
- Installation and service
- Other enhancements.

Chapter 6. Shared File System

The *Shared File System (SFS)* is an extension of the CMS file system that gives you an alternative to personal and shared minidisks for managing and sharing CMS files. In SFS, files are stored in file pools.

File Pools

A file pool is a collection of minidisks in which many users can create, manipulate, and store CMS files. You must be enrolled in a file pool to use SFS functions. As an enrolled user, you can be given exclusive access to an amount of logical file space. Your file space is independent of any minidisks that you can access. You can have file space and minidisks and move files from one to the other.

By assigning logical space in a file pool, SFS conserves system resources, because:

- Minidisk space in the file pool is dynamically allocated; that is, minidisk space is allocated only as you "fill" your logical space with files, and this allocated space can be distributed among several minidisks in the file pool.
- The size of the file pool (the number and size of the minidisks) can be maintained close to actual user requirements (which may be less than actual user allocations). If more space is needed, minidisks can be added to the file pool.
- Efficient use of the file pool minidisks reduces the amount of real DASD space required.

A system can have many file pools. You can have only one file space in a file pool, but you can have space in more than one file pool. You can define a default file pool for your virtual machine by including the name of the file pool on the IPL statement in your system directory entry or by adding a SET FILEPOOL command to your PROFILE EXEC.

Structure of a File Pool

A file pool contains three types of data:

- *Control data* consists of three parts:
  - The *filepoolid* POOLDEF file contains information that SFS uses to locate the minidisks in the file pool. Internally, SFS builds and executes FILEDEF commands based on the information in the POOLDEF file.
  - The control minidisk contains a map of all the 4K blocks of user and catalog data stored in the file pool.
  - The catalog storage group consists of one or more minidisks that contain information about the data that exists in the file pool, such as who owns it and who is authorized to look at it.
- *Log data* consists of two identical log minidisks that SFS uses to protect the integrity of the control data and user data. On each log minidisk, SFS maintains a record of changes to the file pool. If the system fails in the middle of an SFS operation, SFS can recover by referring to the log. Maintaining duplicate logs protects the file pool even if there is a problem with one of the logs. To protect against device errors, each log minidisk should be defined on a separate DASD volume.

- *User data* resides in collections of minidisks called storage groups. Each storage group can contain any number of minidisks, as long as the total number of storage group minidisks (including the catalog storage group) does not exceed 32767. When a user is enrolled in a file pool, the user is assigned to a storage group. A user cannot be assigned to more than one storage group in a file pool. File space in the storage group is allocated to the user in 4KB blocks. The blocks that correspond to a user's file may be distributed among the minidisks that are allocated to that storage group.

File Pool Management and Operation

A file pool is managed by a server virtual machine. A server consists of a set of programs that execute in a virtual machine to manage a resource, such as a file pool. Requests from a CMS virtual machine for data located in a file pool are sent across an Advanced Program-to-Program Communication (APPC) link to the appropriate file pool server. These requests can originate either as user commands or as function calls from a program. The server can be located in the same processor as the user or in another processor in the same Transparent Services Access Facility (TSAF) collection. With APPC/VM VTAM Support (AVS), the server can be located in another TSAF collection.

To manage the resources of a file pool, an SFS administrator uses the FILEPOOL EXEC and special CMS commands that require administrator authority. To operate a server machine, a server operator uses the FILESERV EXEC and special commands that require access to the virtual console of the file server machine. An administrator with access to the virtual console of the server can perform both functions. Administrator/operator tasks include:

- Generating a file pool
- Starting and stopping the server machine
- Adding minidisks to the file pool
- Enrolling and deleting users from the file pool
- Granting and revoking administrator authority
- Querying file pool status and user limits
- Backing up and restoring file pool data
- Conducting internal and external trace processing.

Hierarchical Directories

SFS lets you organize your files in directories. A top directory is automatically defined for you when you are enrolled in a file pool. Below the top directory you can use the CREATE DIRECTORY command to create up to eight levels of subdirectories in a hierarchy or "tree" structure. Files can reside at any level.

You assign each directory a name up to 16 characters long, but directory names do not have to be unique within the hierarchy. For example, if you have directories named TOM and MARY in your hierarchy, you can create a subdirectory named SALES in each one. You can identify a specific directory in the hierarchy by using the fully qualified directory name (DIRNAME). The format of the DIRNAME is:

filepoolid:userid.dir1.dir2.dir3.dir4.dir5.dir6.dir7.dir8 where *filepoolid* is the name of the file pool, (equivalent to a TSAF resource ID), *userid* is the name of your top directory (the same as your user ID[4]), and *dir1* through *dir8* are the names of the subdirectories in that "branch" of the hierarchy. The file pool ID must be followed by a colon; the top directory name must be followed by a period; and subdirectory names must be separated by periods.

For example, if the file pool ID is POOL1 and your user ID is DAVIS, then POOL1:DAVIS. is the dirname of your top directory. If TOM and MARY are subdirectories in DAVIS, and each contains a subdirectory named SALES, you use the following dirnames to identify them:

```
POOL1:DAVIS.TOM.SALES
POOL1:DAVIS.MARY.SALES
```

You access directories with the ACCESS command in the same manner as minidisks and assign each accessed directory a file mode letter. The QUERY ACCESSED command displays information about accessed directories and minidisks. You can remove a directory with the ERASE command.

Most existing CMS commands that display information about minidisks have been modified to include information about directories. For example,

- QUERY DISK and QUERY SEARCH display information about the accessed minidisks and directories.
- FILELIST and LISTFILE display the names of the files on a minidisk or the names of the files and subdirectories in a directory.

If a default file pool is defined on the IPL statement in your system directory entry, and if you are enrolled in that file pool, then your top directory in the file pool is automatically accessed as file mode A when you log on (even if you have a 191 minidisk defined in your entry in the system directory). If the access of the top directory fails for some reason (for example, if the file pool is not available), and if a 191 minidisk is defined in your system directory entry, then the 191 minidisk is accessed as file mode A.

Aliases

Additional images of a file stored in a directory can appear in other directories, and even in the same directory. The original file is called a base file, whether it resides in a top directory or in a subdirectory. The additional images of a base file are called aliases. With the CREATE ALIAS command, you can create aliases in other directories using the same name as the base file or different names, and you can create aliases in the same directory as the base file as long as you give them different names. The QUERY ALIAS command displays information about the aliases that have been created. You can remove an alias with the ERASE command. (The base file and other aliases are not affected).

Aliases have complete file IDs and behave just like base files. You can enter aliases in commands. Aliases appear in FILELIST and other file displays. Wherever you

[4] The name of your top directory in a file pool is the same as your user ID in the TSAF collection in which the file pool is located. Each user ID in a TSAF collection must be unique. If the file pool is located in a different TSAF collection from your virtual machine, your top directory name (user ID) in the file pool is your user identification in that TSAF collection.

can specify the name of a file, you can also use an alias. However, aliases do not occupy additional user storage. A base file and all its aliases represent the same file data; a change to any one is reflected in the others. In this respect, you can think of an alias as a pointer to a base file.

Sharing Files and Directories

Users can share information in SFS by granting each other read authority or write authority on base files, aliases, sets of base files and aliases, and directories. Write authority includes read authority. Granting authority on a base file also grants that authority on all aliases of the file. Granting authority on an alias is the same as granting that authority on the base file. With the GRANT AUTHORITY command, you can grant these authorities to a single user, to a group of users, or to all users. The QUERY AUTHORITY command displays information about the authorities that have been granted. You can revoke authority with the REVOKE AUTHORITY command.

Using these authorities, you can share files and/or directories in three ways:

- Create an alias.

  If another user grants you read authority or write authority on a file, you can create an alias of that file in one of your directories. Your authority for the alias is the same as your authority on the base file. Changes that other users make to the base file or to any alias of the base file are also reflected in your alias. If you have write authority for the alias, any changes that you make to the alias are reflected in the base file and all other aliases, even if you have no authority for the directory in which the base file resides.

- Access a directory.

  If another user grants you read authority on a directory, you can access that directory and look at the names of the base files, aliases, and subdirectories listed there. However, you cannot read the contents of any listed item unless the user also grants you read authority for that item. If the user grants you write authority for the directory, you can access the directory and create new base files and aliases there. However, you cannot modify any existing file unless the user also grants you write authority for that file.

- Read or modify a file directly through a program.

  If another user grants you read authority or write authority for a file, you can run a program (that utilizes the new SFS program functions) using the granted authority to read or modify the file. The directory containing the file does not have to be accessed, and authority for the directory is not required.

Concurrent Access to SFS Files

Under SFS, many users can have access to the same file at the same time. SFS permits multiple readers and one or more writers. Each reader sees the last committed version of the file. For example, if a reader has a file open, and a writer changes the file, the version that the reader sees is not affected by those changes. If another reader opens the file after the writer commits the changes, that reader sees the new version of the file. When the first reader closes the file, if the first reader was the only user of that level of the file, that level is discarded.

Locks

Because SFS files and directories can be shared by many users, CMS provides a way to lock files and directories to prevent simultaneous updates. When you are actively working with SFS files and directories, SFS automatically creates and deletes locks (called implicit locks). With the CREATE LOCK command, you can manually create a lock (called an explicit lock) on a file, a group of files, or a directory. You can select the type of lock and its duration.

There are three types of explicit locks:

share lock All authorized users can read the contents, but no one can make any changes.

update lock All authorized users can read the contents, but only you can make changes.

exclusive lock Only you can read the contents and make changes.

There are two lock durations:

session lock The lock lasts until the end of the CMS session or until the connection with the server is broken, unless you specifically delete it.

lasting lock The lock lasts until you specifically delete it.

The QUERY LOCK command displays information about the locks that have been created. You can delete a lock with the DELETE LOCK command.

Administrator Authority

Administrator authority gives you access to the resources of a file pool. With this authority, you can add minidisks to the file pool, enroll and delete users, and so on. You do not have to be an enrolled user to be an administrator.

Administrator authority also gives you "quasi-ownership" of every object stored in the file pool. As an administrator, you automatically have write authority on all base files, aliases, and directories in the file pool, and you can do anything with them that their owners can do. For example, you can create and delete base files, aliases, and directories. You can grant or revoke read authority or write authority on any object. You can create and delete locks. By the same token, you cannot do anything that the owners cannot do. For example, you cannot create an alias of user A's base file in user B's directory unless user A has already granted at least read authority on the base file to user B. However, as the "quasi-owner" of the base file, you can grant the necessary authority and then create the alias.

It is important to remember that administrator authority is not retained by your user ID. If you create a file in another user's directory while you are an administrator, you have write authority for the file as long as you remain an administrator. However, if you lose your administrator authority, you also lose access to the file, unless you granted yourself (your user ID) authority for the file while you were an administrator, or the real owner of the file grants you authority for the file.

SFS in the Full-Screen Environment

You can obtain information about your files and directories in the full-screen environment. The DIRLIST command lists the names of all your directories, whether or not they are accessed. From the DIRLIST display you can use a PF key to call:

- AUTHLIST to display the authorities granted on a listed directory
- FILELIST to display the files and subdirectories contained in a listed directory.

From the FILELIST display you can use a PF key to call:

- AUTHLIST to display the authorities on a listed file or directory
- ALIALIST to display the aliases of a listed file
- FILELIST to display the files and subdirectories in a listed directory. From this FILELIST, another PF key returns you to the FILELIST of the parent directory or to the DIRLIST.

By using FILELIST within FILELIST, you can move up and down through your directory structure.

CMS Commands

SFS functions are provided through new and enhanced CMS commands. Most existing CMS commands work with SFS files with no changes in syntax. Many CMS commands now accept an SFS directory identifier (*dirid*). The *dirid* can be in any one of the following formats:

- The file mode letter of an accessed directory
- A fully qualified directory name (*dirname*). Letters A-Z are valid characters for the first character of the dirname. The rest of the characters in the dirname can be A-Z or 0-9.

Usage Note: The directory does not have to be accessed.

- A plus (+) modification of a file mode that indicates a directory at a lower level in the hierarchy than the accessed directory. For example, if one branch in your hierarchy is POOL1:DAVIS.TOM.SALES.FIRSTQUARTER and you have POOL1:DAVIS.TOM accessed as B, then you can use the +B.SALES.FIRSTQUARTER abbreviation to indicate the FIRSTQUARTER subdirectory.
- A minus (-) modification of a file mode that indicates a directory at a higher level in the hierarchy than the accessed directory. For example, if another branch in your hierarchy is POOL1:DAVIS.MARY.SALES.FIRSTQUARTER and you have POOL1:DAVIS.TOM accessed as B, you can use the abbreviation -B.MARY.SALES.FIRSTQUARTER to "back up" one directory from TOM to DAVIS and then proceed to the FIRSTQUARTER subdirectory in the other branch.

Note: Commands specifically related to minidisks, such as FORMAT, do not work with SFS files and directories.

Using the System Product Editor with SFS Files

You can use the System Product Editor (XEDIT) to browse, update, or create SFS files in any directory you have accessed, as long as you have the corresponding authority.

- You can use XEDIT to browse through a file if you have at least read authority for the file and at least read authority for the directory where it resides.
- You can use XEDIT to modify a file if you have write authority for the file and at least read authority for the directory.
- You can use XEDIT to create a file if you have write authority for the directory. You automatically have write authority for any file you create. However, the owner of the directory is the real owner of the file, and can change or revoke your authority.

Program Functions

SFS includes new program functions for manipulating files stored in file pools. These program functions are available as routines in the callable services library (CSL) named VMLIB, supplied with your system. These routines

- Do not work with files stored on minidisks.
- Can be called from high-level language programs, such as FORTRAN, or COBOL, as well as from REXX or assembler language programs. No assembler language macros are provided for the new SFS functions.
- Accept either *dirnames* or *namedefs*. A *namedef* is a temporary name that you can use in a program in place of a hardcoded file ID or *dirname*.
- Provide functions such as:
  - OPEN, READ, WRITE, and CLOSE for files
  - OPEN DIRECTORY, GET DIRECTORY, and CLOSE DIRECTORY
  - CREATE ALIAS
  - CREATE DIRECTORY
  - CREATE LOCK and DELETE LOCK
  - GRANT AUTHORITY and REVOKE AUTHORITY.
- Help ensure the integrity of SFS files by requiring the program to explicitly commit (keep) or roll back (undo) all changes. If the program fails before the changes are explicitly committed, the file and/or directory remains unchanged.

Using the System Product Interpreter with SFS

You can process SFS files with System Product Interpreter (REXX) programs. You can issue CMS commands and call SFS functions from REXX programs. The SFS program functions are available as routines in the VMLIB callable services library (CSL). You can call these routines by coding a CSL external function call in your program.

Work Units

During SFS operations, requests from a CMS virtual machine to a file pool, whether through CMS commands, SFS program functions, or FS macros, are combined into groups called work units. The requests associated with a single work unit are all committed or rolled back at the same time. A CMS virtual machine can have more than one active work unit, and multiple work units can be active in a file pool. A work unit can also be associated with more than one file pool, as long as write or change activity is limited to a single file pool.

A work unit is identified by a work unit ID. CMS always uses the default work unit ID for commands and FS macros. However, the new SFS program functions let you specify a work unit ID. You can also change the default work unit ID. There are four program functions directly related to work units:

GET Work Unit ID	Gets a work unit ID if the default is not to be used
Push Default Work Unit ID	Makes a work unit ID the new default
Pop Default Work Unit ID	Removes the latest default work unit ID and restores the previous default work unit ID.
Purge Work Unit IDs	Purges all work unit IDs obtained by Get Work Unit ID.

Data Security

SFS includes the following features to protect data security:

- To access a file pool, you must be authorized (enrolled) by someone with administrator authority for that file pool.
- If an administrator gives you a file space in a file pool, you are the only one (other than an administrator) who can create files in that file space, unless you specifically grant this authority to another user.
- You control access to your files and directories by granting and revoking read authority and write authority.
- Only the owner or administrator of a directory can delete the directory.
- Implicit and explicit locks prevent simultaneous updates.

In addition, an external security manager (ESM) can replace SFS authorizations for those objects protected by the ESM.

Default Filemode A

New IPL Parameter

With the Shared File System, you can have your file mode A on the 191 minidisk or in a shared file. When you IPL CMS in a virtual machine, you can specify the location of your file mode A using a new parameter, FILEPOOL *filepoolid*, on the IPL control statement or the IPL command. Your top directory in the file pool is accessed as file mode A.

DIAGNOSE Code Enhancement

DIAGNOSE code X'B0' now returns information that CMS can use to determine the location of your file mode A. In addition to the re-IPL information, DIAGNOSE code X'B0' also returns from the directory the IPL statement that can contain your file pool specified for. If you specify a file pool, your top directory is accessed as file mode A. If you do not specify a file pool, the 191 minidisk is file mode A.

CP Interface for Application Monitor Data

If it is authorized to use DIAGNOSE code X'DC' by the new APPLMON option in the CP directory, the Shared File System uses this DIAGNOSE code to start the recording of SFS performance-related data in the MONITOR file. DIAGNOSE code X'DC' with subcodes X'00' and X'01' can control start and stop data recording. The APPLDATA MONITOR call class must be enabled to record this data in the MONITOR file.

The VMMAP program product can format Class A records that are built by the Shared File System for performance recording. User-generated data is available for VMMAP user-exits only. You must format your own data.

Reference

For more information about SFS, refer to the following books:

- The *VM/SP CMS User's Guide* describes how to use the SFS file management and file sharing functions.
- The *VM/SP CMS Command Reference* describes the new and enhanced general user CMS commands.
- *VM/SP CMS Shared File System Administration* describes the SFS server machine and the commands and functions available to the SFS administrator.
- The *VM/SP Application Development Guide for CMS* describes how to develop programs that use the SFS program functions.
- The *VM/SP Application Development Reference for CMS* describes the new SFS program functions and the enhanced FS macros.
- The *VM/SP System Product Interpreter Reference* describes how to code the CSL external function call from a REXX program.
- The *VM/SP HPO Planning Guide and Reference* and the *VM/SP HPO CP General User Command Reference* describe FILEPOOL and the APPLMON directory option.
- The *VM System Facilities for Programming* describes DIAGNOSE code X'B0' and DIAGNOSE code X'DC'.

Chapter 7. Extended Architecture Support in CMS

CMS has been enhanced to include bimodal CMS program interfaces. With these assembler interfaces you can code an application that is architecture-independent; it can be executed in a System/370 or 370-XA environment. You can develop applications on VM/SP HPO to exploit the IBM System/370 Extended Architecture when the application is run on Virtual Machine/Extended Architecture™ System Product (VM/XA™ SP). Programs that run in VM/XA SP can be written and compiled on VM/SP but require VM/XA SP to fully test the 31-bit capabilities of the program.

CMS assists application development through the following features:

- The CMS programming interface enables both 24-bit and 31-bit applications to execute on CMS. These interfaces are compatible in both VM/SP and VM/XA SP in System/370 mode and 370-XA mode virtual machines.

  Note: A VM/SP virtual machine is analogous to a System/370 mode virtual machine in VM/XA SP.

- Program link-edit and load facilities support module relocation at program loading time.
- CMS storage management manipulates free storage with a subpooling scheme similar to the MVS storage management subpool scheme. This allows CMS to manipulate subpools of storage as single entities. See "Storage Management" on page 94.
- The CMS OS/MVS environment supports the following:
  - Simulation of selected MVS/XA™ functions. The OS macros previously shipped with CMS are replaced with their equivalent MVS/SP™ 2.2.0 and Data Facilities Products Release (DFP) 2.3.0 versions.
  - Execution of the DFP Release 2.3.0 Linkage Editor containing AMODE and RMODE support.

In addition, CMS assists application development for VM/XA SP 370-XA mode through the following features:

- Program handling functions to support 31-bit addressing
- The CMS storage management facility to support and exploit 31-bit addressing
- The CMS programming interfaces to support programs that require more than 16 megabytes of storage for program execution or data storage
- The programming interfaces that can exploit the 370-XA architecture when running MVS/XA macros in a 370-XA mode virtual machine.

Note: IN VM/XA SP, CMS supports CMS/DOS, VSE/VSAM, and the EDIT command in System/370 mode virtual machines *only*.

Virtual Machine/Extended Architecture (VM/XA), MVS/XA and MVS/SP are trademarks of the International Business Machines Corporation.

31-Bit Addressing

System/370 architecture uses 24-bit addressing. This means that in VM/SP HPO (and in a VM/XA SP System/370 mode virtual machine), you can address up to 2^{24} - 1 (about 16 million) bytes of storage.

The 370-XA architecture uses 31-bit addressing. This means that, logically, you can address up to 2^{31} - 1 bytes of storage. In practice, the amount of real storage you have is limited by the processor unit you use. The amount of virtual storage you can address is limited by the operating system you run. VM/XA supports 370-XA mode virtual machine storage sizes of up to 999MB.

Bimodal Addressing

CMS supports *bimodal addressing*. This allows programs running in a VM/XA SP 370-XA mode virtual machine to execute in 24-bit addressing mode, 31-bit addressing mode, or a combination of both. Programs that have addressing sensitivities can run in a VM/XA SP 370-XA mode virtual machine without being converted to 31-bit addressing.

To support bimodal addressing, CMS recognizes two new program attributes: addressing mode (AMODE) and residency mode (RMODE).

Note: VM/SP supports System/370 architecture only and therefore does not support programs that use 31-bit addresses. In VM/XA SP, System/370 mode virtual machines simulate System/370 architecture; therefore they do not support programs that use 31-bit addresses either. The RMODE and AMODE attributes are ignored when a program runs in a VM/SP HPO or VM/XA SP System/370 mode virtual machine.

Addressing Mode (AMODE)

Addressing mode refers to the type of address (31-bit or 24-bit) a program expects to handle when it receives control. A program's AMODE attribute determines its addressing mode:

- AMODE 24 means a program can handle 24-bit mode addresses only. An AMODE 24 program must reside below the 16MB line.
- AMODE 31 means a program can handle 31-bit mode addresses. An AMODE 31 program can reside above or below the 16MB line.
- AMODE ANY means you are deferring the decision to assign the program an addressing mode. There are several points in the program cycle when you can assign an AMODE or override an existing AMODE. You can also use AMODE ANY to let the program's addressing mode default to the value of the program that called it.

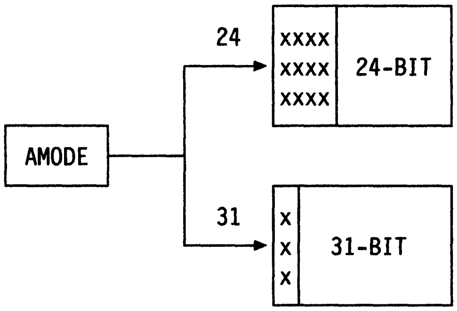


Figure 4. How CMS Interprets the AMODE Attribute. The AMODE attribute is used by the CMS supervisor-assisted linkage when the program is called.

Residency Mode (RMODE)

Residency mode refers to where a program resides when CMS loads it (above or below the 16MB line). A program's RMODE attribute determines its residency mode:

- RMODE 24 indicates that CMS loads the program below the 16MB line.
- RMODE ANY indicates that CMS loads the program above 16MB unless insufficient storage is available above the 16Mb line. Use RMODE ANY for 31-bit or undefined programs.

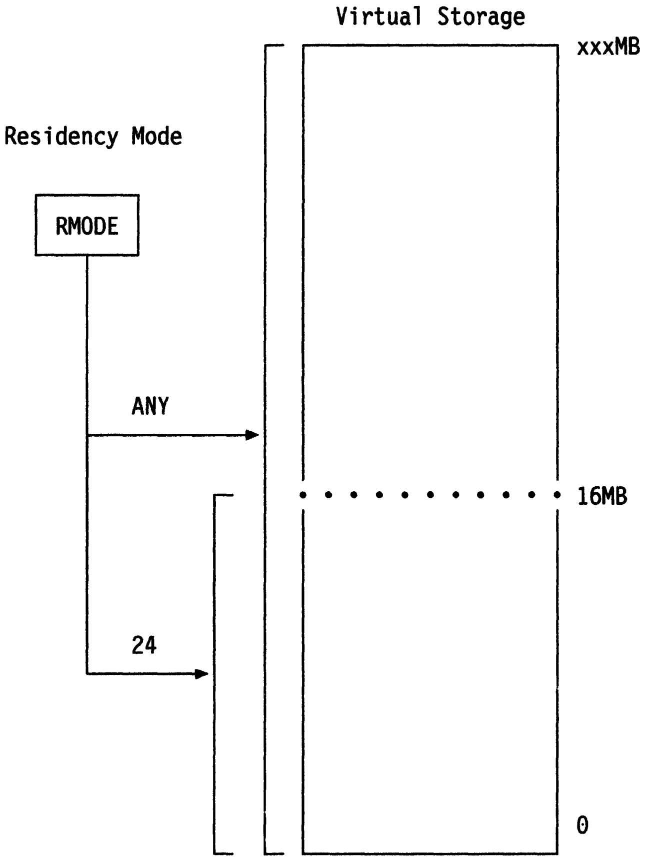


Figure 5. How RMODE Affects Where CMS Loads Programs. The RMODE attribute is used by the CMS program loader.

4-Digit Device Addresses

Command Changes: The following CMS commands are enhanced to support 4-digit virtual device addresses.

ACCESS
FILEDEF
FORMAT
LANGMERG
RELEASE
QUERY
TAPE
VMFPLC2

Macro Changes: The following macros are enhanced to support 4-digit virtual device addresses.

DEFNUC
APPLMSG
RDTAPE
TAPECTL
TAPESL
WRTAPE

DEBUG Command Replacement

The DEBUG command returns the following information:

- The PSW at the time of an ABEND
- General registers at the time of an ABEND
- Floating point registers at the time of an ABEND
- The vector status register if Vector Facility is in use
- The following fields at the time of an ABEND
  - The external old PSW
  - The SVC old PSW
  - The program old PSW
  - The machine-check old PSW
  - The input/output old PSW.

DEBUG subcommands and the DEBUG subcommand environment have been removed. The function removed from DEBUG is available through the CP DISPLAY, TRACE, and PER commands.

The CMS Programming Interface

The CMS has defined programming interfaces that are architecturally independent. The CMS programming interfaces consist of the following defined groups:

- CMS preferred interface group
- CMS compatibility group
- OS/MVS and DOS simulation interface group.

All interfaces in CMS that are not part of the above three groups are part of the CMS internal interface group. These interfaces are for CMS's use only. Using items from this group may adversely affect the portability and ease of migration from release to release of CMS application programs.

Figure 6 presents the relationship among the above groups. A description of the groups follows the figure.

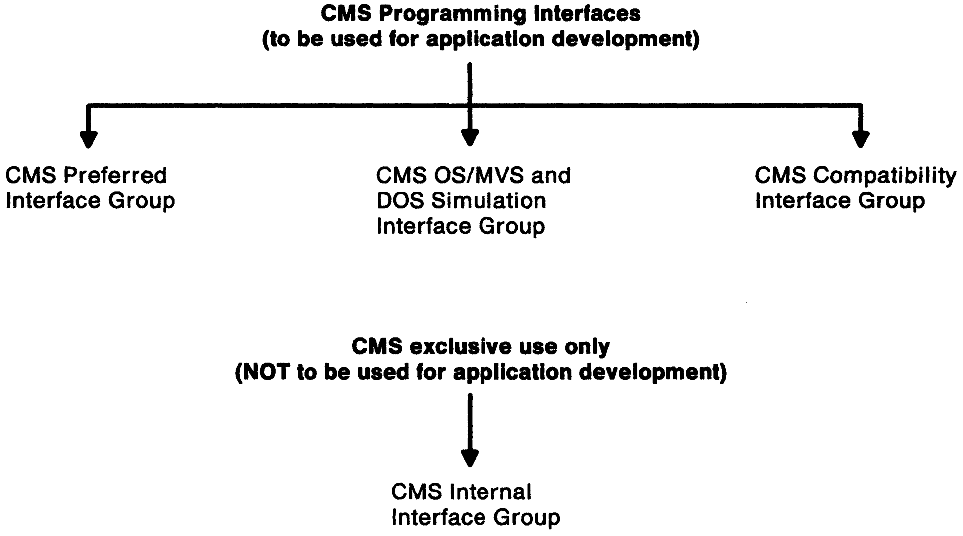

CMS Programming Interfaces
(to be used for application development)
CMS Preferred Interface Group
CMS OS/MVS and DOS Simulation Interface Group
CMS Compatibility Interface Group
CMS exclusive use only
(NOT to be used for application development)
CMS Internal Interface Group

Figure 6. Relationship of CMS interface groups to each other

The Preferred Interface Group

These macros, routines, and functions make up the heart of the CMS programming interface. They provide you with a means of making program calls, managing storage, performing I/O, handling interrupts, and processing abends. They run in System/370 mode virtual machines and 370-XA mode virtual machines; in 370-XA mode virtual machines they support 24-bit and 31-bit addressing. They help you avoid architecture-constrained facilities like I/O instructions, they reduce your need to reference CMS internal data areas and control blocks, and they make it easier for you to develop programs that are portable across architectures.

IBM encourages you to use preferred interface macros in your CMS application programs.

CMS Compatibility Group

These are macros, functions, and services that CMS maintains for compatibility with previous releases. Existing programs that use interfaces in the compatibility group can run in System/370 mode virtual machines or in 24-bit addressing mode in 370-XA mode virtual machines. Compatibility group interfaces cause unpredictable results in 31-bit addressing mode.

For new programs, IBM recommends that you use interfaces in the preferred group rather than interfaces in the compatibility group.

OS/MVS and DOS/VSE group

These are macros also provided by the OS/MVS and DOS/VSE operating systems. CMS supports these macros to make it easier to run on CMS programs developed for OS/MVS or DOS/VSE. The OS/MVS and DOS/VSE group consists of the following sub-groups:

1. Simulated OS/MVS macros: These are OS/MVS macros that CMS simulates[5] and that you can use in your programs. While these macros provide some portability between VM and OS/MVS systems, the CMS simulation of these macros is *not* necessarily the same as the current MVS support. CMS simulates only a selected subset of OS/MVS macros and, because of operational differences in VM and MVS, macros that are supported may work differently between the two systems. For complete information on how to use OS/MVS macros, you may need to refer to OS/MVS publications.

   For CMS application programs, IBM recommends that you use macros in the preferred group rather than OS/MVS macros.

2. Non-simulated OS/MVS macros: You can use these macros to develop and compile programs for execution on MVS systems; however, because CMS does not simulate these macros, programs that use them will not run on CMS.

3. DOS/VSE macros: These are DOS/VSE macros that CMS simulates. You can use them in System/370 mode only. Note that the CMS simulation of these macros is not necessarily the same as the current DOS/VSE support.

   For CMS application programs, IBM recommends that you use macros in the preferred group rather than DOS/VSE macros.

Note: CMS macros, control blocks, and functions that are not part of the defined programming interface are considered CMS internal interfaces. They should not be used by programs other than CMS.

[5] Note that CMS actually simulates the **function** the macro provides, not the macro itself.

CMS Command and Macro Support of Extended Architecture

New and Changed CMS Macros

Table 9 lists the new and significantly changed CMS macros, describes the functions they provide, and describes the functions (if any) they replace or supersede. Note that in addition to the macros in the following figure, several macros have been enhanced to support 31-bit addressing and/or 4-digit virtual device addresses.

Table 9. New and Changed CMS Macros

Macro	Replaces	Function
AMODESW	Branch instructions (BAL, BALR, BSM, BASSM)	AMODESW provides an architecture-independent method of branching. It calls programs, switches or sets a program's addressing mode, and queries a program's addressing mode.
CMSCALL, CMSRET	SVC 202 SVC 203	Supervisor-assisted linkage mechanism. (It calls and returns from programs).
CMSSTACK	ATTN function	Places data on the program stack.
CMSSTOR	DMSFREE, DMSFRET	Obtains and releases free storage.
CONSOLE	Full-screen I/O interfaces.	Provides an architecture-independent interface to the CMS console facility. Additionally, in CMS the console facility supports the 370-XA channel subsystem, addresses greater than 16MB, and 3270 graphics printers.
ENABLE	SSM instruction	Enables and disables the PSW for various interrupt types.
GETSID	No System/370 equivalent.	Stores a device's subchannel identification number (SID) in register 1. GETSID is required in a 370-XA mode virtual machine only.
HNDEXT	HNDEXT	Allows you to define interrupt handlers for specific external interrupt codes, allows you to specify whether an event control block (ECB) is posted, and allows you to define a default external interrupt handler.
HNDIO	HNDINT (*see note*)	In a 370-XA mode virtual machine, HNDIO returns device-related information that HNDINT does not. (Note that HNDINT does support 31-bit addressing.)
HNDSVC	HNDSVC	Allows you to save SVC handlers across end-of-command processing and abend processing, allows you to pass a user word to the handler routine.
NUCEXT	NUCEXT function	Declares, clears, and queries nucleus extensions.
SCAN		Creates tokenized and extended parameter lists from input data.
SUBPOOL		Manages free storage subpools.
SUBCOM	SUBCOM function	Defines, clears, and queries subcommand environments.

Note: HNDIO replaces HNDINT in that it returns device related information HNDINT does not. HNDINT has been enhanced to support 31-bit addressing, but otherwise works the same as before.

New and Changed CMS Commands

Table 10 lists the new and significantly enhanced CMS commands.

Table 10. New and Changed Commands to support bimodal 31-bit Addressing

Command	Changes/Notes
ACCESS	Supports 4-digit device addresses.
CMSBATCH	Supports 4-digit device addresses.
FILEDEF	Supports 4-digit device addresses.
FORMAT	Supports 4-digit device addresses.
GENMOD	GENMOD now allows you to specify addressing mode (AMODE) and residency mode (RMODE) values for a module. It also allows you to restrict modules to either System/370 or 370-XA architecture.
INCLUDE	Supports 31-bit addressing.
LOAD	LOAD now allows you to specify addressing mode (AMODE) and residency mode (RMODE) values for a text file.
LOADMOD	Supports relocatable modules and 31-bit addressing.
PROGMAP	PROGMAP is a new command that allows you to obtain the name, entry point, origin, addressing mode, and attributes of programs loaded by the LOAD or LOADMOD commands. PROGMAP aids in debugging.
QUERY	Supports 4-digit device addresses.
RELEASE	Supports 4-digit device addresses.
SET LDRTBLS	SET LDRTBLS defines the number of pages used for loader tables. Now you can issue it at anytime. In previous releases, it had to be issued immediately after IPL.
SET LOADAREA, QUERY LOADAREA	The SET LOADAREA command helps users to specify where CMS loads programs. The QUERY LOADAREA command displays the current SET LOADAREA setting.
SET STORECLR, QUERY STORECLR	The SET STORECLR command specifies when CMS returns GETMAIN storage and how CMS handles user invocations of STRINIT. The QUERY STORECLR command displays the current setting of SET STORECLR.
TAPE	Supports 4-digit device addresses.

Program Invocation

As mentioned earlier, CMS provides two new macros that you can use to call other programs and, if necessary, switch addressing modes:

- CMSCALL requests supervisor assisted linkage to another program.
- AMODESW is an architecture-independent replacement for direct branch instructions such as BAL, BALR, BAS, BASR, BSM, and BASSM.

Supervisor Assisted Linkage — The CMSCALL and CMSRET Macros

In prior releases, CMS used the SVC 202 and SVC 203 instructions to perform supervisor assisted linkage. Now, CMS uses the CMSCALL and CMSRET macros. While SVC 202 and SVC 203 continue to work from below the 16Mb line only, CMSCALL and CMSRET work from any program; therefore, for any program that might run above the 16MB line, you must use CMSCALL and CMSRET instead of SVC 202 or SVC 203.

AMODESW Macro

If you need a program switching mechanism that works in either a System/370 mode or 370-XA mode virtual machine, use the AMODESW macro.

Use the AMODESW macro to:

- Do address switching as a part of subroutine calls and returns. Programs can use the AMODESW macro as an architecture-independent version of the BALR or BASSM instruction for branching to other subroutines within a module.
- Switch a program's current addressing mode to 24-bit or 31-bit. The AMODESW macro eliminates the need to call a “glue” subroutine for operations in 31-bit mode. (A glue subroutine is one that handles the switch between 24 and 31-bit programs.)
- Save the current addressing mode when switching to a new mode. This is useful when (a) a program does not know the current mode, (b) wants to make sure it is in a particular mode, and (c) eventually wants to return to the original mode.
- Determine the current addressing mode.

SCAN Macro

The new SCAN macro provides 31-bit capable, formal capability interface for the assembler programmer to create tokenized and extended parameter lists from input data. In addition to setting up the input lists, SCAN stores the address of the extended parameter list in register 0 and the address of the tokenized parameter list in register 1.

Storage Management

CMS storage management has been enhanced to support 31-bit addressing, to improve free storage management, and to use a subpooling scheme similar to the OS/MVS subpooling scheme. CMS includes two new macros for storage management (CMSSTOR and SUBPOOL) and two new commands (SET STORECLR and QUERY STORECLR). These functions coexist with the DMSFREE, DMSFRET, DMSFRES, and STRINIT macros. DMSFREE and DMSFRET work only under 16MB. CMS performs DMSFRES functions internally; however, to maintain compatibility with existing applications, the DMSFRES macro remains but is not operative.

CMS OS/MVS Simulation Storage Management Routines

CMS no longer has separate routines to simulate the OS/MVS storage management macros GETMAIN and FREEMAIN. Instead, CMS translates each GETMAIN and FREEMAIN request into a corresponding CMS storage request (CMSSTOR OBTAIN or CMSSTOR RELEASE). The GETMAIN and FREEMAIN macros still work; however, using the CMSSTOR and SUBPOOL macros is faster and more efficient.

CMS also returns GETMAIN free storage in a different manner than it did in prior releases. If your programs depend on other programs (invoked via SVC 202 or CMSCALL) to obtain *and* return GETMAIN free storage, you may need to use the SET STORECLR command.

Storage Layout

CMS now organizes virtual machine storage in a different manner. Previously, CMS set up a storage area called the user area. The user area is now free storage. It starts above the transient program area and it extends up to the limit of the virtual machine size (excluding any storage CMS uses). The following figure shows a sample storage layout.

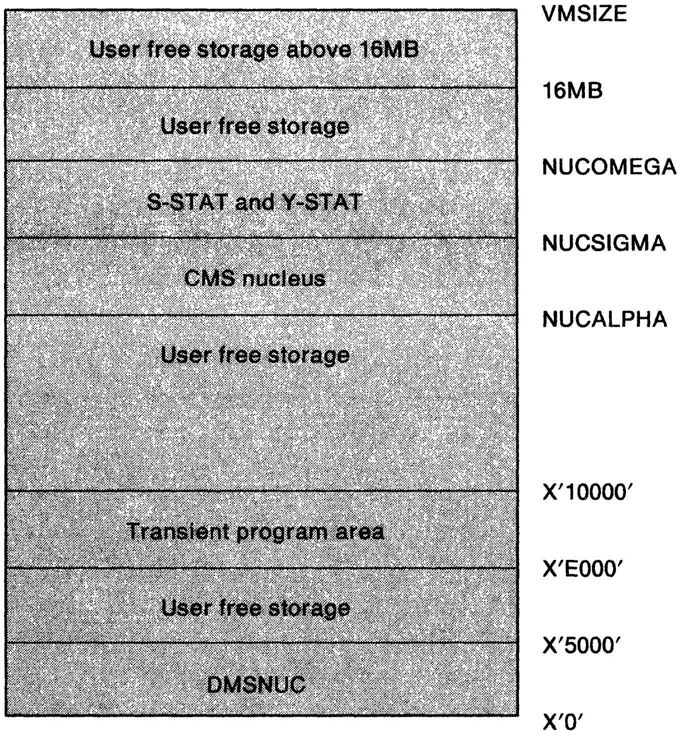


Figure 7. Storage Map of a CMS System

Note: You cannot obtain storage that crosses the 16MB line.

Managing Subpools

In prior releases, user storage was split into storage obtained by the DMSFREE macro and storage obtained by the GETMAIN macro. Now, CMS obtains all user storage, including storage requested by the GETMAIN macro, from the next available piece of free storage large enough to satisfy the request. This means that, CMS native free storage and OS GETMAIN simulated free storage are no longer segregated and the fields MAINHIGH and FREELOWE are no longer used. In earlier releases, MAINHIGH and FREELOWE were boundaries between the two storage management areas. Programs that are dependent upon these internal fields must adjust accordingly.

You can use two new macros, SUBPOOL and CMSSTOR, to manage subpools; however, to the application or system programmer, subpooling can be transparent. You do not have to use the SUBPOOL macro or the SUBPOOL parameter on the CMSSTOR macro. CMS can automatically satisfy free storage requests from the USER subpool, in the way a call to DMSFREE worked in earlier releases.

CMSSTOR and SUBPOOL Macros

The CMSSTOR and SUBPOOL macros work together. Programs must issue CMSSTOR explicitly to obtain and release storage; subpool management works whether programs specify the SUBPOOL macro or not.

Interrupt Handling

CMS interrupt handling routines support the 370-XA environment and make it easier for application programs to handle interrupts (or easier for programmers to code interrupt handlers). The following sections summarize the changes to CMS interrupt handling.

Machine Check Interrupt Handling

The CMS machine check handler is enhanced in two ways:

- 370-XA I/O configuration updates: During initialization in a 370-XA mode virtual machine, CMS (a) examines the virtual I/O device configuration, (b) enables all defined virtual subchannels, and (c) saves the subsystem-identification words that correspond to the permanent DEVTAB (device table) entries. Whenever the I/O device configuration of the virtual machine is updated, CP generates a channel report word and makes pending a machine check interrupt to the virtual machine (assuming that the virtual machine has not set bit 13 of the PSW and bit 3 of control register 14 to 0 to disable this class of machine checks).

  The machine check handler accepts the channel report word and then examines the CMS device tables (DEVTAB, the user I/O table, or CDEV, the console device table). If the device is there, the machine check handler updates the table. If the device is not there, CMS saves the subsystem ID for later use. When devices are detached, they are also released and the second-level interrupt handler for the device is cleared.

- Vector Facility machine checks: The Vector Facility creates two new machine check conditions. How CMS handles the conditions depends on the particular virtual machine and the type of interrupt:

  - Vector Facility failure: This means that the Vector Facility is no longer usable. After examining the interrupt, CMS returns control to the program that was running. The Vector Facility failure does not affect a program unless it attempts to execute a vector instruction. If a program issues a vector instruction *after* a Vector Facility failure, CMS generates a program interrupt and resets the vector control bit in control register 0.

- Instruction processing damage: This causes the Vector Facility to perform a validation process that destroys its current status and makes the current operation invalid. The active program cannot recover from the instruction failure; however, it can continue execution if another Vector Facility in the processor complex is active or if the program does not issue any more vector operations. For this reason, you may want to use the ABNEXIT macro to create an abend exit routine. In response to Vector Facility instruction processing damage, CMS generates an X'1F3' abend code.

Program Interrupt Handling

OS/MVS simulation routines support ESPIE. This allows you to specify program interrupt exits in 24-bit or 31-bit addressing mode. CMS still simulates SPIE, but in System/370 mode virtual machines only.

I/O Interrupt Handling

Three CMS macros provide I/O interrupt handling support: the CONSOLE macro (for applications doing full-screen I/O to 3270 devices), the HNDINT macro, and the HNDIO macro. CONSOLE and HNDINT are enhanced to handle 31-bit addresses. HNDIO, a new macro, also handles 31-bit addresses; in addition it uses keyword parameters that allow a programmer to obtain complete I/O status (INTBLOK parameter), retain specific handlers past end-of-command and/or abend processing (SYSTEM and KEEP parameters), pass a user word (UWORD parameter), and create reentrant parameter lists (MF parameter).

HNDIO Macro

Use the HNDIO macro to handle interrupts and, for a 370-XA mode virtual machine, to obtain complete I/O interrupt status for specified I/O devices. For devices on a System/370 mode virtual machine, HNDIO returns the same interrupt information as HNDINT.

To use HNDIO to obtain complete interrupt status, use the HNDIO INTBLOK parameter to provide an area where HNDIO can copy the interrupt status. The INTBLOK control block describes the format of the interrupt status.

HNDINT Macro

The HNDINT macro has been enhanced to support 31-bit addressing; otherwise, it works the same as before. CMS calls the second-level handler routine in the addressing mode that was in effect at the time the HNDINT macro was issued.

External Interrupt Handling

In prior releases, CMS allowed programs (or users) only one external interrupt routine to handle all interrupt codes. This limitation caused application developers to “steal” the external new PSW and replace it with a version specific to their needs. Enhancements to CMS external interrupt handling (in particular, enhancements to the HNDEXT macro) eliminate the need to steal the external new PSW. CMS enhancements to HNDEXT allow you to:

- Define interrupt handling routines for specific external interrupt codes. These routines can exploit 31-bit addressing.
- Define an event control block (ECB) for specific interrupt codes; this allows a program to use the WAITECB macro to detect the external interrupt event.

- Define a default interrupt handling routine to process any interrupt that does not have a specific handler.
- Specify that handlers survive end-of-command processing and abend processing.

Upon completion, your routine can specify through its return code that (a) processing is complete, or (b) the first-level interrupt handler should post the ECB defined for this interrupt code.

The Default External Interrupt Handler

CMS has a default interrupt handler that processes the external interrupts that do not have specific handlers. The default handler returns the interrupt code and generates a message that asks if it should (a) ignore the interrupt and resume processing or (b) pass control to the CMS abend routines. By contrast, previous releases put the virtual machine in the DEBUG environment when no interrupt handler was defined.

SVC Interrupt Handling

The HNDSVC macro now supports 31-bit addressing and has new parameters that allow you to (a) specify that SVC trap routines survive end-of-command and/or abend processing and (b) pass an optional fullword to the SVC trap routine.

CMS IUCV Support

CMS IUCV external interrupt handling now includes a second-level handler. CMS invokes the IUCV second-level handler when it detects an X'4000' IUCV interrupt. The IUCV second-level handler then passes control to the address you specify on the HNDIUCV macro EXIT= parameter in support of PENDING CONNECT interrupts. For other than PENDING CONNECT interrupts, the IUCV second-level handler passes control to the address specified on the CMSIUCV macro EXIT= parameter.

Program Handling

CMS program handling services are enhanced (or created) to support 31-bit addressing, relocatable modules, compatibility with the MVS/XA load process, and the more efficient use of free storage.

The SET LOADAREA Command

The SET LOADAREA command adds another dimension to your ability to control the CMS program loader; you can use it to help determine where CMS loads programs (specifically, programs for which you do not specify an ORIGIN).

The SUBCOM and NUCEXT Macros

The SUBCOM macro defines, clears, and queries a subcommand environment. The NUCEXT macro defines, clears, and queries a nucleus extension. SUBCOM and NUCEXT are new macros: they support 31-bit addressing, they are independent of the PSW, and they provide a more usable interface to the existing SUBCOM and NUCEXT functions.

GENMOD Command

Use the GENMOD command to create a module from TEXT files you have loaded with the LOAD or INCLUDE command. The changes to the GENMOD command are:

- The AMODE operand of the GENMOD command allows you to specify an addressing mode (24, 31, or ANY) for the program. The value specified on the GENMOD command overrides the AMODE previously set (during assembly or as part of the LOAD/INCLUDE process).
- The RMODE operand specifies the residency mode (24 and ANY). For relocatable modules, the RMODE operand overrides any RMODE values previously set. If the module is non-relocatable, the module will be loaded based on the location of the text files when the module was created.
- The 370 and XA operands allow you to specify that a module run only in a particular mode (System/370 mode or 370-XA mode) virtual machine. If you do not specify 370 or XA, the module can run in any virtual machine.
- The CLEAN and NOCLEAN options allow you to specify when a module is cleaned from storage.

LOADMOD Command

Use the LOADMOD command to load a module into storage. The changes to the LOADMOD command are:

- The LOADMOD command now supports relocatable modules and 31 bit addressing CMS can relocate any module for which relocation dictionary (RLD) information has been saved (during LOAD).
- The LOADMOD command requests free storage according to a module's residency mode (RMODE).
- LOADMOD continues to support non-relocatable and transient area modules.
- The PRES and NOPRES options allow you to specify whether or not you want the currently loaded programs to be preserved in storage.
- If the module being loaded was generated with the STRINIT option, then CMS deletes and returns the storage used by each program currently loaded in storage through the LOADMOD, LOAD, or INCLUDE commands, or through OS/MVS or DOS simulation.

PROGMAP Command

The PROGMAP command is new and can help you debug programs. It allows you to obtain the name, entry point, origin, addressing mode, and relocation attributes of programs that you load through LOAD, INCLUDE, LOADMOD, or OS Simulation.

SET LDRTBLS Command

The SET LDRTBLS command defines the number of pages to be used for loader tables. In prior releases of VM/SP, it had to be issued immediately after IPLing CMS. Now you can enter SET LDRTBLS at anytime. Note however, that the SET LDRTBLS command resets the loader table; any information that was in the loader table prior to the SET LDRTBLS command being issued is no longer available.

OS/MVS Simulation

OS/MVS simulation allows you to use MVS/XA macros in your application program. For selected MVS/XA macros, CMS provides your application program with results equivalent to what the macros would provide running natively on MVS/XA.

OS/MVS simulation is an enhancement of existing CMS OS macro simulation. The OS macros previously shipped with CMS are replaced with their equivalent MVS/SP Release 2.2.0 and DFP Release 2.3.0 versions.

Although OS/MVS simulation supports 31-bit addressing, there are still MVS/XA restrictions (MVS/XA running natively) that the CMS simulation honors. (For example, DCB addresses must reside below the 16MB line.)

In addition to native MVS/XA restrictions, you should also be aware of software differences between MVS/XA and CMS that still apply (for example, MVS/XA supports multitasking but CMS does not).

CMS uses the MVS/XA macros to simulate a specific subset of MVS/XA functions. These are listed and described in *VM/SP Application Development Reference for CMS*. These MVS/XA macros are part of the CMS programming interface described in "The CMS Programming Interface" on page 89.

In addition, CMS provides selected MVS/XA macros for assembly purposes only.

CMS provides an interface that lets you code your own simulation routine in your application program if you need an MVS/XA function that CMS does not simulate.

The OSMACRO and OSMACRO1 macro libraries are refreshed and reorganized and retain their old names to aid you in migrating your existing CMS programs utilizing OS simulation to the enhanced CMS OS/MVS simulation interfaces.

The MVS/XA macros are organized in the macro libraries in the following manner:

- **OSMACRO MACLIB** - This MACLIB contains only the simulated subset of the MVS/XA macros. These macros are modified, where necessary, to be System/370 and Assembler F compatible. These macros are intended for programmers who wish to use OS/MVS simulation but do not plan to migrate their programs to an MVS environment.
- **OSMACRO1 MACLIB** - This MACLIB contains both the simulated subset of MVS/XA macros and unsimulated (for assembly only) MVS/XA macros. These macros do *not* contain modifications for System/370 and Assembler F compatibility. Many of these macros use Assembler H features and contain 370-XA assembler mnemonics such as BSM and BASSM. These macros are intended for programmers who wish to use OS/MVS simulation in a 370-XA mode virtual machine only and to transport their programs to an MVS environment.

Reference

For more information about Extended Architecture support, refer to the following books:

- The *VM/SP Application Development Reference for CMS* describes the CMS programming interface, the CMSSTOR and SUBPOOL macros, and interrupt handling macros.
- The *VM/SP Application Development Guide for CMS* describes storage management macros, using the CMS interrupt handling services, interrupt processing, the HNDIO, HNDINT, and CONSOLE macros, and OS/MVS simulation restrictions.
- The *VM/SP CMS Command Reference* describes the new and enhanced program handling commands.
- The *VM/SP Application Migration Guide for CMS* describes the changes made to CMS to support the IBM System/370 Extended Architecture and the changes (if any) that you should make to your programs for System/370 compatibility, 370-XA toleration, and 370-XA exploitation.

Chapter 8. Connectivity

Connectivity has been enhanced by the following:

- Addition of APPC/VM VTAM (AVS) Support
- Addition of common programming interface (CPI) communications
- APPC/VM enhancements
  - Private resource support
  - Security support for connections
  - Mapped conversation support
- Addition of CMS communications directory
- TSAF enhancements
- Pass-Through use of APPC/VM

Common Programming Interface for Communications

In VM/SP Release 6, you can write APPC application programs in REXX or a high-level language (such as C) that communicate with other programs. In simplest terms, this communication consists of a source program that wants a resource from a target program.

Systems Application Architecture Common Routines

Most of the routines you will need are provided by IBM's Systems Application Architecture Common Programming Interface (CPI) Communications. Programs using Systems Application Architecture CPI Communication routines can easily be moved to other systems that support Systems Application Architecture.

The basic set of CPI Communications routines allow applications written in REXX or high-level languages to connect to each other, send and receive data, and disconnect from each other. In addition, more advanced routines allow programs to set or extract parameter values before making a call to a CPI-Communications routine.

The CPI-Communications routines are described in the *VM/SP SAA Common Programming Interface Communications Reference.*

VM Extensions to CPI Communications

VM provides additional routines that extend CPI Communications. These routines allow for server programs to (1) handle more than one conversation at a time and (2) provide additional security capabilities. These extended communications routines are described in a VM appendix in the *VM/SP SAA Common Programming Interface Communications Reference.*

Application Programs

All the CPI Communications routines, including the extended communications routines, are stored in VMLIB, the callable services library supplied with VM/SP. When a CPI-Communications routine is called, it internally calls APPC/VM functions. If the conversation is to go out on an SNA network, the new AVS (APPC/VM VTAM Support) component “translates” the call request into APPC (LU 6.2) protocol.

Applications using these communication routines can be written for mapped or basic conversations.

APPC/VM VTAM Support

APPC/VM VTAM Support (AVS) is a VM/SP component that lets programs in a TSAF collection or single VM system communicate with programs in a Systems Network Architecture (SNA) network. This communication is achieved using APPC (LU 6.2) protocol.

AVS, along with VTAM, handles communications between a TSAF collection of VM systems and systems in an SNA network. It allows APPC/VM or CPI-Communications programs that run in in a TSAF collection to communicate with APPC programs that run in either VM or non-VM systems in an SNA network. This allows programs in different TSAF collections to communicate with each other. In addition, APPC programs running in the SNA network can connect with with resources in the TSAF collection.

SNA View of the TSAF Collection

APPC/VM VTAM Support, along with VTAM, provides the general SNA network with a view of the TSAF collection. The TSAF collection is defined to the SNA network as one or more logical units (LUs). These LUs, defined by the AVS operator to represent the collection to VTAM, are called gateways in VM. These gateways carry on communications with the remote LUs in the SNA network for the APPC/VM and CPI-Communications programs in the TSAF collection. With the proper authorization, APPC programs in the SNA network can connect to global and private resources in the TSAF collection, without the network user being logged onto a system in the VM collection. Multiple users can access resources through a gateway at the same time.

AVS Virtual Machine

AVS is a separate component of VM/SP that runs with VTAM in a Group Control System (GCS) group. The AVS virtual machine:

- Manages the gateways defined by its operator
- Provides VM - VTAM communications
- Is controlled by AVS commands.

The AVS virtual machine communications with other VM systems in the TSAF collection through TSAF. It communicates with VTAM by transforming APPC/VM commands to APPC/VTAM (the VTAM implementation of APPC), and vice versa.

APPC/VM

APPC/VM now supports private resources, connection security, and mapped conversations.

Private Resource Support

Private resources, like global and local resources, are located on a local or remote system in a TSAF collection. These resources are not identified to the TSAF collection or the local system. CMS provides support so that the private resource manager can register the private resources it manages along with a list of authorized requesters. The private resource name only needs to be unique within the private resource manager virtual machine.

Unlike local and global resource managers, the private resource manager virtual machine does not need to be logged on when a program requests to be connected to one of its private resources. If the private resource manager virtual machine is not logged on and its directory entry contains an IPL statement, CP will autolog it.

Security Support for Connections

The following security features have been added:

- Support to allow user ID and password to be specified by the requester
- Authorization lists for access to private resources
- An indicator on the connection to *IDENT to specify the type of security information required from the requester for a connection to be made to the resource.

Support for Mapped Conversations

In addition to supporting basic conversations, APPC/VM now includes support for mapped conversations. Using mapped conversations, application programs can exchange data in an arbitrary format, as long as both sides of the communication agree to that format.

CMS Communications Directory

The CMS Communications Directory enhances the usability and security of VM communications. The directory allows APPC/VM programs to specify a symbolic name to connect to a resource rather than resource ID and the locally known name. Symbolic names allow requestor programs to communicate transparently within a TSAF collection or across an SNA network.

A communications directory is a special NAMES file that assigns symbolic names to actual resources. There are two levels of communications directory. One is referred to as the *system level* communications directory. This file is a general purpose communications directory that can be accessed by users on a system-wide basis. The second directory is referred to as the *user-level* communications directory. This file is the primary file accessed by the communications directory, as it contains user-specific communications information.

These two directory types enhance the usability of VM connectivity by allowing requestor programs to specify simply a symbolic name. Since the location of the resource is transparent to the requestor programs, resources can be moved to another resource manager without the necessity to change the requestor program. Only the communications directory needs to be updated.

For more information on communications directories, see *VM/SP Connectivity Planning, Administration, and Operation.*

TSAF Enhancements

The TSAF component has been enhanced through more support in the Collection Resource Management (*CRM) and the Identify System Service (*IDENT).

*CRM now supports private resources, and provides validation for user IDs and passwords within the collection. *IDENT now lets authorized virtual machines connect to it as gateway managers and to notify the TSAF virtual machines of changes to the resource/gateway table. A new QUERY GATEWAY command has been added.

Pass-Through Use of APPC/VM

In VM/SP, Pass-Through (PVM) uses the APPC/VM protocols to communicate with its counterparts on other VM systems. This support lets an installation have VTAM or TSAF manage the communications between PVM machines that would otherwise require their own independent physical links.

Reference

For more information about connectivity, refer to the following books:

- The *VM/SP Connectivity Programming Guide and Reference* describes APPC/VM.
- The *VM/SP Connectivity Planning, Administration and Operation* describes AVS and TSAF.

Chapter 9. Installation and Service

Enhancements to installation and service include:

- Enhancement of the Installation Procedures
- Enhancement of the Service Process
- Addition of Starter System Support for 3380 DASD Models E4 and K4
- Addition of the Product Parameter File
- Addition of the CMSVMLIB Saved Segment
- Addition of the CMSFILES Saved Segment
- Enhancement of the DMKLD00E LOADER
- Addition of the GENSERVE EXEC
- Enhancement of the GENTSAF EXEC
- Enhancement of the Primitive Functions Performed by the ITASK EXEC
- Enhancement of the UPDATE MODULE
- Enhancement of the UTILITY EXEC
- Addition of the VMFAPPLY EXEC
- Enhancement of the VMFASM EXEC
- Addition of the VMFBLD EXEC
- Addition of the VMFHASM EXEC
- Enhancement of the VMFLOAD MODULE
- Addition of the VMFOVER EXEC
- Enhancement of the VMFPLC2 MODULE
- Addition of the VMFREC EXEC
- Addition of the VMFSETUP EXEC

Enhancement of the Installation Procedures

The following steps have been added to the Starter System and existing VM/SP HPO system procedures for installing VM/SP HPO:

- Building the system file pool (VMSYS)

  This step is required because GCS, TSAF, and AVS code is loaded to directories defined in VMSYS.

- Building the user file pool (VMSYSU)

  This step is optional but recommended.

- Loading the APPC/VM VTAM Support (AVS) component.

  This step is optional.

DELTA service code (update files and serviced text decks) and APPLY service code (auxiliary control files) for each component is shipped on the product tape and loaded during installation to establish a service base.

Installation procedures have been added for the new CMSVMLIB and CMSFILES physical saved segments.

Enhancement of the Service Process

The service process has been enhanced to provide a compatible service control system for objects serviced by update and objects serviced by replacement. These enhancements include the following:

- New tools are provided for loading the files from the service tape, applying the PTFs, and rebuilding the serviced objects.
- Parameters for receiving service, applying service, and building objects are centralized in the product parameter file.
- Program update tapes and corrective service tapes have similar tape formats and can be received with the same tool.
- Service tapes contain replacement text decks for object-serviced objects and update files and replacement text decks for source-serviced objects.
- The service tape contains multiple service levels.
- A PTF parts list is shipped on the service tape for each PTF. The parts list contains the file name and file type of each part included in the PTF package.
- Documentation in the text decks, text deck shells, update files, and update shells provides PTF and APAR tracking.
- Requisite information (PREREQs, CO-REQs, and IF-REQs) is included in the prologs in text decks and text deck shells.
- The file type of text decks and text deck shells contains the PTF number.
- The file mode number of the update file indicates its status: 1 means not "applied"; 5 means "applied".
- A new single path service procedure is provided for program update service and corrective service.

Addition of the Product Parameter File

The product parameter file is a multipart file that contains installation and service parameters for a product. These parameters are used by various installation and service programs. The file name of the product parameter file is the product identification number (*prodid*) and the file type is $PPF. For example, the product parameter file supplied with VM/SP Release 6 is called 5664167E $PPF, while the product parameter file supplied with VM/SP HPO Release 6 is called 5664173H $PPF.

The product parameter file contains several specialized areas:

- A product parameter area
- One or more component parameter areas (one for each component in the product
- Optionally, one or more component parameter override areas.

The product area lists the component areas and override areas in the file.

A component area has three sections:

- The first section contains control options, such as the control file name, the system level and version, the national language, and so on.

- The second section assigns specific minidisks and/or CMS Shared File System (SFS) directories to the following symbolic strings:

SYSTEM	working system environment
BASE	source and object code
DELTA	update files
APPLY	auxiliary control files
LOCAL	local service files
BUILD	where the new system is built.

- The third section is a part type or function list that identifies specific execs for receiving service, applying service, and building each object in the component.

You can tailor the product parameter file by creating override areas, which are similar to component areas. An override area has a name that relates to a specific component. It contains parameters that update or replace parameters in the corresponding component area. The VM/SP product parameter file includes CP override areas for building an attached processor nucleus or a multiprocessor nucleus. The VM/SP product parameter file also includes an override area for each component to define corrective service parameters.

You can also define override areas in separate product parameter override files. This is the recommended tailoring method. An override file can have any file name (except *prodid*, which is reserved as the file name of the base product parameter file), but the file type must be $PPF. Each override area defined in the override file contains a pointer to the base product parameter file. When entering a command that requires the name of the product parameter file, you can specify the name of the override file instead.

Addition of the CMSVMLIB Saved Segment

A NAMESYS entry has been added to DMKSNT for a physical saved segment called CMSVMLIB that occupies the four 64K segments 172-175. CMSVMLIB is built with the SEGGEN EXEC and contains a single logical saved segment called VMLIB that is used to store the VMLIB callable services library (CSL). CMSVMLIB overlays CMSFILES, but these saved segments are never attached to the same virtual machine.

Addition of the CMSFILES Saved Segment

A NAMESYS entry has been added to DMKSNT for a physical saved segment called CMSFILES that occupies the sixteen 64K segments 160-175. CMSFILES is built with the SEGGEN EXEC and contains two logical saved segments called DMSDAC and DMSSAC that are used to store the DMSDAC and DMSSAC file pool server modules. If you have multiple SFS server machines on your system, you can improve performance by loading and saving these modules. CMSFILES overlays CMSVMLIB, but these saved segments are never attached to the same virtual machine.

Installation and Service Tools

Note: The SPGEN EXEC and SPGEN PROFILE are no longer supported.

Enhancement of the DMKLD00E LOADER

The DMKLD00E LOADER is invoked when you IPL 00C to build a new nucleus. The loader has been enhanced to process VERify and REPlace cards (control statements) contained in the text deck being loaded.

Addition of the GENSERVE EXEC

The GENSERVE EXEC builds CMS Shared File System (SFS) load modules by calling the PRELOAD, LOAD, and GENMOD commands to process the text decks in the corresponding loadlists. GENSERVE locates text decks by PTF number. You can specify the name of a control file to control the file types of the text decks searched for.

Enhancement of the GENTSAF EXEC

When building the RUNTSAF MODULE, the GENTSAF EXEC now locates text decks by PTF number.

Enhancement of the ITASK EXEC

The ITASK EXEC calls the VMFASM EXEC to do assemblies and calls the VMFBLD EXEC to do builds. Component operands have been added to the ASSEMBLE function. When assembling a CP file, you can specify CP, CPAP, or CPMP to indicate the type of system you are building (uniprocessor, attached processor, or multiprocessor, respectively).

The CPAP and CPMP operands have been added to the BUILD function to build an attached processor or multiprocessor system. The appropriate control file name is obtained from the corresponding CP override area in the VM/SP product parameter file.

The FILESERV function has been added to build CMS file pools. FILESERV VMSYS builds the required system file pool (VMSYS) and starts the system file pool server machine (VMSERVS). FILESERV VMSYSU builds the optional user file pool (VMSYSU) and starts the user file pool server machine (VMSERVU).

The AVS and LANG AVS operands have been added to the LOAD function to load AVS code from the product tape or AVS files from a national language feature tape.

See Chapter 2 for a description of the VM/SP HPO HITASK EXEC.

Enhancement of the UPDATE MODULE

The UPDATE MODULE has been enhanced to process PREREQ, CO-REQ, and IF-REQ statements in update files. PREREQs and CO-REQs describe prerequisites and co-requisites in the same product. IF-REQs describe dependencies in other products. UPDATE adds this information to the UPDATES file.

Enhancement of the UTILITY EXEC

The DTYPE operand has been added to the UTILITY EXEC for determining information about a specified DASD. This function is identical to the DTYPE function formerly available with the SPGEN EXEC, which is no longer supported.

See Chapter 2 for a description of the HUTILITY EXEC.

Addition of the VMFAPPLY EXEC

The VMFAPPLY EXEC creates auxiliary control files for text decks in the PTFs you want to apply to a specific component. VMFAPPLY obtains parameters for the component from the product parameter file or product parameter override file that you specify. When you invoke VMFAPPLY, you can also enter operands and options with the command to override certain parameters in the product parameter file.

Enhancement of the VMFASM EXEC

The VMFASM EXEC has new PPF and CTL options that let you specify whether VMFASM should obtain data from the product parameter file or from the control file. New SETUP and NOSETUP options let you specify whether or not VMFASM should call the VMFSETUP EXEC to set up minidisk and directory accesses. (The SETUP option is valid only when the PPF option is used.)

Addition of the VMFBLD EXEC

The VMFBLD EXEC obtains data from the product parameter file or a product parameter override file to set up a minidisk/directory access order for the build function, then calls part specific execs to:

- Apply patches
- Build a CP, CMS, or GCS nucleus
- Copy PTF-numbered text decks to the build minidisk/directory
- Build the RUNTSAF MODULE
- Build SFS load modules.

Addition of the VMFHASM EXEC

The VMFHASM EXEC performs the same function as the VMFASM EXEC except that VMFHASM invokes the H-Assembler. To use VMFHASM you must have the optional H-Assembler product installed on your system. Service updates are provided in object code for users who do not have the H-Assembler.

VMFHASM can and will cause assembler error messages and is not the suggested assembler for VM/370 or VM/SP HPO.

Enhancement of the VMFLOAD MODULE

The PTF option has been added to the VMFLOAD MODULE. Using this option directs VMFLOAD to search for PTF-numbered text decks. If the option is not specified, VMFLOAD searches for text decks using the level identifier in the control file.

Addition of the VMFOVER EXEC

The VMFOVER EXEC creates a temporary product parameter file that contains the parameters from a single component area of the base product parameter file. The file type of the temporary file is $PPFTEMP. If VMFOVER is passed the name of an override area instead of a component name, and the name of an override file (if it exists) instead of the base product parameter file, then VMFOVER merges the overrides with the contents of the temporary file. VMFOVER is called by VMFAPPLY, VMFASM, VMFBLD, VMFHASM, VMFREC, and VMFSETUP, or you can invoke it directly.

Enhancement of the VMFPLC2 MODULE

The BLKSIZE option has been added to the VMFPLC2 MODULE to let you specify the approximate size of the tape block to which files are dumped. The choices with this option are 4000, 32K, or 64K bytes.

The WTM and NOWTM options have been enhanced. WTM writes a tape mark between each CMS file when multiple files are dumped. When a single file is dumped or the last file is dumped, two tape marks are written and one is backspaced over. NOWTM does not write a tape mark after each CMS file. When a single file is dumped or the last file is dumped, two tape marks are written and both are backspaced over.

Addition of the VMFREC EXEC

The VMFREC EXEC maps the program update tape or corrective service tape and receives the service files to the targets specified in the product parameter file. The primary difference between VMFREC and the previous receive service tool, VMSERV EXEC, is that VMFREC operates on a part-specific basis instead of a product basis. For products that do not use a product parameter file, VMFREC calls the product-specific service exec from the service tape.

Addition of the VMFSETUP EXEC

The VMFSETUP EXEC sets up a minidisk and SFS directory access order according to parameters defined in the product parameter file or a product parameter override file. The ACCESS option accesses the minidisks and directories required for the assemble, receive, apply, or build function, or accesses all minidisks and directories. VMFSETUP is called by VMFASM, VMFHASM, VMFAPPLY and VMFBLD.

Chapter 10. Callable Services Library

The *callable services library (CSL)* is a facility that lets application programs written in REXX or high-level languages (such as PL/I) access VM-specific services without requiring programmers to write unique assembler routines.

VMLIB Routines

VMLIB is a callable services library that contains a set of routines. As an application programmer, you can call these routines, much like subroutines, to perform VM services without writing unique assembler subroutines.

The supplied VMLIB library contains routines that:

- Call shared file system functions
- Access the current generation of REXX variables
- Interface with the VM command environment through a REXX EXEC
- Invoke the CMS EXTRACT/REPLACE facility that allows application programs to obtain or modify selected system information without release and/or VM system dependencies
- Call CPI-Communications routines to perform program-to-program communications.

Supported Languages

You can call CSL routines from application programs written in the following languages:

- Assembler
- COBOL (includes OS/VS COBOL and COBOL II)
- VS FORTRAN
- VS Pascal
- PL/I
- REXX
- C.

VMLIB and any callable service library you create can be automatically loaded into storage when either the system profile or a user's PROFILE EXEC is invoked.

Callable service library routines can be stored on disk, in a shared file system directory, or in a logical saved segment using the new "saved segment management" feature.

EXTRACT/REPLACE

You can now extract or replace some system information (such as information about minidisks and directories, devices, OS simulation, files, active files, shared files, and general system information) in CMS control blocks by using the EXTRACT/REPLACE facility in your application programs.

You no longer have to know the names of specific control blocks and fields or exactly where the information is within CMS. In addition, the information returned by this facility is in a form that will be consistent from release to release. So, it will be easier for you to migrate your applications to new releases of VM/SP and VM/XA.

Reference

For more information about CSL, refer to the following books:

- The *VM/SP Application Development Reference for CMS* describes VMLIB routines except those for program-to-program communication.
- The *VM/SP Connectivity Programming Guide and Reference* describes program-to-program communication. The *VM/SP SAA Common Programming Interface Communications Reference* contains details on program-to-program communications routines.
- The *VM Application Development Guide for FORTRAN and COBOL*, the *VM/SP Application Development Reference for CMS*, and the *VM/SP Application Development Guide for CMS* describe EXTRACT/REPLACE.

Chapter 11. Other Enhancements

Other enhancements include:

- Control Program (CP) Enhancements
- Conversational Monitor System (CMS) Enhancements
- IPCS Enhancements
- 3090 Processor Resource/System Manager™ (PR/SM) Support

Control Program (CP) Enhancements

GIVE Command

If you are a System Resource operator, you can use the CP GIVE command to transfer a dedicated tape drive to another user on your system. When the receiver detaches the tape drive, you can optionally reclaim it.

Librarian and tape management systems such as VMTAPE can use this support. For example, all drives in a tape pool can be ATTACHed to the tape manager's virtual machine. Then, as mount requests are received, the tape manager can use the GIVE command to give the tape to the requesting user.

Previously, the tape manager had to DETACH the tape specifying the LEAVE option and the operator had to issue the ATTACH command to reattach the tape to the intended receiver. However, another user could ATTACH the tape before the intended receiver. With the GIVE command, you can be sure that the intended receiver gets the tape.

You can invoke the CP GIVE command by typing the command at your terminal or from a program via the CP diagnose X'08' interface. When invoked, control of the designated tape drive is transferred to the specified virtual machine. The receiving virtual machine then logically owns the tape drive until it is detached either via the CP DETACH command or by logging off the system.

Refer to the *VM/SP HPO CP General User Command Reference* for more information.

DASD Block I/O System Service

You can now read and write up to 256 contiguous or discontiguous CMS formatted FBA or CKD blocks with a single Block I/O request. A single CCW chain is created for all requested blocks on the same FBA or CKD cylinder within a multiple request. Individual blocks can be read and written within the same request, and the buffers used for these blocks do not have to be contiguous.

You must specify multiple Block I/O requests with a new request code in the IUCV parameter list.

3090 Processor Resource/System Manager is a trademark of the International Business Machines Corporation.

Note: The blocks can be physically read or written in an order different than that indicated by the application. Do not use this function to do I/O if you have to maintain a specified order of read or write.

Refer to the *VM System Facilities for Programming* for more information.

Performance Monitor

This support improves the quality and usefulness of system measurement data reported by the system monitor in two ways:

- The monitor data indicates when a virtual machine is waiting for a response from a server (such as SQL/DS or SFS). In addition, server virtual machines are identified.
- Previously, certain data fields were shared by the monitor code and the accounting code in CP. The ACCOUNT command reset some of these fields and gave no indication in the monitor data record that the fields had been reset.

  Now, the ACCOUNT command indicates in the monitor record that shared data has been reset.

Refer to the *VM/SP HPO CP General User Command Reference* for more information.

CPTRAP Facility

The CPTRAP facility aids field engineers and system programmers with problem determination. The facility allows you to define what events are traced and to collect vital system diagnostic data while the system is running.

Four types of traps are provided to help you diagnose system problems:

- The DATA trap lets you examine data conditions after execution of nearly any path in CP. You can use this trap to solve problems with communications between CP and guest operating systems and components in CP.
- The IO trap lets you examine all of the I/O activity to any devices or range of devices. For any transaction, CPTRAP records the CSW, the channel program, and, optionally, the data transferred. You can use this trap to solve problems with the interfaces between CP and I/O devices.
- The GT trap lets you record virtual machine events. You can use this trap to solve problems involved with guest virtual machines.
- The TTABLE trap lets you control which CP internal trace entries are collected. You can also alter the trace flags in the PSA with this trap type. You can use this trap to solve problems in CP code.

To help you solve specific problems, The CPTRAP command is enhanced so you can specify the data that you want CPTRAP to record. To see the status of the trapid and trapset that you are enabling, you can enter either the CPTRAP DISPLAY command or the QUERY CPTRAP command.

The CPTRAP command is a privilege class C, type P command. For security reasons, you can use the Resource Access Control Facility (RACF) to control who can use this command.

In addition to providing more function, the CPTRAP algorithms are refined to keep pace with faster trace table generation rates.

In addition, a new CP DIAGNOSE code, X'EC', recognizes a query of the CPTRAP facility. The query lets a virtual machine determine whether data it sends to the CPTRAP facility is accepted and placed in a CPTRAP file.

You can use the Interactive Problem Control System (IPCS) component of VM to access CPTRAP data. You can report and diagnose software failures and manage problem information and status.

Refer to *VM/SP HPO CP System Command Reference* for the syntax of the CPTRAP command. Refer to the *VM System Facilities for Programming* for more information about DIAGNOSE code X'EC'. Refer to *VM/SP HPO Interactive Problem Control System (IPCS) Guide* for additional information on the CPTRAP facility and how IPCS can be used to process CPTRAP files.

DIAGNOSE Code X'64' Subcode X'18'

A new subcode, X'18' is added to DIAGNOSE code X'64'. This subcode invokes the FINDSYS function, as does subcode X'000C', which finds the starting and ending address of the named system. Subcode X'0018' is intended to be used when the named saved system is defined as a Virtual Machine Group. with the VMGROUP=YES option of the NAMESYS macro. Subcode X'000C' does not allow a named segment to be a virtual machine group.

Refer to the *VM System Facilities for Programming* for more information.

Conversational Monitor System (CMS) Enhancements

Saved Segment Management

Overview

You can now view a physical saved segment as a set of logical saved segments. In addition, logical saved segments can now contain multiple objects such as modules, text files, execs, callable services libraries, language repositories, and user-defined objects or a single shared minidisk directory as supported by the SAVEFD command. For example a program that requires execs, CSL routines, and NLS message files can put all of these into one logical saved segment and issue one SEGMENT LOAD command or macro to access them.

You can put as many program objects as can fit into a single logical segment and as many logical segments as can fit into a physical saved segment.

New and Changed Commands

The new SEGGEN command builds and saves a physical saved segment made up of one or more logical saved segments. The QUERY SEGMENT command displays information about physical or logical segments. The SEGMENT command works with logical saved segments as well as physical saved segments and can associate logical segments with physical saved segments. Saved segments can be loaded within (or outside) the address space of a virtual machine. Saved segment space can also be reserved for subsequent loading. In addition, the SEGMENT and NOSEGMENT options are added to the NUCXMAP, PROGMAP, and EXECMAP, and the SEGINFO and ATTRIBUTES options are added to NUCXMAP.

Refer to the *VM/SP CMS Command Reference* for more information about the QUERY SEGMENT, SEGMENT, NUCXMAP, PROGMAP, and EXECMAP commands. Refer to the *VM/SP Application Development Guide for CMS* for more information about the SEGGEN command.

SET AUTODUMP and QUERY AUTODUMP

You can now control the creation of an automatic dump, which contains information about the DMSNUC region of CMS, storage management work area, page allocation table, and loader tables, if an abend occurs. Two new commands are SET AUTODUMP and QUERY AUTODUMP.

SET AUTODUMP

The SET AUTODUMP has three options:

SET AUTODUMP CMS lets you create an automatic dump for only unrecoverable CMS system abends.

SET AUTODUMP ALL lets you create an automatic dump for all abends occurring in your virtual machine.

SET AUTODUMP OFF lets you turn the creation of an automatic dump off.

QUERY AUTODUMP

The QUERY AUTODUMP command gives you the current setting of the AUTODUMP command.

Refer to the *VM/SP HPO Diagnosis Guide* for more information about these commands.

New QUERY DISK Option

The FIRSTR/W option has been added to the QUERY DISK command to let you search for the first filemode accessed as read/write. Search order is by filemode from A through Z. If no disks are accessed as read/write, you receive an error message.

Refer to the *VM/SP CMS Command Reference* for more information about QUERY DISK.

READCARD, DISK LOAD, and NETDATA Commands

When you use the READCARD and DISK LOAD commands, you can no longer overlay existing files without being notified.

New defaults for DISK LOAD and READCARD

The defaults for the DISK LOAD and READCARD commands are MINPROMPT and NOREPLACE.

NETDATA Command

There are three NETDATA commands: NETDATA RECEIVE, NETDATA SEND, and NETDATA QUERY. The defaults for the NETDATA RECEIVE command are LOG, OLDDATE, NOTEBOOK ALL, and MINPROMPT. The defaults for the NETDATA SEND command are LOG, TYPE, and NOACK. The NETDATA QUERY command does not have defaults.

Refer to the *VM/SP HPO Diagnosis Guide* for more information about these commands.

System Product Interpreter

REXX 3.45 Language Upgrade

The CMS System Product Interpreter has the following added features to support the REXX 3.45 language level:

- WORDPOS built-in function
- Civil and Normal options of TIME built-in function
- Normal option of DATE built-in function
- Nomatch option of VERIFY built-in function
- Failure option on SIGNAL instruction
- Failure option on TRACE instruction and TRACE built-in function
- VALUE option on NUMERIC FORM.
- DIGITS, FORM, and FUZZ built-in function
- Strict comparison operators < <, > >, ¬ < <, and ¬ > >
- Backslash character as a synonym for not character (¬)
- Enforcement of restriction on placement of PROCEDURE statement.

Refer to the *VM/SP System Product Interpreter Reference* for more information.

DBCS Support

REXX implements Double Byte Character Set (DBCS) function and handling techniques such as the following:

- New OPTIONS modes to handle DBCS not only as literal strings but also in data operations
- New external function package that adds 13 new DBCS functions
- Enhanced current instructions and functions to handle DBCS.

IPCS Enhancements

CPTRAP Files

IPCS now supports processing of CPTRAP files. To provide this support, the following IPCS commands have been enhanced:

- IPCSSCAN, previously DUMPSCAN, lets you view CPTRAP files as well as dumps. New subcommands let you scroll through a CPTRAP file and modify the selectivity of what you can view in the CPTRAP file or the CP DUMP trace table.
- IPCSPRT, previously PRTDUMP, prints CPTRAP files as well as dumps. You can specify the CPTRAP file you want to print using the IPCS problem number or using the spoolid of the CPTRAP file.
- APAR tells you what CPTRAP files are associated with the problem being sent to IBM.
- PROB associates a CPTRAP file with the IPCS problem number.
- IPCSDUMP associates a CPTRAP file with a dump you are processing.

In addition, now that CPTRAP files are supported by IPCS, another tape is needed to transfer spool files with the CP SPTAPE command. So, to make the transfer easier, the SPTAPE command lets the operator assign a new owner to the spool file as it is being loaded from tape.

Refer to the *VM/SP HPO Interactive Problem Control System Guide and Reference* for more information about IPCS.

3090 Processor Resource/System Manager (PR/SM) Support

VM/SP and VM/SP HPO support the 3090 Processor Resource/System Manager (PR/SM) environment. PR/SM is an enhancement for the 3090 processor that provides a logical SCP partitioning capability. This allows you to run multiple 370 Operating Systems on the 3090 processor.

The following functional enhancement supports the PR/SM environment:

- The control Program replaces the PR/SM processor address number with the real processor address number in VM/SP HPO machine check and channel check records.

Part 3. Migration Considerations

This part of the manual discusses:

- Migration to VM/SP HPO Release 6
- Performance considerations

Chapter 12. Migration Considerations

This section describes considerations for migrating to VM/SP HPO Release 6:

- Previous releases
- CMS growth
- CP changeable parts
- Installation and service
- Migrating spool files
- CMS shared file system
- Extended architecture support in CMS
- Connectivity
- VS COBOL II Programming Language
- CPTRAP
- Saved segment management
- IPCS enhancements
- 3380 Models AJ4/BJ4 and AK4/BK4
- 3990 Storage Control and 3380 Model CJ2
- DASD dump restore streaming support improvements
- Support for the IBM 3990 Storage Control Model 3
- National language support file naming conventions enhancements
- Security and integrity enhancement
- Support for mixed directory
- Support for VM/Directory Maintenance Licensed Program (DIRMAINT) Release 4
- Free storage enhancements.

Previous Releases

The text of HELP files and some messages have been changed. You might have to update existing execs and/or programs to reflect the changes in messages and HELP text.

CMS Growth

CMS has grown about 1MB. This may impact your shared segment layouts.

CP Changeable Parts

All VM/SP CP changeable parts are versioned and contained on the VM/SP HPO product tape.

Installation and Service

Consider the following where appropriate:

- To migrate from a release prior to VM/SP HPO Release 4.2, you *must* use the starter system procedure. To migrate from VM/SP HPO Release 4.2 or VM/SP HPO Release 5, the starter system procedure is recommended. However, if you do not have enough DASD space available to use the starter system, or if your system configuration differs substantially from the configuration defined in the sample system files, you can use the existing VM system procedure. This procedure requires a greater working knowledge of VM/SP HPO and the installation process. To use the existing VM/SP HPO system procedure, you must first load and apply the service APAR VM32788 (CP) and either APAR 32863 (CMS) for VM/SP HPO Release 4.2 or APAR VM34631 (CMS) for VM/SP HPO Release 5. These APARs contains updated UPDATE and VMFLOAD MODULEs, a modified DMKLD00E LOADER, and a new VMFLDS MODULE that are required to use the new service functions in VM/SP HPO Release 6.

- VM/SP HPO Release 6 includes substantial changes from previous releases. The *VM/SP HPO Installation Guide* highlights the changes that relate to installation and system generation.
- Before building a component, make sure that any minidisk (other than 190) containing previous-release files for that component is either not accessed or accessed lower (close to Z) than any minidisk or SFS directory used in that step of the procedure.

  Also, consider the following where appropriate:

  - Current VM/SP HPO Release customers may not use a 3278 Model 5, 3279 Model 3, or any terminal that supports greater than 24 x 80 screen sizes.
  - Current VM/SP HPO Release 3.2 customers with service level less than SL304 must not use a 3278 Model 5, 3279 Model 3, or any terminal that supports greater than 24 x 80 screen sizes.
- All VM/SP CP code has been versioned for VM/SP HPO Release 6.

Real Storage Requirements

VM/SP HPO requires at least 2 megabytes of real storage.

Migrating Spool Files

VM/SP HPO releases prior to VM/SP HPO Release 5 had a limit of 9900 spool files with system-unique identification numbers. VM/SP HPO Release 5 and later allows many more spool files, identified by the user ID and an identification number which is unique for that user, but which may be duplicated by another user on the same system. For example user A and user B may each have a spool file 0001. You should be aware of the changes in spool file support when you:

- Migrate to VM/SP HPO Release 6 from VM/SP HPO Release 4.2.
- Migrate back to VM/SP HPO Release 4.2 from VM/SP HPO Release 6.

To receive this support apply the following APARs to your VM/SP HPO Release 4.2 system:

- VM26856 (PUT 8703)
- VM27304 (PUT 8703)
- VM26477 (Prerequisite to VM27304)
- VM28427

See the *VM/SP HPO Installation Guide* for more information on migrating spool files.

CMS Shared File System

Installation

The VM/SP HPO Release 6 installation procedures include steps for building a system file pool (named VMSYS) and a user file pool (named VMSYSU). The system file pool is required; GCS, TSAF, and AVS code is loaded to directories defined for MAINT. The user file pool is optional.

The system file pool server machine (VMSERVS) and the user file pool server machine (VMSERVU) are defined in the sample CP directory. Sample file pool definition files are loaded to the MAINT 295 minidisk:

- VMSYS POOLDEF contains information about the location of the minidisks in the system file pool.
- VMSERVS DMSPARMS contains start-up parameters for the system file pool server.
- VMSYSU POOLDEF contains information about the location of the minidisks in the user file pool.
- VMSERVU DMSPARMS contains start-up parameters for the user file pool server.

In the DMSPARMS files, MAINT is defined as the first administrator of the file pool.

To build the system file pool, you invoke the ITASK EXEC installation tool with the FILESERV VMSYS operands. In addition to building the file pool and starting the server, ITASK enrolls MAINT as a user in the system file pool. To build the user file pool, you invoke the ITASK EXEC with the FILESERV VMSYSU operands. ITASK does not enroll MAINT as a user in the user file pool.

DIRNAME Syntax

Because the colon (:) and the period (.) are used as separators in the dirname syntax, and because the (+) and minus (-) signs are used in variations of the dirid syntax, these characters should not be used in any user ID.

CMS Commands

The following CMS commands have changed syntax and/or modified functions due to SFS:

- ACCESS accesses minidisks or SFS directories.

  An access of another user's directory is always R/O. This prevents an existing program from assuming that, if it can write to a file in a directory or create a file in a directory, then it can write to any file in the same directory. However, even though the directory is accessed R/O, you can update any file for which you have write authority, and you can create new files if you have write authority on the directory.

  Note: You can update these files and create new files if you are using program functions or the XEDIT and COPYFILE commands.

  A directory with no files in it can be accessed R/W or R/O.

  You can use QUERY ACCESSED to display information about the accessed minidisks and directories.

- COPYFILE copies files within minidisks, within directories, or between minidisks and directories.

  You can use COPYFILE to create a new file in any directory for which you have write authority, even if that directory is accessed R/O.

  You can use COPYFILE (REPLACE) to update an existing file on which you have write authority, even if the directory is accessed R/O. Existing aliases of the file and authorities granted on the file are retained. The contents of the base file are updated even when you reference the file through an alias.
- DEFAULTS supports DIRLIST options and new options on FILELIST.
- DISCARD erases base files, aliases, and directories from the full-screen DIRLIST and FILELIST environments.
- ERASE erases minidisk files, SFS files, and directories.
- FILELIST lists the contents of minidisks and directories.
- LISTFILE lists the contents of minidisks and directories.
- QUERY DISK displays information about minidisks and directories.
- QUERY SEARCH displays information about minidisks and directories. If the listed item is a directory, a new fifth column displays the directory's complete dirname. Existing execs that do not use "." to capture unwanted parameters at the end may not work.
- RELEASE releases accessed minidisks and directories.
- RENAME renames a file on a minidisk or renames a base file, an alias, or a subdirectory in a directory.
- STATE/ESTATE returns "file not found" for another user's file on which you do not have authority. STATE/ESTATE also does not find erased aliases (aliases of a file that has been erased), revoked aliases (aliases of a file for which your authority has been revoked), or subdirectories.
- STATEW/ESTATEW returns "file not found" for a file in a directory that is accessed R/O. The file is found only if the directory is accessed R/W and if you have write authority on the file.
- ZAP stops processing if it encounters an open file. All target files must be closed before ZAP is invoked or called.
- ZAPTEXT stops processing if it encounters an open file. All target files must be closed before ZAPTEXT is invoked or called.

For more information about the changed CMS commands, see the *VM/SP CMS Command Reference*. (ZAP and ZAPTEXT are described in the *VM/SP HPO Service Guide*.)

The following CMS commands have no function in SFS and produce an error message if issued against SFS files:

- FORMAT
- RESERVE
- GENDIRT

XEDIT

- You can use XEDIT to create a new file in any directory for which you have write authority, even if the directory is accessed R/O.
- You can use XEDIT to replace an existing file if you have write authority on the file, even if the directory is accessed R/O. The contents of the base file are updated even when you reference the file through an alias.
- If you use the CANCEL macro to terminate an XEDIT session that was entered from the FILELIST environment, you exit FILELIST regardless of how many levels deep you are into the directory structure.

Program Functions

Using Existing Programs on Shared File System (SFS) Files

Existing programs that use FS macros, OS simulation, DOS simulation, or EXECIO continue to work correctly on non-shared CMS files residing in an SFS file pool. This support is limited to files referenced by filename filetype filemode, and minidisks (or directories) referenced by filemode.

Existing programs can manipulate non-shared SFS files without modification. Your programs will work the same as they did before if:

- The files they use are not concurrently updated by other users.
- There are no aliases associated with the files.
- The files are not locked.
- There is no dependence on minidisk addresses, such as 191.

Note: Even when files residing in an SFS file pool are not being shared, any user with administrator authority can change any of the files, which could cause an application to fail.

Existing programs that use FS macros, OS simulation, DOS simulation, or EXECIO also work in certain file sharing situations, such as:

- Reading shared files that are not locked or being updated by other users.
- Updating (but not replacing) shared files that meet ALL of the following criteria:
  - The file is not concurrently locked or open for update by other users.
  - The file is being referenced through a directory accessed R/W (only possible if the base file or alias is in your own directory).
  - You have write authority on the file (always true if the base file is in your own directory).

In other file sharing situations, existing programs might act differently and might require changes to work correctly. For example:

- When a program is reading shared files and directories, another user could have a lock on a file that prevents the program from reading a file that it knows exists.
- If a program can write to a file in a filemode (directory), the program assumes that it can write to any other file in that filemode. However, this is not true if you have an alias in your directory of another user's file on which you have only read authority. The result is an error condition.

- CMS programs that replace an existing file by erasing it (FSERASE macro or ERASE command) then rewriting it require modifications to work correctly when the file is shared:
  - If the file is referenced by an alias, the alias is erased and the base file remains intact. When the file is rewritten, it becomes a new base file and not an alias of the original file.
  - If the file is referenced by its base file name, all aliases and authorities on the file are dropped when the file is erased. The file is successfully rewritten, but you must reestablish all the dropped aliases and authorities.

  Note: If the file is replaced using FS macros, OS simulation, or by a CMS command such as COPYFILE (REPLACE), the aliases and authorities are retained.

Existing programs that call low level file system routines or reference file system control blocks require code changes to work with CMS files in an SFS file pool. In certain cases, existing programs that reference internal system control blocks may require reassembly, code changes, or both to work, even for CMS minidisk files.

For more information about using existing programs with SFS files, refer to the *VM/SP Application Development Guide for CMS.*

Enhanced FS Macros

Enhancements to the FSOPEN, FSCB, and FSCBD macros allow existing programs, with minor modifications, to work with both minidisk files and SFS files:

- The FSOPEN and FSCB macros now include the OPENTYP option. This option is intended for existing programs that do not need the full capabilities of SFS, but need to operate on both minidisk files and SFS files that are being shared.

  With the OPENTYP option, the file is opened immediately. For an SFS file, this places the appropriate locks on the file at the time of open and prevents another user from changing the file characteristics (size, record length, number of records) between the open and the first read or write. Without the OPENTYP option, the FSOPEN macro performs a STATE call, which does not actually open or lock the file.

  When replacing a file, specifying OPENTYP=REPLACE eliminates the need to erase the file first before rewriting the file. For an SFS file, FSERASE destroys all authorities and aliases of the file, but OPENTYP=REPLACE preserves them.

- The FSOPEN macro now includes the MSG and NOMSG options to control whether or not CMS issues error messages when an error occurs. These options aid migration from FSSTATE (which, along with the STATE command, provides some control over error messages) to FSOPEN. The MSG option indicates whether or not an error message is to be displayed. The NOMSG option can be used in conjunction with MSG=YES to prevent an error message from being displayed in certain specified situations.

- The FSOPEN and FSCB macros now include the CACHE option. This option lets you indicate whether or not the file system should cache data blocks, which may reduce the number of actual I/O operations performed on a given file.

- The FSCBD macro has been enhanced to include fields pertaining to the FSOPEN macro and FSCB macro enhancements.

These new options apply to both SFS files and minidisk files (the CACHE option applies only to SFS files and EDF minidisk files). FS macros operate only on files and not on directories. The fileid specified in these macros continues to be in the form filename filetype filemode. Write operations (FSWRITE, FSERASE, FSOPEN OPENTYP=WRITE, FSOPEN OPENTYP=NEW, FSOPEN OPENTYP=REPLACE) require that the target filemode be accessed R/W, even if you have write authority on the file.

Applications that need the full capabilities of SFS (such as writing to another user's files, manipulating subdirectories, referencing files without accessing the parent directory) should use the new SFS program functions. These functions are stored as routines in the VMLIB callable services library (CSL).

For more information about the enhanced FS macros, refer to the *VM/SP Application Development Reference for CMS.*

CMS Initialization and Termination

CMS Initialization at System Build

During CMS nucleus build processing, the DMSINQ module issues a new prompt,

```
File system read/write cache buffer size =
```

to let you specify the amount of read-ahead/write-behind buffering to be used for file input and output operations. You can tune system performance by entering a value based on the characteristics of your system, including real storage size, paging rate, number of users, and so on. If you do not specify a value, the default is 12 (K bytes).

If you specify an incorrect value, the following prompt is issued:

```
Buffer size must be 1-28 (K bytes); reenter
File system read/write cache buffer size =
```

Additional considerations:

- DEFNUC MACRO and DMSNGP ASSEMBLE have been updated to include the new prompt.
- The S-disk, Y-disk, HELP disk, and the disk that the CMS nucleus is written to must be CMS minidisks. However, after the build you can move the Y-disk data to a directory.

CMS Initialization at IPL

You can define a primary (default) filepool id as a parameter on your CP Directory IPL statement. The parameter has the format,

FILEPOOL *filepoolid*

where *filepoolid* is the 1-to-8 character name of a file pool in which you have a top directory.

When you IPL:

- If the primary file pool is not available, then no current file pool is set.
- If the primary file pool is available, then the primary file pool is set as the current file pool.

DMSINS continues to issue the ACCESS command with no parameters if no command or a command other than ACCESS is issued at the initial VM READ.

When issued with no parameters, ACCESS accesses as filemode A either your top directory in the current file pool or, if no current file pool is set, your 191 minidisk.

The system profile (SYSPROF EXEC) issues ACCESS with no parameters and also issues the command SET FILEPOOL PRIMARY.

Additional considerations:

- The S-disk, the 195 disk (for the CMS batch machine), and the HELP disk must all be CMS minidisks at IPL.
- The D-disk (192) and the Y-disk (19E) can be overridden by the system profile to be SFS directories.
- The Y-STAT is not created when an SFS directory is used as the Y-disk.

Abend Recovery (DMSABN)

The abend recovery routine has been modified to clean up the SFS environment by:

- Rolling back all uncommitted changes to SFS files and directories.
- Clearing all workunitids other than the CMS reserved workunitid.
- Clearing each namedef that has been defined.

Normal Termination and Subset Mode (DMSINT)

Upon normal end-of-command, all open shared files and directories are closed and all uncommitted changes are committed. If any work units cannot be committed, they are rolled back and error messages are issued. Any command-defined default work units are removed, and all work units obtained during the command are returned to CMS.

At subset entry, precautions are taken to "hide" the existing SFS environment (open files, open directories, active work units, and so on) from subset mode. At subset exit, this "hidden" environment is restored.

AFT Control Block

The Active File Table (AFT) control block has been reformatted. Some existing fields have been deleted, new fields have been added, and the meaning of some fields has been changed. Any program that uses the AFT should be examined to determine the impact of these changes. All programs that use the AFT must be recompiled.

File System Control Block (FSCB)

The File System Control Block (FSCB has been updated and the meaning of some fields has changed. If you have an application that uses the FORM=E parameter on the FSCB, FSOPEN, FSPOINT, FSREAD, FSSTATE, and FSWRITE macros and you alter fields FSCBITNO and FSCBNOIT, note that these fields are reserved. For more information, see the FSCB and FSCBD macros in the *VM/SP Application Development Reference for CMS*.

Message and Return Code Changes

Modification of some existing CMS commands to include SFS function has resulted in message and return code changes. If you have applications that key off message numbers and/or return codes, you should check the *VM/SP CMS Command Reference* for these changes and modify your applications accordingly.

In Release 5, if the RENAME command or the ERASE command was entered with an invalid operand, message DMSSTT062E was issued with a return code of 20.

In Release 6, if the RENAME command or the ERASE command is entered with an invalid operand, message DMSPCL389E is issued with a return code of 24.

Using an External Security Manager

If you use an external security manager (ESM), some additional things to consider are:

- Objects that are placed under an ESM's protection must have their existing authorities recreated under the ESM, either individually or through an ESM-provided migration utility.
- Users must enter ESM commands instead of GRANT/REVOKE AUTHORITY.
- ESM commands must be used to determine authority information for protected files and directories. The QUERY AUTHORITY command indicates only the existence of ESM protection.
- It is possible for a STATE command entered by an unauthorized user to find an object protected by an ESM, although the subsequent open fails.

Extended Architecture Support in CMS

This section describes the changes to CMS that you must consider before you run your programs in a System/370 mode virtual machine. Keep in mind that if your programs use CP facilities (CP commands, DIAGNOSE codes, message responses, and/or data areas), changes to the CP components of the various VM releases can cause incompatibilities in your programs. Also, note that the changes described in this chapter may affect all programs that run under CMS—not just those that run in System/370 mode.

System/370 Mode Compatibility Conversion Checklist

Use the following table as a checklist to see what you might need to change in programs you plan to run in a System/370 mode virtual machine:

Table 11 (Page 1 of 3). System/370 Mode Compatibility Conversion Checklist

CMS Facility	Notes
Four-digit device addresses	Various I/O related commands (such as QUERY DISK and QUERY SEARCH) return 4-digit device addresses. Applications that expect 3-digit device addresses must be modified to handle 4-digit device addresses.

Table 11 (Page 2 of 3). System/370 Mode Compatibility Conversion Checklist

CMS Facility	Notes
QUERY COMMAND	QUERY DISK output has been changed because of the modification to return 4-digit addresses. Applications that check for column dependent data must be modified to check the new column output changes. QUERY SEARCH output has been extended to include a new fifth column. Applications may have to be modified to handle the new fifth column of data. The fifth column displays the complete dirname for the directory.
DEBUG command	CMS does not support the DEBUG command environment. To perform functions that the DEBUG subcommands provided, use the CP PER, DISPLAY, and TRACE commands. Also, note that program and external interrupts no longer call DEBUG.
NUCXMAP output	The information that the NUCXMAP command returns includes 31-bit addresses and information about the nucleus extension addressing mode. Because of this, NUCXMAP may return more than one line of output to describe a nucleus extension. Programs that expect one line of output from the NUCXMAP command must be modified.
NUCXMAP, NUCXLOAD, and NUCXDROP	These commands now restrict the characters that may appear in a nucleus extension or subcommand processor name. The invalid characters are: =, *, (,), and X'FF'.
Data areas	Several CMS internal control blocks have been updated. You may need to modify programs that reference CMS internal data areas.
Reassembling and executing existing programs	If you reassemble existing programs without change, they should provide the same functions as before. Differences may occur when a macro's parameter list has been changed. For example, if a program uses fixed displacements to dynamically modify a parameter list and the parameter list has changed in CMS, the program may get unexpected results. See the *VM/SP Application Migration Guide for CMS* for a list of the macros with changed parameter lists.
SUBCOM and NUCEXT	You must change any program that uses the SUBCOM or NUCEXT functions to specify interrupt masks that enable or disable specific I/O channels. These programs should use the SUBCOM or NUCEXT macros. The SUBCOM and NUCEXT functions only support the three classes of interrupts supported by the ENABLE macro. Programs that are disabled or enabled for all interrupts are not affected.
OS/MVS simulation	CMS OS/MVS simulation supports most of the functions that were previously supported. New support includes those MVS/XA services that replace or extend the OS/MVS services previously supported. For more information, see "OS/MVS Simulation" on page 100.
Overlay structures	CMS does not support overlay structures. If you use the LOAD or LOADMOD commands to load a non-relocatable program into storage occupied by another program, the existing program is deleted (regardless of whether you issue the PRES option of LOAD or LOADMOD).
MAINHIGH and FREELOWE	CMS does not segregate CMS native free storage and OS GETMAIN simulated free storage. The fields MAINHIGH and FREELOWE are no longer used. In the past, MAINHIGH and FREELOWE were boundaries between the two storage management areas. Programs that are dependent upon these internal fields must adjust accordingly. See "Managing Subpools" on page 96 for more information.

Table 11 (Page 3 of 3). System/370 Mode Compatibility Conversion Checklist

CMS Facility	Notes
GETMAIN free storage	By default, CMS releases GETMAIN storage at SVC 202/CMSCALL termination and it treats the STRINIT macro as a no-op. In prior releases of VM/SP HPO, CMS released GETMAIN storage at command end and respected the STRINIT macro. This change should not affect a program unless it depends on the ability to invoke another program via an SVC 202 or CMSCALL and have the called program obtain *and* return free storage obtained via GETMAIN. If necessary, you can use the SET STORECLR command to retain GETMAIN storage until end-of-command and "enable" STRINIT.

Connectivity

Because connections to private resources are routed via user ID, it is particularly important that no two users in the TSAF collection have the same user ID. Since a private resource connection is routed to the first user ID matching the specified user ID, if there are multiple virtual machines with the same user ID, there is no guarantee that the connection is made to a particular virtual machine. Also, SFS file space is identified by userid. Security problems can occur if there are multiple virtual machines with the same user ID within a filepool of a TSAF collection.

APAR VM30047 (for TSAF) and APAR VM30333 (for CP) are migration APARs that let VM/SP HPO Release 5 and VM/SP HPO Release 6 TSAF virtual machines co-exist in the same collection until all TSAF virtual machines in the collection are at the VM/SP HPO Release 6 level. Not all VM/SP HPO Release 6 functions are available on the VM/SP HPO Release 5 TSAF virtual machine. However, a VM/SP HPO Release 5 TSAF virtual machine with this migration APAR can run on VM/SP HPO Release 5 CMS.

The following is a list of new functions for VM/SP HPO Release 6 that are not available on the VM/SP HPO Release 5 TSAF virtual machine:

- Notification from CP of ownership of gateways,
- The TSAF command QUERY GATEWAY,
- Routing of connections to private servers,
- Connections to applications outside the collection through gateways,
- Requests from CP for USERID/PASSWORD verification,
- Sending and receiving LOGDATA,
- The TYPE, LOGDATA, and FMH5 parameters of APPC/VM CONNECT,
- The LOGDATA parameter of APPC/VM ACCEPT,
- The TYPE=PROG/SVC parameter of APPC/VM SENDERR,
- The TYPE=PREPRECV parameter of APPC/VM SENDCNF, and
- The sever codes for DEALLOCATE_ABEND_SVC and DEALLOCATE_ABEND_TIMER.

Full VM/SP HPO Release 6 TSAF functions are available on the VM/SP HPO Release 6 systems in the collection.

Deletion of SPGEN EXEC and SPGEN PROFILE

The SPGEN EXEC and SPGEN PROFILE are no longer supported. The ITASK EXEC now calls the VMFASM EXEC to do assemblies and the new VMFBLD EXEC to do builds. The DTYPE function of SPGEN EXEC has been added to the UTILITY EXEC. System generation parameters formerly contained in the SPGEN PROFILE are now contained in the new product parameter file.

Starter System Support

In addition to the existing 3350, 3375, and 3380 Starter System version, the following versions are available.

- 3380 Model E4 and
- 3380 Model K4.

DASD Requirements

To install the VM/SP HPO base, you need the following DASD volumes:

VMSRES	system residence volume
VMPK01	additional system volume
VMPK02	additional system volume (required only for 9313 DASD)
VMPK04	source code volume (required only if loading source).

VS COBOL II Programming Language

If you are migrating from CMS 5.0 to CMS 5.5 or CMS 6, you must completely reinstall VS COBOL II.

CPTRAP

The structure of the CPTRAP command has been completely changed. As a result, you have to learn the new syntax for doing the same functions that were available on previous releases. You have to change EXECs and user programs that relied on the previous CPTRAP syntax. In addition, you can now use RACF to control who can use the CPTRAP command.

The QUERY CPTRAP command responses are enhanced and expanded.

Any programs that depend on trace table entry formats have to be updated to handle the new trace entries.

VM/SP HPO Release 6 adds trace table entries of the type X'3C'. If you have programs that create X'3C' entries, update them so they do not conflict with CP's entries. Release 6 also changes X'3D', X'3E', and X'3F' trace entries. In addition, X'20' time of day entries are now located in the trace table and in the CPTRAP file. Update programs that format these trace entries.

With Release 6, the use of CPTRAP *typenums* is changed.

The CPTRAP reduction routine, TRAPRED, has been replaced by IPCS commands. The name of the DUMPSCAN command has been changed to IPCSSCAN and the PRTDUMP command has been changed to IPCSPRT. Minor changes have also been made to some related functions.

Refer to the *VM/SP HPO Interactive Problem Control System (IPCS) Guide* and the *VM/SP HPO Diagnosis Guide* for more information.

Saved Segment Management

You should change all of your DIAGNOSE code X'64' instructions into calls to the SEGMENT macro to make sure you do not have segment overlaps.

Command Changes

Interactive Problem Control System (IPCS) Enhancements

Customers migrating to VM/SP HPO 6 will be able to fully use all IPCS functions with this support. All migration/coexistence factors that exist for VM/SP HPO Release 5, also exist for VM/SP HPO Release 6. The CP TRAPRED command is no longer supported. DUMPSCAN has been replaced by IPCSSCAN. PRTDUMP has been replaced by IPCSPRT. Update products or execs that invoke these commands.

HELP Changes

Update products or execs that accessed the DUMPSCAN HELP menu by direct invocation of the HELP command to invoke the appropriate submenu (HELP CPTRAP MENU or HELP DUMP MENU).

CPTRAP files

IPCS can process CPTRAP files created on previous releases of VM/SP HPO, but time stamp functions are not supported.

The SVC Trace Entry has two CP trace table entry formats. IPCS processing will be changed to provide support for the HPO unique format for the IPCS SELECT command.

3380 Models AJ4/BJ4 and AK4/BK4

3380 Migration Aid

The 3380 Migration Aid (MIG3380) is a new tool that you can use to adjust the allocation map after you migrate data between different 3380 models.

A 3380 DASD volume is formatted with a 1K, 2K, or 4K allocation map according to the number of cylinders contained in the volume, which varies from model to model. When you back up a 3380 volume and restore the data to another 3380 volume that has a different number of cylinders, the original allocation map is carried over, and does not match the new volume. The MIG3380 program rearranges the contents of cylinder 0 on the new volume to correct the size of the allocation map and filler records, while preserving CP-specific information.

Intervention-Required Error Recovery

For the 3375 and 3380 DASD, when handling an intervention-required situation, the error recovery procedure (ERP) sends a message to the operator but does not wait for device end signal to arrive. System operation continues because the interrupt pending flag is not set to on.

To migrate the contents of an existing 3380 CP volumes to a 3380 AJ4/BJ4 or AK4/BK4 volume, do the following steps *after* you complete the service application procedure.

1. Use the updated FORMAT/ALLOCATE program to CP format the new 3380 volume (if not done already).
2. Use the DDR program to COPY or RESTORE the existing 3380 volume contents to the new volume.
3. Use the 3380 Migration Aid (MIG3380) to adjust the allocation map on the new volume. Enter the following command:

   ```
   mig3380 ccuu
   ```

   where ccuu is the address of the new volume.
4. If the new 3380 volume has more cylinders than the old 3380 volume, use the ALLOCATE function of the FORMAT/ALLOCATE program to specify allocation values for the additional cylinders, if you want allocations other than PERM.

You can use this procedure to migrate the contents of any 3380 CP volume to another 3380 volume that has a different number of cylinders.

3990 Storage Control and 3380 Model CJ2

When the 3990 storage control or 3380 Direct Access Storage Direct Channel Attach Model CJ2 enhancement is installed you can request or decline write-verification (read-back checking). When you format count key data (CKD) DASD with the CP Format/Allocate stand-alone utility (DMKFMT), you are prompted after entering the DASD label with the following:

WRITE VERIFICATION NOT PERFORMED UNLESS REQUESTED.
ENTER "YES" FOR WRITE VERIFICATION:

If you enter all or part of the word yes ("y", "ye" or "yes") in upper or lower case, write verification is performed during formatting. If you accept the default, write verification is *not* done, and formatting time is reduced. Explicit write verification is usually not necessary for direct access storage. We do, however, recommend write verification for newly installed DASD or after DASD maintenance has been done.

Note: For the 2305, 2311 or 2314 DASD write verification is always performed.

DASD Dump Restore Streaming Support Improvements

Consider the following when using the enhanced DASD Dump/Restore support:

- DDR backup tapes produced in the old tape format can be used as input to the enhanced DDR without any change to user action.
- DDR backup tapes produced using the new format *cannot* be used as input to prior levels of DDR.
- You can specify the OLDFORM option on the DDR DUMP statement to produce tapes in the old format. You can use this format if you want to produce a backup tape that could be restored using an older version of DDR, or if you do not want the support when you are using non-streaming tape devices.
- The tape-to-tape COPY function produces tape output in the same format (new or old) as the input tape. An older version of DDR cannot be used to copy a tape produced with the new format.
- If you are running DDR in a virtual machine and you want to use the larger internal buffers, you must have at least 3M of virtual storage defined. Otherwise, DDR defaults to running the older version.

Support for the 3990 Storage Control Model 3

- Operational changes are *not* required after applying this support, *except* when 3990 Model 3 Storage Controls are installed. The new CP commands and directory control statements may be used with existing caching storage controls (3880 Models 11, 13, 21, and 23), but are *not* required.
- The syntax and function of the existing CACHE command remain unchanged. The CACHE command cannot be used to control caching for the 3990 Model 3; the SET/QUERY CACHE command *must* be used. SET/QUERY CACHE supports all existing cache storage controls and is recommended instead of the CACHE command. The SET/QUERY CACHE command is also recommended to help future conversion to Virtual Machine/Extended Architecture System Product (VM/XA SP) because the syntax and function of the command in VM/SP HPO and VM/XA SP are compatible.
- Before this support, minidisks on 3380 DASD attached to the 3880 Model 13 or 23 did *not* use the cache unless you specified 3380C as the device type on the MDISK statement. With this support, minidisks with 3380C specified as the device type *or* with MINIOPT CACHE directory statements will *use* the cache. The default still remains *not* to use the cache for 3380 DASD attached to the 3880 Model 13 or 23.

  The default is to *use* the cache for minidisks on 3380 DASD attached to a 3990 Model 3 Storage Control. The cache is used regardless of whether the device type on the MDISK statement is 3380 or 3380C. The only way to bypass the cache of a 3990 Model 3 is to specify MINIOPT NOCACHE. VM/XA SP uses the MINIOPT statement as the method for specifying minidisk caching.

- CP does not prevent or detect activation/deactivation of caching by channel programs issued on other channel interfaces. Installation of VM/SP HPO initialization procedures via the AUTOLOG1 user ID's PROFILE EXEC should ensure that caching is properly activated or deactivated by using the SET CACHE command. For example, paging and swapping volumes are not recommended as good candidates for caching so you should issue SET CACHE DEVICE OFF for them.
- The control level mechanism (SYSCTL, DEVCTL, NOCTL) for control of CCW execution on the 3990 is also implemented for the 3880 Models 11, 13, 21, and 23. It should be used, instead of the CACHE OWN command, to control the use of CCW's affecting device or subsystem level resources or functions. This method is also used by VM/XA SP.

National Language Support File Naming Conventions Enhancements

The only necessary migration path is to install the NLS file naming conventions enhancements from the Program Update Tape (PUT). Once this support is on your system(s), any new VM language object files are created with the traditional VM naming conventions. Any previous VM language files continue to be supported as described below.

Coexistence

VM object files created before installation of the NLS file naming conventions enhancements continue to be supported. This is accomplished by first searching the new convention of TXT*ptfno,* TXT*lvlid*, or TEXT object file names. If none of these can be located, the affected commands use the previous TXT*langid* object file name.

Security and Integrity Enhancements

In the past, calls from CP to ACI were unconditional; with this enhancement they are conditional. So, ensure that all previous unconditional calls remain unconditional by turning on their PROTECT bits in the ACI bit map. Normally, they are on, unless you turn them off.

Support for Mixed Directory

Migration to and from HPO and XA systems is eased with this support. With the addition of this support, users may use the same directory on VM/SP HPO 4.2, 5.0, VM/XA SP1, and VM/XA SP2 systems. Applications that issue the DIRECT command from CMS will have to be altered to issue the MIXED option if applicable.

Support for VM/Directory Maintenance Licensed Program (DIRMAINT)

With this support, DIRMAINT migrates the entire CP user source file (or source directory) from its current format to cluster format. In cluster format, user definitions reside in cluster files. Each cluster file may contain several user definitions. The number of users in one cluster file depends on the size of the individual user definitions.

Free Storage Enhancements

The free storage assists provided by ECPS Levels 18 and 19 are disabled by this support.

The default for free storage management error recovery is ON. System programmers do not have to take any direct action to benefit from this function. This function can be deactivated by issuing the command SET FREERECV OFF.

The threshold level for SWPTABLE migration remains at the same percentage of initial free storage occupied by SWPTABLEs (13 percent).

Any local modifications or programs that reference the ALOCBLOK, class 0 code 7 monitor record, FRELS list, SSBLOK or subpools should be examined for conflict with this support.

Chapter 13. Performance

This chapter describes the performance impact of enhancements in VM/SP HPO Release 6.

Free Storage Enhancements

Free storage error recovery detects and recovers from corrupted free storage allocation chains and certain incorrect calls to deallocate free storage blocks. With this support, you can also specify when swap table migration occurs.

Free storage errors may still result in lost storage in the following cases:

- When an invalid block is returned to free storage
- When a subpool is corrupted
- When the free storage chain is corrupted.

Depending on the amount of storage that is lost, performance may be degraded due to increased storage constraint below the 16MB line.

VM/Directory Maintenance Licensed Program (DIRMAINT) Release 4

This support improves performance:

- Since DIRMAINT does not need to process the entire source directory each time it makes an update
- By reducing I/O contention to the minidisk that contains the real object directory.

DASD Dump Restore Streaming Support Improvements

Benefits

This enhancement:

- Reduces the time required to transfer data from an FBA DASD to a streaming tape device, and vice versa
- Utilizes less space on the system minidisk.

IBM 3380 Direct Access Storage Models AJ4/BJ4 and AK4/BK4

VM/SP HPO supports the IBM 3380 Direct Access Storage Models AJ4/BJ4 and AK4/BK4. These enhanced subsystem models offer improved data availability and error recovery compared to earlier 3380 models. The support for the 3380 AJ4/BJ4 and AK4/BK4 is functionally equivalent to that for currently supported 3380 units, with additional support to recognize the increased number of cylinders in the 3380 AK4/BK4.

Terminal Usability Enhancements

There are several new enhancements to VM/SP HPO and VM/VTAM support. You will notice the increased similarity between VM/VTAM and native terminals.

These enhancements include:

- The PFnn RETRIEVE function of the CP SET command operates both forward and backward.
- When in full-screen status, you can have line-mode WRITEs wait in a queue rather than let them appear immediately at your VM/VTAM terminal.
- CP performs data length checking for all types of devices, including VM/VTAM terminals.
- Full-screen error handling on a VM/VTAM terminal is more compatible with native terminals.
- The length of data processed from a console mode read in a VM/VTAM environment can be the same as in a native environment.

Security and Integrity

This support is designed so that a system that does not have the auditing features enabled will not experience a change in overall performance.

The performance of a system using security features will be impacted in proportion to the number of security auditing functions that are turned on.

VSE

As a preferred guest, VSE operates in supervisor state with direct control of its hardware resources and I/O operations. VSE can also use the storage above 16MB, providing additional performance potential for a storage constrained system.

Checkpoint Operations

By reducing the number of checkpoint operations, the number of operations that the checkpoints cylinders must handle is reduced. This support reduces possible system bottlenecks which would cause decreased response time for the users.

Improved Dispatcher Request Queue Management

By making the scheduler run without the system lock, this support enables both processors to process their queues more frequently, thus reducing wait time on queues and storage utilization.

Support for the IBM 3990 Storage Control Model 3

The 3990 Model 3 provides the following performance enhancement:

- Improved data transfer rate between the cache and the channel.
- Improved internal cache algorithms resulting in greater efficiency.

Part 4. Internal Design Changes

This part of the manual lists changes to the internal design of VM/SP HPO that affect performance, usability, or maintenance.

It includes:

- A list of new and changed modules, macros, and control blocks for each addition or enhancement
- A list of general changes for CP, CMS XEDIT, and IPCS. It also includes a list of module splits.
- A list of Object Code Only (OCO) modules

Chapter 14. Design Changes by Function

This section applies to CP, CMS, XEDIT and IPCS. Information about GCS and TSAF is optional. Information about AVS is not included because the source code is not distributed.

This section lists new and changed CP, CMS, XEDIT and IPCS modules, control blocks, macros, and so on, for the new and changed functions. It can help you with planning. The functions are listed alphabetically.

Callable Services Library

New and Changed CMS and XEDIT Modules

DMSCAX DMSCCR DMSCFL DMSCFX DMSCGT
DMSCLL DMSCRI DMSCRL DMSCRM DMSCRR
DMSCRS DMSCSM DMSRCS

Note: Individual CSL routines are text files located in the VMLIB CSLSEG.

New and Changed CMS and XEDIT Control Blocks and Macros

CSLENTRY CSLEXIT CSLFPI CSLFRWA
CSLFRWB CSLGETP CSLGTWA CSLGTWB
PROFCLST PROFTMPL X$CLST$X X$TMPL$X

New and Changed CMS and XEDIT EXECs

CSLGEN CSLLIST DATADEF

New Miscellaneous CMS File

DMSCSL
TEXT

Connectivity

New and Changed CP Modules

DMKALG DMKCPJ DMKCRM DMKDIR DMKIDR
DMKIUA DMKIUB DMKIUC DMKIUE DMKIUG
DMKIUJ DMKIUL DMKIUN DMKIUP DMKIUS
DMKIUT DMKIUU DMKIUW DMKJRL DMKLOG
DMKLOH DMKLOJ DMKSCH DMKTOD DMKUDR

New and Changed CMS and XEDIT Modules

DMSABN DMSAXR DMSCDB DMSCDN DMSCRD
DMSINT DMSITE DMSIUC DMSIUH DMSMES
DMSNAM DMSNUC DMSPSV DMSQRC DMSQRT
DMSQRY DMSSEC DMSSEF DMSSET DMSSPA

New and Changed CP Control Blocks and Macros

ACIPARMS APPCVM CONEXT DCRBLOK
IPARML IUCV IUCVBLOK LSRBLOK
SRTBLOK SSCBLOK UDIRECT VMBLOK

Changed CMS EXEC

SYSPROF

New and Changed CMS and XEDIT Control Blocks and Macros

CMSIUCV DMSPSIB HNDIUCV IUCVIDBK IUCVTAB

New Miscellaneous CMS File

DMSSAA
TEXT

CP GIVE Command

New and Changed CP Modules

DMKGIV

New and Changed CP Control Blocks and Macros

GIVEBLOK RDEVBLOK VDEVBLOK

CPTRAP

New and Changed CP Modules

DMKCFC DMKCPP DMKCPU DMKCQC DMKCQD
DMKDSP DMKDTR DMKDTS DMKFDA DMKFDL
DMKFIO DMKGTR DMKIOT DMKMES DMKSVC
DMKTCW DMKTCX DMKTRF DMKTRO DMKTRP
DMKTRT DMKTRU DMKTRX DMKTTD DMKTTR
DMKUSO DMKVMG

New and Changed CP Control Blocks and Macros

DLSRC EVTBLOK INTBLK NAMENTRY
PSA RDEVBLOK SFBLOK TCXBLOK
TRDISPLY TRPBLOK TRTDATA

CP DASD Block I/O System Service

New and Changed CP Modules

DMKBIM DMKBIO

New and Changed CP Control Blocks and Macros

BIMBLOK BIRBLOK MBPLIST VMBLOK

CP Interface for Application Monitor Data

New and Changed CP Modules

DMKCFP DMKDIR DMKHVC DMKHVH DMKLOH
DMKMCC DMKMIA DMKMOO DMKPRG

New and Changed CP Control Blocks and Macros

EQU SYSMON UDIRECT VMBLOK

New and Changed CP Copy Blocks

MONBLOKS

DASD Dump Restore Streaming Support Improvements

New and Changed CP Modules

DMKDDR DMKDDT

Default A-Disk

New and Changed CP Modules

DMKHVE DMKREI

DIAGNOSE Code X'64' Subcode X'0018'

New and Changed CP Modules

DMKCFG

New and Changed CMS Modules

DMSRSF

DIAGNOSE Code X'E4'

New CP Module

DMKHVH

Changed CP Modules

DMKCCD DMKCCF DMKCCS DMKDEF
DMKDIR DMKHVC DMKJRL DMKLNK
DMKLNM DMKLOH DMKPSA DMKSCO
DMKUNT DMKVDR

New CP Control Block

FPOBLOK

Changed CP Control Blocks

PSA RBLOKS UDIRECT VMBLOK VBLOKS

DIAGNOSE Code X'64' Subcode X'0014'

New and Changed CP Modules

DMKCFF DMKCFG DMKHVC

Enhanced Connectivity Facilities

New and Changed CMS and XEDIT Modules

DMSBOF DMSBUS DMSCCM DMSCIA DMSCUT
DMSICT DMSMES DMSPBS DMSSAP DMSSMG
DMSSNC DMSSPA DMSSSY DMSSUP DMSTRC
DMSUSR

New and Changed CMS and XEDIT EXECs

CMSSERV CMSGEND DEFAULTS

Extended Architecture Support in CMS

New and Changed CMS Modules

DMSABN DMSABX DMSACC DMSACF DMSACG
DMSACM DMSACP DMSACS DMSALU DMSAMS
DMSARE DMSARX DMSASM DMSAUD DMSBOP
DMSBRD DMSBTB DMSBWR DMSCAT DMSCCX
DMSCIO DMSCIT DMSCLS DMSCMP DMSCPF
DMSCPY DMSCRD DMSCSF DMSCWR DMSDBD
DMSDBG DMSDDL DMSDFT DMSDID DMSDIE
DMSDIO DMSDLB DMSDLK DMSDOS DMSDSC
DMSDSK DMSDSV DMSEDI DMSEDX DMSEIO

DMSERD DMSERR DMSERS DMSEXE DMSEXG
DMSEXI DMSEXL DMSEXT DMSFET DMSFLD
DMSFLO DMSFNC DMSFNS DMSFOR DMSFRA
DMSFRI DMSFRM DMSFRO DMSFRR DMSFRS
DMSFRU DMSGND DMSGRN DMSGTS DMSHDE
DMSHDI DMSHDS DMSHLB DMSHLL DMSHLT
DMSHLZ DMSHTB DMSICP DMSIFI DMSIMA
DMSIMM DMSIND DMSINI DMSINN DMSINQ
DMSINS DMSINV DMSIOW DMSITE DMSITI
DMSITM DMSITP DMSITS DMSIUV DMSLAD
DMSLBO DMSLBM DMSLBT DMSLDF DMSLDR
DMSLIB DMSLIO DMSLKD DMSLMX DMSLOA
DMSLOS DMSLSB DMSLST DMSMCM DMSMDP
DMSMES DMSMGC DMSMGD DMSMGM DMSMOD
DMSNAM DMSNCP DMSNUC DMSNXD DMSNXL
DMSNXM DMSOVS DMSPCA DMSPCL DMSPCT
DMSPCW DMSPGM DMSPHA DMSPIO DMSPLA
DMSPMD DMSPNT DMSPOA DMSPOL DMSPON
DMSPOP DMSPOQ DMSPOR DMSPOS DMSPRE
DMSPUN DMSQRF DMSQRG DMSQRH DMSQRS
DMSQRT DMSQRY DMSRCN DMSRDC DMSRDR
DMSREX DMSRLD DMSRNM DMSROS DMSRPG
DMSRSF DMSRSV DMSRXE DMSSAB DMSSBS
DMSSCA DMSSCT DMSSEB DMSSEF DMSSEG
DMSSET DMSSEV DMSSFD DMSSFF DMSSLG
DMSSLN DMSSMN DMSSOP DMSSPA DMSSPI
DMSSPR DMSSRS DMSSSK DMSSTA DMSSTG
DMSSTK DMSSTM DMSSTQ DMSSTT DMSSTX
DMSSVN DMSSVQ DMSSVT DMSSYN DMSTIO
DMSTLB DMSTLM DMSTMA DMSTMR DMSTPD
DMSTPE DMSTPF DMSTPG DMSTPH DMSHLB
DMSTPJ DMSTRK DMSTVS DMSTYP DMSUPD
DMSVIS DMSWIN DMSWIO DMSWIR DMSWIS
DMSWLR DMSWLW DMSWRT DMSWST DMSWTE
DMSXCT DMSXIN DMSXMA DMSXMB DMSXPO
DMSXSC DMSXSD DMSXSF DMSXSU DMSZAP

New and Changed CMS Control Blocks and Macros

ABNEXIT ABNXTCB ACCSECT AMODESW APPLMSG
BBOX BGCOM CLOSE CMSCALL CMSCVT
CMSDEV CMSIUCV CMSLEVEL CMSRET CMSSTACK
CMSSTOR COMPSWT CONSOLE CQYSECT DBGSECT
DEFNUC DEVGEN DEVSECT DEVTAB DIOSECT
DMSABN DMSABW DMSCBLK DMSCDEV DMSEDFPL
DMSEPIEB DMSFRENT DMSFREXT DMSFRIPL DMSFRPSE
DMSFRWRK DMSGTWA DMSGTWB DMSGTWE DMSHDRBK
DMSIOCON DMSLICMD DMSLINK DMSLITXA DMSMACXA
DMSMCKSA DMSMSG DMSMSU DMSMSZ DMSORG
DMSPATH DMSPBIT DMSPGEN DMSPHABK DMSPHAST
DMSPID DMSPIECB DMSPLGN DMSPLIN DMSPLTL
DMSPRGBK DMSPRIML DMSPRTRG DMSPSBC DMSPSEC
DMSPSW DMSPSXA DMSPVER DMSPWOC DMSPWPC
DMSPWUQ DMSPWVF DMSREFCB DMSSGIOP DMSSIMWK
DMSSPPL DMSSTACB DMSSTOBK DMSSTPL DMSSUBBK

DMSSYID DMSTCBCB DMSTIMCB DMSTLW DMSUSRPL
DMSUSVC DMSXBASE DMSXCONT DMSXDISC DMSPNTR
ENABLE EQUATES EXTHBLK EXTSECT EXTUAREA
FDEFSECT FSCB FSCBD FSCLOSE FSERASE
FSOPEN FSPOINT FSREAD FSSTATE FSWRITE
FVS GETSID HNDEXT HNDINT HNDIO
HNDIUCV HNDSVC HSVCSAVE IMMBLOK IMMCMD
IMWKSECT INTBLOK IOSECT IUCVIDBK LDRST
LINERD LINEWRT NUCEXT NUCON OPEN
PARSECMD PGMSECT PGSER PRINTL PUNCHC
PVCENTRY QRYWORK RDCARD RDTAPE REXEXT
SCAN SEGMENT SUBCOM SUBPOOL SVCSECT
SYSNAMES TAPADD TAPECTL TAPESL TAPEWORK
TSOBLKS USERSAVE WAITD WAITECB WAITT
WRTAPE WTO WTOR

EXTRACT/REPLACE

New and Changed CMS Modules

DMSERP DMSETP DMSERX DMSEXR

Installation and Service

Changed CP Module

DMKLD00E

New and Changed CMS Execs

GENSERVE GENTSAF HITASK HUTILITY
VMFAPPLY VMFASM VMFBLD VMFHASM
VMFOVER VMFREC VMFSETUP VMFTXT

Deleted CMS Exec

SPGEN

New CMS File

5664173h
$PPF

Deleted CMS File

SPGEN
PROFILE

New and Changed CMS Modules

UPDATE VMFLOAD VMFPLC2

IPCS Enhancements

New and Changed CP Modules

DMKHVC DMKHVG DMKTED DMKTEE DMKTEF
DMKTEH DMKTEM DMKTES DMKSPS DMKSPT

New and Changed IPCS Modules

DMMCOM DMMCPA DMMDCM DMMDCP DMMDID
DMMDIN DMMDIR DMMDSC DMMFEX DMMHEX
DMMINP DMMLOC DMMLUN DMMMAP DMMMCS
DMMMOD DMMPRO DMMPRT DMMPTR DMMSCN
DMMSCR DMMSEL DMMSLD DMMSLT DMMSVT
DMMTAB DMMTBL DMMTFR DMMTIM DMMTPL
DMMTRC DMMTRD DMMTRP DMMTRS DMMTSC
DMMTSL DMMTSP DMMTSU DMMVMF

New and Changed CP Control Blocks and Macros

HEXDEC TED

New and Changed CP Copy Files

CPTRAP SPOOL SPTBLOKS TRACEFMT

New and Changed IPCS Control Blocks and Macros

CCODE MSGP

New and Changed IPCS Copy Files

GSA MACENTRY SLCDATA SVC199
TSC TSU WORKCP

New and Changed IPCS Execs

APAR TRAPFILE

New and Changed Miscellaneous IPCS Files

CUSTOMER PROFILE

National Language Support Enhancement

Changed EXECS

PEEK RECEIVE

Changed HELP Files

RDR HELPCMS

Changed CMS Module

DMSRDR

National Language Support File Naming Conventions Enhancements

Changed CMS Modules

DMSMGC DMSQRS DMSSLD DMSSLG

Changed CP Execs

APLOAD AVLOAD CPLOAD CPLOADSM VRLOAD

Changed CMS Execs

CMSLOAD CONVERT LANGGEN LANGMERG VMFAPTXT
VMFLOAD VMFNLS

Performance Monitor

New and Changed CP Modules

DMKACO DMKDIR DMKIUA DMKIUL DMKIUN
DMKIUS DMKLOH DMKMCC DMKMON DMKMOO
DMKVMC

New and Changed CP Control Blocks and Macros

UDIRECT UMACBLOK VMBLOK

New and Changed CP Copy Files

MONBLOKS COPY

QUERY DISK

New and Changed CMS Modules

DMSPRI DMSQRS DMSSPA

READCARD and DISK LOAD Command Options

New and Changed CMS and XEDIT Modules

DMSGLO DMSNAD

New and Changed CMS Control Blocks and Macro

GLVBLOCK

Saved Segment Management

New and Changed CMS Modules

DMSABN DMSDCS DMSDCT DMSEXD DMSEXL
DMSEXM DMSEXQ DMSGEN DMSGEO DMSGEP
DMSINS DMSITS DMSLDR DMSLOA DMSLSB
DMSNXD DMSNXM DMSRLD DMSSEG DMSSFA
DMSSLG DMSSFD DMSTBA DMSTBB DMSTBC
DMSTBD DMSTBE DMSTBG DMSTBS

New and Changed CMS Control Blocks and Macros

EXISBLK NUCEXT SCBLOCK SGMTEXIT
COPY

SET AUTODUMP and QUERY AUTODUMP

New and Changed CMS and XEDIT Modules

DMSABN DMSDIE DMSMES DMSQRT DMSQRY
DMSSET DMSSPA

New and Changed CMS Control Blocks and Macros

ABWSECT

New and Changed IPCS Modules

DMMDCC

Shared File System

New and Changed CMS and XEDIT Modules

DMSACC DMSACP DMSACR DMSALU DMSAMS
DMSARE DMSARN DMSARX DMSASM DMSASN
DMSAUD DMSBRD DMSBTB DMSBWR DMSCAT
DMSCMP DMSCPY DMSCRD DMSDEL DMSDDL
DMSDLK DMSDSK DMSDSL DMSDSV DMSEIO
DMSERA DMSERC DMSEXG DMSEXI DMSEXL
DMSFNC DMSGLB DMSGLO DMSGND DMSGRN
DMSGVE DMSHLB DMSHLC DMSHLI DMSHLL
DMSHLM DMSHLP DMSHLS DMSHLZ DMSHTB
DMSICP DMSIMA DMSINS DMSINX DMSJDE
DMSJLK DMSJND DMSJNL DMSJSE DMSJVS
DMSLAD DMSLAF DMSLBM DMSLBT DMSLDR
DMSLFS DMSLGT DMSLIB DMSLIO DMSLKD
DMSLLU DMSLOA DMSLSB DMSLSS DMSLST
DMSMES DMSMGC DMSMOD DMSNAM DMSNGP
DMSNUC DMSNXL DMSOSC DMSOUT DMSOVR
DMSPCL DMSPMD DMSPNT DMSPRE DMSPRT
DMSPRV DMSPUN DMSPVF DMSQRE DMSQRG
DMSQRH DMSQRN DMSQRO DMSQRP DMSQRQ

DMSQRR DMSQRS DMSQRY DMSRDC DMSRLD
DMSRND DMSRNM DMSRRV DMSRSV DMSSET
DMSSLG DMSSOP DMSSPA DMSSPR DMSSRT
DMSSTT DMSSVQ DMSSVR DMSSYN DMSTMA
DMSTPD DMSTPE DMSTPF DMSTPG DMSTPH
DMSTPI DMSTPJ DMSTRT DMSTYP DMSUPD
DMSUTL DMSWLR DMSWVL DMSWVT DMSXBG
DMSXCP DMSXCT DMSXED DMSXFI DMSXFW
DMSXGT DMSXIN DMSXMB DMSXMD DMSXPO
DMSXPT DMSXQR DMSXSC DMSXSE DMSXSU
DMSXTR DMSXUP DMSZAP DMS2AB VMFDATE
VMFLOAD VMFPLC2

CMS Module Splits

DMSERS	DMSERC	DMSERS
DMSINT	DMSINT	DMSINX
DMSSOP	DMSOSC	DMSSOP

New and Changed CMS and XEDIT EXECs

ALIALIST AUTHLIST CANCEL CONVERT DEFAULTS
DIRLIST DISCARD DOSPLI EXPAND ESERV
EXECUPDT EXECUTE FILELIST FCOBOL ITASK
LANGMERG LANGGEN PROFALIA PROFAUTH PROFDLST
PROFFLST PROFFSEA PROFFSHR SDIR SMODE
SPLOAD STATUS SYSPROF VMFAPPLY VMFASM
VMFBLD VMFHASM VMFLKED VMFOVER VMFREC
VMFSETUP VMFTXT VSAMGEN VSEVSAM X$DLST$X
X$FLST$X X$NDIR$X $EXIST$

New and Changed CMS Control Blocks and Macros

ACCSECT ADT ADTSECT ADTP AFT
CMSSTORP CONTROL DCH DCHP DEFNUC
DIRBUFF DIRBUFFP DMSABW DMSABWP DMSAFT
DMSAFTP DMSDNT DMSERSWK DMSEXPID DMSEXPIP
DMSFINAT DSMFLPWK DMSFNSDT DMSFNT DMSFQDN
DMSFQDNP DMSGTWA DMSGTWB DMSGTWE DMSGTWL
DMSGUSA DMSGUSAP DMSHELP DMSINCCW DMSINTWK
DMSJNEPL DMSJNEPP DMSJRSC DMSJRSCP DMSJWUM
DMSJWUMP DMSKXCHP DMSLSTPL DMSLSTWP DMSNADDR
DMSNADGN DMSNDEN DMSNGMAP DMSODT DMSOUTPL
DMSOUTPP DMSPDCB DMSPDCBP DMSSDEFP DMSSDERR
DMSSDMGS DMSSDMSP DMSSFSCA DMSSFSCB DMSSFSCP
DMSSOPWK DMSSPSI DMSSYID DMSSUIDP DMSUCT
DMSVAFT DMSXADT EXSBUFF EXSBUFFP FPERROP
FPERROR FSCB FSCBD FSCBP FSCBSET
FSCLOSE FSCLOSEP FSOPEN FSOPENP FSREAD
FSREADP FSWRITE FSWRITEP FSTB FSTBP
FSTD FVS FVSP GETTOKEN HASHTAB
HASHTABP HLPSECT KXCHK LDRST NUCON
PVCENTRP PVCENTRY QRYWORK QRYWORKP REGEQU
SUBSECT SUBSECTP TAPEWORK TRANTBL TRANTBP
USERSAVP WUERROP WUERROR ZBLOCKS

SPOOL Enhancement Accommodation

New and Changed CP Modules

DMKAPS DMKAPT DMKAPU DMKAPV DMKAPW
DMKAPZ DMKCKH DMKCSW DMKCSU DMKCQH

New and Changed CP Control Blocks and Macros

LBLOKS

System Product Interpreter

New and Changed CMS and XEDIT Modules

DMSREV DMSRXE DMSSTR

Terminal Usability Enhancements

New and Changed CP Modules

DMKCFT DMKCFY DMKCQY DMKGRG DMKLOH
DMKRET DMKRGC DMKTTX DMKVCN DMKVCR
DMKVCS DMKVCQ DMKVCV DMKVCU

New and Changed CP Control Blocks and Macros

PFKTABLE RETBUF SNARBLOK
WEBLOK WEIBLOK

3380 DASD Models AJ4/BJ4 and AK4/BK4

New and Changed CP Modules

DMKALO DMKCCD DMKCCH DMKCPI DMKCPW
DMKCPZ DMKDAD DMKDDR DMKDIR DMKDMP
DMKFMT DMKHVD DMKHVE DMKIDU DMKIOC
DMKIOE DMKIOJ DMKLNK DMKMAP DMKMIG
DMKMNI DMKMNT DMKMOO DMKPAG DMKPMA
DMKSAD DMKTOD DMKUDR DMKVDE DMKVDG
DMKVER DMKVSC

New and Changed CMS Modules

DMSFOR

New and Changed CP Copy Files

CCHREC MDRREC OBRRECN
PPMAP RBLOKS TNSREC

New and Changed Sample Files

Note: The sample system files assume that all system DASD volumes are on the same device type and model as VMSRES. If you are using a different type or model for any volume, you may have to make some adjustments to these files.

DIRECT 3380-K4 DMKSNT 3380-K4

Changed Installation Tools

$DASD$ CONSTS ITASK EXEC DIRECT 3380-3

DMKSNT 3380-3

3990 Storage Control Models 1 and 2 and 3380 Direct Access Storage Direct Channel Attach Model CJ2

New and Changed CP Modules

DMKCAC DMKCAO DMKCCD DMKCCO DMKCCS

DMKCCW DMKCFK DMKCMD DMKCPJ DMKCPN

DMKCPW DMKCQF DMKCQG DMKCQP DMKCQQ

DMKCQV DMKDAD DMKDDR DMKDEF DMKDIR

DMKDSB DMKFMT DMKIOC DMKIOE DMKIOF

DMKIOJ DMKIOS DMKLNK DMKLOJ DMKMES

DMKMNL DMKMNT DMKTRA DMKTRC DMKTRD

DMKUNT DMKVDA DMKVDB DMKVDC DMKVDE

DMKVDS DMKVDT DMKVER DMKVES DMKVIO

DMKVSC DMLCQK

New and Changed CP Control Blocks and Macros

ALRTREC RCTLUNIT VESPARMS

New and Changed CP Copy Files

CCWSAVE IOBLOKS IOER

RBLOKS RCUBLOK TREXT

Support for Mixed Directory

New and Changed CP Modules

DMKDIR

Support for the IBM 3990 Storage Control Model 3

New and Changed CP Modules

DMKCAC DMKCAO DMKCCD DMKCCO DMKCCS
DMKCCW DMKCFK DMKCMD DMKCPJ DMKCPN
DMKCPW DMKCQF DMKCQG DMKCQP DMKCQQ
DMKCQV DMKDAD DMKDEF DMKDIR DMKIOS
DMKLNK DMKLOJ DMKMNL DMKMNT DMKVDA
DMKVDB DMKVDC DMKVDE DMKVDS DMKVDT
DMKVES DMKVSC DMLCQK

Changed CP Control Blocks

CACHBLOK CCWSAVE EQU MONBLOKS UDIRECT
VBLOKS

Free Storage Changes

New and Changed CP Control Blocks

ALOCBLOK CPEXBLOK FEVENTRY FRECOM FRELBLOK
FSMPLIST MONBLOKS MN007 SSBLOK

New and Changed CP Modules

DMKCFO DMKCKR DMKCMD DMKCPI DMKCPP
DMKCQR DMKCSZ DMKFEV DMKFRE DMKFRT
DMKFSM DMKHVD DMKMES DMKMOO DMKPAH
DMKPGS DMKPGT DMKPGU DMKPTR DMKPTS
DMKPTT DMKRSP DMKSEL DMKSRM DMKSTD
DMKSTP DMKSTR DMKSWA DMKTMR DMKTOD
DMKUSO DMKUSP

New and Changed CP Macro and Copy Files

ALLOC CALL CORE FREBLOKS
FRECOM MONBLOKS PPMAP PSA
PXA

Paging Storage and Vector Facility Support

Changed CP Modules

DMKCQS DMKMES DMKMESA DMKMESB
DMKMESD DMKMESE DMKMNI DMKPST
DMKVFD DMKVFE DMKXST

VM/Directory Maintenance Licensed Program (DIRMAINT)

Changed CP Modules

DMKDIR DMKHVD DMKUDU DMKUDV

Alternate VSAM

Changed CMS Modules

DMSABN DMSFLD DMSFLE DMKFLO DMSMES
DMSOSC DMSQRU DMSSEV DMSSET DMSSOP

Changed CMS Control Blocks

FCBSECT FDEFSECT NUCON QDRYWORK TABENT

Security and Integrity Enhancements

Changed CP Modules

DMKCFC DMKCFB DMKCKS DMKCMD DMKCQS
DMKCSP DMKCST DMKCSX DMKCSY DMKDRD
DMKDRE DMKHPS DMKHVC DMKHVF DMKHVH
DMKHVI DMKIUC DMKIUJ DMKLNK DMKLOG
DMKRWA DMKSPS DMKUSQ DMKVMC DMKVMF
DMKVSW

Chapter 15. General Design Changes

This section also applies to CP, CMS, XEDIT and IPCS. Information about GCS and TSAF is not included because the source code for GCS and TSAF modules is optional. Information about AVS is not included because the source code is not distributed.

This section summarizes the new and changed modules, control blocks, macros, execs, and so on. It also included a list of CP module splits.

New CP Modules

DMKACP DMKBIM DMKCDD DMKCDN DMKCFI
DMKCFK DMKCFX DMKCNT DMKCQD DMKCQF
DMKCQK DMKCQZ DMKCSY DMKCSZ DMKDIG
DMKDRE DMKDTR DMKDTS DMKFDA DMKFDL
DMKFEV DMKFIO DMKFSM DMKGIV DMKGRD
DMKGRE DMKGRI DMKGTR DMKHPU DMKHPV
DMKHVG DMKHVH DMKHVI DMKIUT DMKIUU
DMKIUV DMKIUW DMKMAP DMKMIG DMKRPL
DMKSPR DMKTCW DMKTCX DMKTED DMKTEE
DMKTEF DMKTEH DMKTRF DMKTRO DMKTRV
DMKTTD DMKTTR DMKUDV DMKVCY DMKVES
DMKVMF DMKWRU

Changed CP Modules

DMKACO DMKALG DMKALO DMKAPS DMKAPT
DMKAPU DMKAPV DMKAPW DMKAPZ DMKBIO
DMKCAC DMKCAO DMKCCD DMKCCF DMKCCH
DMKCCO DMKCCS DMKCCW DMKCFB DMKCFC
DMKCFF DMKCFG DMKCFO DMKCFP DMKCFT
DMKCFY DMKCKF DMKCKH DMKCKR DMKCKS
DMKCMD DMKCPI DMKCPJ DMKCPN DMKCPP
DMKCPU DMKCPW DMKCPZ DMKCQC DMKCQF
DMKCQG DMKCQH DMKCQP DMKCQQ DMKCQR
DMKCQS DMKCQV DMKCQY DMKCRM DMKCSP
DMKCST DMKCSU DMKCSW DMKCSX DMKDAD
DMKDDR DMKDEF DMKDIR DMKDMP DMKDRD
DMKDRE DMKDSB DMKDSP DMKFMT DMKFPS
DMKFRE DMKFRT DMKGRC DMKGRF DMKGRG
DMKHPS DMKHVC DMKHVD DMKHVE DMKHVF
DMKHVH DMKHVI DMKIDR DMKIDU DMKIOC
DMKIOE DMKIOF DMKIOJ DMKIOS DMKIOT
DMKIUA DMKIUB DMKIUC DMKIUE DMKIUG
DMKIUJ DMKIUL DMKIUN DMKIUP DMKIUS
DMKJRL DMKLNK DMKLOG DMKLOH DMKLOJ
DMKMCC DMKMCH DMKMES DMKMIA DMKMNI
DMKMNL DMKMNT DMKMON DMKMOO DMKOVR
DMKPAG DMKPAH DMKPGS DMKPGT DMKPGU
DMKPMA DMKPRG DMKPRV DMKPSA DMKPST
DMKPTR DMKPTS DMKPTT DMKREI DMKRET

DMKRGC DMKRSP DMKRWA DMKSAD DMKSCH
DMKSEC DMKSPS DMKSPT DMKSRM DMKSTD
DMKSTP DMKSTR DMKSVC DMKSVD DMKSWA
DMKSYS DMKTDK DMKTED DMKTEE DMKTEF
DMKTEM DMKTES DMKTMR DMKTOD DMKTRA
DMKTRC DMKTRD DMKTRP DMKTRT DMKTRU
DMKTRX DMKTTX DMKUDR DMKUDU DMKUDV
DMKUNT DMKUSC DMKUSO DMKUSP DMKVCN
DMKVCQ DMKVCR DMKVCS DMKVCU DMKVCV
DMKVDA DMKVDB DMKVDC DMKVDE DMKVDG
DMKVDR DMKVDS DMKVDT DMKVER DMKVFD
DMKVFE DMKVIO DMKVMC DMKVMF DMKVMG
DMKVSC DMKVSW DMKXST

CP Module Splits

Original Modules	**Split Into**	
DMKACO	DMKACO	DMKACP
DMKBIO	DMKBIO	DMKBIM
DMKCDB	DMKCDB	DMKCDD
DMKCDM	DMKCDM	DMKCDN
DMKCFC	DMKCFC	DMKCFB
DMKCFG	DMKCFG	DMKCFI
DMKCFU	DMKCFU	DMKCFX
DMKCQG	DMKCQG	DMKCQF
DMKCFY	DMKCFY	DMKCFX
DMKCNS	DMKCNS	DMKCNT
DMKCQY	DMKCQY	DMKCQZ
DMKCSX	DMKCSX	DMKCSY
DMKDLB	DMKDIB	DMKDIG
DMKDRD	DMKDRD	DMKDRE
DMKHPS	DMKHPS	DMKHPV
DMKHVC	DMKHVC	DMKHVI
DMKIUC	DMKIUC	DMKIUU
DMKIUS	DMKIUS	DMKIUT
DMKTEF	DMKTEF	DMKTEH
DMKTRT	DMKTRT	DMKTRV
DMKUDU	DMKUDU	DMKUDV
DMKVCR	DMKVCR	DMKVCY
DMKVMC	DMKVMC	DMKVMF
DMKWRM	DMKWRM	DMKWRV
DMKSPS	DMKSPS	DMKSPR

New CP Control Blocks, Copy Files, and Macros

ALRTREC BIMBLOK DCRBLOK FEVENTRY
FPOBLOK FREBLOKS FRELBLOK FSMPLIST
GIVEBLOK LSRBLOK MBPLIST SPTBLOKS
VESPARMS FPOBLOK FEVENTRY FRELBLOK
FSMPLIST FREBLOKS

Changed CP Control Blocks, Copy Files, and Macros

ACIPARMS ALLOC ALOCBLOK APPCVM
BIRBLOK CALL CCHREC CONEXT
CCWSAVE CORE CPEXBLOK CPTRAP
DLSRC DCBLOK DEXDEC EVTBLOK
EQU FRECOM IOBLOKS IOER
IPARML INTBLK IUCV IUCVBLOK
MDRREC MN007 MONBLOKS NAMENTRY
OBRRECN PFKTABLE PPMAP PSA
RETBUF RBLOKS RCUBLOK RDEVBLOK
SFBLOK SNARBLOK SPOOL SRTBLOK
SSBLOK SSCBLOK SYSMON TCXBLOK
TED TNSREC TRACEFMT TRDISPLY
TREXT TRPBLOK TRTDATA UDIRECT
UMACBLOK VBBLOKS VDEVBLOK VMBLOK
WEIBLOK

New CMS and XEDIT Modules

DMSAXR DMSBOF DMSBUS DMSCAX DMSCCM
DMSCCR DMSCDB DMSCDN DMSCFL DMSCFX
DMSCGT DMSCIA DMSCLL DMSCRI DMSCRL
DMSCRM DMSCRR DMSCRS DMSCSL DMSCSM
DMSCUT DMSDCT DMSERC DMSERP DMSERX
DMSETB DMSETP DMSEXR DMSGEN DMSGEO
DMSGEP DMSGLO DMSICT DMSINX DMSIUH
DMSMDP DMSNAD DMSOSC DMSPBS DMSPSV
DMSQRC DMSRCS DMSRFF DMSSAA DMSSAP
DMSSEC DMSSFA DMSSFD DMSSLD DMSSNC
DMSSPA DMSSTR DMSSUP DMSTBA DMSTBB
DMSTBC DMSTBD DMSTBE DMSTBG DMSTBS
DMSTRC DMSTRI DMSWEN DMSWIN DMSWIT
DMSWVD DMSXFI DMSXFW DMSXTS

Changed CMS and XEDIT Modules

DMSABN DMSACC DMSACF DMSACG DMSACM
DMSACP DMSACR DMSACS DMSALA DMSALU
DMSAMS DMSARD DMSARE DMSARN DMSASM
DMSASN DMSAST DMSAUD DMSBOP DMSBRD
DMSBTB DMSBTP DMSBWR DMSCAT DMSCCK
DMSCCS DMSCCX DMSCIO DMSCIT DMSCLN
DMSCLR DMSCLS DMSCMP DMSCPF DMSCPY
DMSCRD DMSCSF DMSCUR DMSCWR DMSCWT
DMSDBD DMSDBG DMSDCS DMSDDL DMSDEF
DMSDEL DMSDFT DMSDID DMSDIE DMSDIO
DMSDIP DMSDLB DMSDLK DMSDOS DMSDRP
DMSDSK DMSDSL DMSDSV DMSEDI DMSEDX
DMSEIO DMSERD DMSERR DMSERS DMSETR
DMSEXD DMSEXE DMSEXG DMSEXI DMSEXL
DMSEXM DMSEXQ DMSEXT DMSFCH DMSFLD
DMSFLE DMSFLO DMSFNC DMSFNS DMSFOR
DMSFRA DMSFRE DMSFRI DMSFRM DMSFRO

DMSFRR	DMSFRS	DMSFRU	DMSGET	DMSGIO
DMSGLB	DMSGND	DMSGRN	DMSGTS	DMSGVE
DMSHDE	DMSHDI	DMSHDS	DMSHID	DMSHLB
DMSHLC	DMSHLD	DMSHLI	DMSHLL	DMSHLM
DMSHLP	DMSHLR	DMSHLS	DMSHLT	DMSHLZ
DMSHTB	DMSICP	DMSIDE	DMSIFI	DMSIMA
DMSINA	DMSINC	DMSIND	DMSINI	DMSINM
DMSINN	DMSINQ	DMSINS	DMSINT	DMSINV
DMSIOW	DMSITE	DMSITI	DMSITM	DMSITP
DMSITS	DMSIUC	DMSLAD	DMSLAF	DMSLBD
DMSLBM	DMSLBT	DMSLDF	DMSLDR	DMSLDS
DMSLFS	DMSLGT	DMSLIB	DMSLIO	DMSLKD
DMSLLU	DMSLMX	DMSLOA	DMSLOS	DMSLSB
DMSLST	DMSMAX	DMSMCM	DMSMES	DMSMGC
DMSMGD	DMSMGE	DMSMGM	DMSMGX	DMSMIN
DMSMOD	DMSMVE	DMSNAM	DMSNCP	DMSNGP
DMSNXD	DMSNXL	DMSNXM	DMSNUC	DMSOSR
DMSOVR	DMSOVS	DMSPAR	DMSPCA	DMSPCB
DMSPCC	DMSPCL	DMSPCR	DMSPCT	DMSPCW
DMSPDB	DMSPGM	DMSPHA	DMSPIO	DMSPKR
DMSPKT	DMSPLC	DMSPMD	DMSPNT	DMSPOA
DMSPOC	DMSPOE	DMSPOL	DMSPON	DMSPOP
DMSPOQ	DMSPOR	DMSPOS	DMSPPL	DMSPPO
DMSPRB	DMSPRE	DMSPRI	DMSPRJ	DMSPRT
DMSPRV	DMSPSM	DMSPST	DMSPTC	DMSPTK
DMSPTL	DMSPTR	DMSPTT	DMSPUN	DMSPUT
DMSPVF	DMSQRF	DMSQRG	DMSQRH	DMSQRS
DMSQRT	DMSQRU	DSMQRV	DMSQRW	DMSQRX
DMSQRY	DMSRDC	DMSRDR	DMSREF	DMSREV
DMSRES	DMSREX	DMSRLD	DMSRNM	DMSROS
DMSROU	DMSRRV	DMSRSF	DMSRSV	DMSRXE
DMSSAB	DMSSBD	DMSSBS	DMSSCA	DMSSCL
DMSSCT	DMSSEB	DMSSEF	DMSSEG	DMSSET
DMSSEV	DMSSFD	DMSSHO	DMSSIZ	DMSSLG
DMSSLN	DMSSMG	DMSSMN	DMSSOP	DMSSPA
DMSSPI	DMSSPR	DMSSQS	DMSSRE	DMSSRQ
DMSSRS	DMSSRT	DMSSRV	DMSSSK	DMSSSY
DMSSTA	DMSSTG	DMSSTK	DMSSTM	DMSSTP
DMSSTQ	DMSSTT	DMSSTX	DMSSVN	DMSSVQ
DMSSVT	DMSSVU	DMSSYN	DMSTCA	DMSTIO
DMSTLB	DMSTLM	DMSTMA	DMSTMR	DMSTPD
DMSTPE	DMSTPF	DMSTPG	DMSTPH	DMSTPI
DMSTPJ	DMSTRC	DMSTRT	DMSTVS	DMSTYP
DMSUPD	DMSUSR	DMSUTL	DMSVAN	DMSVAS
DMSVAX	DMSVIB	DMSVIP	DMSVIS	DMSVLD
DMSVLT	DMSVSR	DMSWAT	DMSWBX	DMSWEX
DMSWGF	DMSWID	DMSWIF	DMSWIM	DMSWIO
DMSWIR	DMSWIS	DMSWLR	DMSWLW	DMSWMI
DMSWMM	DMSWMO	DMSWMU	DMSWMX	DMSWQI
DMSWQM	DMSWRD	DMSWRT	DMSWST	DMSWTE
DMSWVC	DMSWVE	DMSWVL	DMSWVQ	DMSWVS
DSMWVT	DMSWVX	DMSXBG	DMSXCG	DMSXCM
DMSXCN	DMSXCP	DMSXCT	DMSXDC	DMSXDS
DMSXED	DMSXER	DMSXFC	DMSXFD	DMSXFL
DMSXGT	DMSXIN	DMSXIO	DMSXMA	DMSXMB

DMSXMC DMSXMD DMSXML DMSXMS DMSXPO
DMSXPT DMSXPX DMSXQR DMSXRE DMSXSC
DMSXSD DMSXSE DMSXSF DMSXSU DMSXSS
DMSXST DMSXTE DMSXTR DMSXUP DMSXWS
DMSZAP

Deleted CMS Modules

DMSHLE

CMS Module Splits

Original Modules	Split Into	
DMSIUC	DMSIUC	DMSIUH

New CMS and XEDIT Control Blocks, Copy Files, and Macros

CSLENTRY CSLEXIT CSLFPI CSLFRWA
CSLFRWB CSLGETP CSLGTWA CSLGTWB
DIRBUFF DMSPSIB DMSSYLST EXSBUFF
FPERROR GLVBLOCK PROFCLST SGMTEXIT
WUERROR X$CLST$X X$TMPL$X

Changed CMS and XEDIT Control Blocks, Copy Files, and Macros

ABWSECT ADTSECT AFTSECT CMSIUCV
EXECUTE EXISBLK FCBSECT FDEFSECT
FSTD FSTSECT FVSECT HNDIUCV
IUCVIDBK IUCVTAB NUCEXT NUCON
PROFTMPL QRYWORK REXEXT SCBLOCK
SYNSUB TABENT ZBLOCKS

New CMS and XEDIT Execs

CSLGEN CSLLIST DATADEF

Changed CMS and XEDIT Execs

CANCEL CMSGEND CMSLOAD CMSSERV
CONVERT DEFAULTS LANGGEN LANGMERG
SYSPROF VMFAPTXT VMFLOAD VMFNLS

Miscellaneous New CMS Files

DMSCSL TEXT DMSSAA TEXT PROFCSLT XEDIT

New IPCS Modules

DMMINP DMMLUN DMMMCS DMMPTR DMMSEL
DMMSLD DMMSLT DMMSVT DMMTBL DMMTFR
DMMTIM DMMTPL DMMTRP DMMTRS DMMTSC
DMMTSL DMMTSP DMMTSU

Changed IPCS Modules

DMMCHN DMMCLR DMMCMS DMMCOM DMMCPA
DMMCVD DMMDCC DMMDCD DMMDCM DMMDCO
DMMDCP DMMDID DMMDIN DMMDIR DMMDSC
DMMEDM DMMFED DMMFEX DMMHEX DMMINI
DMMINT DMMLOC DMMMAP DMMMOD DMMPRM
DMMPRO DMMPRT DMMREG DMMSCN DMMSCR
DMMSEA DMMSEG DMMSTA DMMSUM DMMTAB
DMMTIU DMMTRC DMMTRD DMMVAL DMMVMF
DMMWRT

IPCS Module Splits

Original Modules	Split Into
DMMDIN	DMMDIN DMMTBL
DMMDSC	DMMDSC DMMINP
DMMPRT	DMMPRT DMMINP
DMMTRD	DMMTRD DMMTRS DMMTFR

New IPCS Control Blocks and Macros

CCODE

Changed IPCS Control Blocks and Macros

MSGP SEGMAP TABCOMM

New IPCS Copy Files

GSA MACENTRY SLCDATA SVC199
TSC TSU

Changed IPCS Copy Files

ADSSR DMMSAVE DMPEQU ENTRYMAP
HEADMAP HDRPLIST IDASECT SAVESECT
WORKCP

New IPCS Execs

TRAPFILE

Changed IPCS Execs

APAR

Changed Miscellaneous IPCS Files

CUSTOMER PROFILE

Chapter 16. Object Code Only (OCO) Modules

This section lists the OCO modules in VM/SP Release 6.

AGWACF	AGWACS	AGWACT	AGWCMC	AGWCMD
AGWCMF	AGWCMH	AGWCMJ	AGWCMQ	AGWCTR
AGWDGW	AGWDSP	AGWEOT	AGWFAS	AGWGAB
AGWGGW	AGWGSP	AGWIIS	AGWINI	AGWIOM
AGWIUE	AGWMGM	AGWOLM	AGWPGT	AGWSET
AGWTIM	AGWTRM	AGWTRS	AGWUSM	AGWVAF
AGWVBC	AGWVCB	AGWVCC	AGWVCP	AGWVDA
AGWVFD	AGWVGB	AGWVME	AGWVMF	AGWVMP
AGWVMR	AGWVMS	AGWVMX	AGWVRA	AGWVRC
AGWVTA	AGWVTC	AGWVTD	AGWVTE	AGWVTF
AGWVTP	AGWVTS	AGWVTT	AGWVTX	AGWZAM
AGWZDS	AGWZEN	AGWZEX	AGWZGC	AGWZGW
AGWZMA	AGWZMN	AGWZMO	AGWZRS	AGWZSG
AGWZTD	AGWZTE	AGWZTF	AGWZTO	AGWZTR
AGWZTS	CSIYTEXB	DMSAAC	DMSAAL	DMSABA
DMSACB	DMSACD	DMSACL	DMSACN	DMSACU
DMSADE	DMSAEA	DMSAED	DMSAFL	DMSAID
DMSAIH	DMSAIN	DMSALD	DMSAMO	DMSAPR
DMSAPS	DMSAPT	DMSARB	DMSARS	DMSART
DMSARW	DMSASA	DMSASC	DMSASE	DMSASF
DMSASL	DMSASP	DMSASR	DMSASS	DMSASU
DMSATD	DMSATP	DMSATR	DMSATY	DMSAWD
DMSAWE	DMSAXC	DMSAXL	DMSATS	DMSAXM
DMSAXP	DMSAXR	DMSAXT	DMSAXU	DMSBOF
DMSBUS	DMSCCM	DMSCCR	DMSCDN	DMSCFX
DMSCGT	DMSCIA	DMSCSL	DMSDFT	DMSERD
DMSETB	DMSFNS	DMSICT	DMSINI	DMSINQ
DMSINV	DMSJAC	DMSJAL	DMSJAS	DMSJAU
DMSJBE	DMSJBR	DMSJCA	DMSJCB	DMSJCC
DMSJCD	DMSJCE	DMSJCK	DMSJCL	DMSJCM
DMSJCO	DMSJCR	DMSJCS	DMSJCU	DMSJDC
DMSJDF	DMSJDG	DMSJDL	DMSJDO	DMSJDS
DMSJDU	DMSJED	DMSJEF	DMSJEN	DMSJER
DMSJES	DMSJEU	DMSJEX	DMSJFC	DMSJFI
DMSJFM	DMSJFO	DMSJFR	DMSJFW	DMSJGA
DMSJGD	DMSJGF	DMSJGK	DMSJGL	DMSJGS
DMSJGT	DMSJLA	DMSJLB	DMSJLD	DMSJLU
DMSJMF	DMSJNA	DMSJOB	DMSJOC	DMSJQC
DMSJQM	DMSJRA	DMSJRB	DMSJRC	DMSJRE
DMSJRL	DMSJRN	DMSJRO	DMSJUA	DMSJUC
DMSJUQ	DMSJUU	DMSJWA	DMSJWB	DMSJWC
DMSJWE	DMSJWG	DMSJWP	DMSJWR	DMSJWS
DMSJXD	DMSJXF	DMSLMX	DMSOPN	DMSPBS
DMSQRC	DMSRCM	DMSSAA	DMSSAP	DMSSDE
DMSSDM	DMSSEC	DMSSNC	DMSSUP	DMSTBA
DMSTBB	DMSTBC	DMSTBD	DMSTBE	DMSTBG
DMSTBS	DMSTRC	DMSTRS	DMS2CA	DMS2CB
DMS2CC	DMS2CD	DMS2LA	DMS2TA	DMS2TB
DMS2TC	DMS2TD	DMS2TE	DMS3AA	DMS3AB
DMS3AC	DMS3AD	DMS3AF	DMS3AG	DMS3AH

DMS3AI DMS3AJ DMS3AK DMS3AL DMS3AM
DMS3AN DMS3AO DMS3AP DMS3AQ DMS3AR
DMS3AT DMS3AU DMS3AV DMS3AW DMS3AX
DMS3AY DMS3AZ DMS3BA DMS3BB DMS3BC
DMS3BD DMS3BE DMS3BF DMS3BG DMS3BH
DMS3BJ DMS3BK DMS3BL DMS3BM DMS3BN
DMS3BO DMS3BP DMS3BQ DMS3BR DMS3BS
DMS3BT DMS3NA DMS3NB DMS3NC DMS3ND
DMS3NE DMS3NF DMS3N1 DMS3N2 DMS3N3
DMS3N4 DMS3N5 DMS3N6 DMS3N7 DMS3N8
DMS3N9 DMS3RA DMS3RB DMS3RC DMS3RD
DMS3RE DMS3RF DMS3SB DMS3SC DMS3SM
DMS3SP DMS3SR DMS3TA DMS3TB DMS3TC
DMS3TD DMS3TE DMS3TF DMS3TP DMS3TR
DMS3VA DMS3VB DMS3VC DMS3WM DMS3XG
DMS3XH DMS3XL DMS3XM DMS3XP DMS3XU
DMS4AA DMS4AB DMS4AC DMS4AD DMS4AE
DMS4AF DMS4AG DMS4AH DMS4AI DMS4AJ
DMS4AK DMS4AL DMS4AM DMS4AN DMS4AO
DMS4AP DMS4AQ DMS4AR DMS4AS DMS4AT
DMS4AU DMS4AV DMS4AW DMS4AX DMS4AY
DMS4AZ DMS4BA DMS4BB DMS4BC DMS4BD
DMS4BE DMS4BF DMS4BG DMS4BH DMS4BI
DMS4BJ DMS4BK DMS4BL DMS4BM DMS4BN
DMS4BO DMS4BP DMS4BQ DMS4BR DMS4BS
DMS4BT DMS4BU DMS4BV DMS4BW DMS4BX
DMS4BY DMS4CI DMS4CJ DMS4DA DMS4DB
DMS4DC DMS4DD DMS4DE DMS4DF DMS4DG
DMS4DH DMS4DI DMS4DJ DMS4DK DMS4DL
DMS4DM DMS4DN DMS4DO DMS4DP DMS4DQ
DMS4DR DMS4DS DMS4DT DMS4DU DMS4DV
DMS4DW DMS4DX DMS4DY DMS4DZ DMS4EA
DMS4EB DMS4EC DMS4ED DMS4EE DMS4EF
DMS4EG DMS4EH DMS4EX DMS4EY DMS4EZ
DMS4FA DMS4FB DMS4FD DMS4FE DMS4FG
DMS4FH DMS4GC DMS4GD DMS4GE DMS4GF
DMS4GG DMS4GH DMS4GI DMS4GJ DMS4GK
DMS4GL DMS4GM DMS4GN DMS4GO DMS4GP
DMS4GQ DMS4GR DMS4GS DMS4GW DMS4GY
DMS4GZ DMS4HA DMS4HB DMS4HC DMS4HD
DMS4HE DMS4HG DMS4HZ DMS4JA DMS4JB
DMS4JC DMS4JD DMS4JE DMS4JF DMS4JG
DMS4JH DMS4JI DMS4JJ DMS4JK DMS4JL
DMS4JS DMS4KA DMS4KB DMS4KC DMS4KD
DMS4KE DMS4KF DMS4KH DMS4KI DMS4KJ
DMS4KK DMS4KM DMS4KO DMS4KP DMS4LA
DMS4LB DMS4LC DMS4LD DMS4LE DMS4LF
DMS4LG DMS4LH DMS4LI DMS4LJ DMS4LK
DMS4LL DMS4LM DMS4LN DMS4LO DMS4LP
DMS4LQ DMS4LR DMS4LS DMS4LT DMS4LU
DMS4LV DMS4LW DMS4LX DMS4LY DMS4NA
DMS4NB DMS4NC DMS4ND DMS4NE DMS4NF
DMS4NG DMS4NH DMS4NI DMS4NJ DMS4NK
DMS4NL DMS4NM DMS4NO DMS4NP DMS4NQ
DMS4NR DMS4NS DMS4NT DMS4NU DMS4NV

DMS4NW DMS4NX DMS4ON DMS4PA DMS4PB
DMS4PC DMS4PD DMS4PE DMS4PF DMS4PG
DMS4PH DMS4PI DMS4QA DMS4QB DMS4QC
DMS4QD DMS4QE DMS4QF DMS4QG DMS4QH
DMS4QJ DMS4QK DMS4QL DMS4QM DMS4QN
DMS4QO DMS4QP DMS4QQ DMS4QR DMS4QS
DMS4QX DMS4RA DMS4RB DMS4RC DMS4RD
DMS4RE DMS4RF DMS4RG DMS4RH DMS4RI
DMS4RJ DMS4RL DMS4RM DMS4RN DMS4RX
DMS4RY DMS4RZ DMS4SA DMS4SB DMS4SR
DMS4SS DMS5AA DMS5BA DMS5BB DMS5BC
DMS5BD DMS5BE DMS5BF DMS5BG DMS5CA
DMS5CB DMS5CC DMS5CD DMS5DA DMS5DB
DMS5DC DMS5DD DMS5DE DMS5DF DMS5DG
DMS5DH DMS5DI DMS5DJ DMS5DK DMS5DS
DMS5FA DMS5FB DMS5FC DMS5FD DMS5FE
DMS5FF DMS5GA DMS5GB DMS5GC DMS5GM
DMS5GN DMS5GO DMS5HA DMS5HB DMS5HC
DMS5IA DMS5IB DMS5IC DMS5ID DMS5IE
DMS5IF DMS5IG DMS5IL DMS5IN DMS5JA
DMS5JB DMS5JC DMS5JD DMS5JE DMS5JF
DMS5MA DMS5MB DMS5MC DMS5MT DMS5MU
DMS5MV DMS5MX DMS5NA DMS5NB DMS5NC
DMS5ND DMS5NE DMS5NF DMS5NG DMS5PR
DMS5QA DMS5QB DMS5QE DMS5SB

This section lists the OCO modules in VM/SP HPO.

DMKVFI DMKBTS

Appendix A. Overview of VM/SP HPO prior to Release 6

This appendix introduces you to IBM Virtual Machine/System Product High Performance Option (VM/SP HPO). It provides an overview of VM/SP HPO prior to Release 6.

It:

- Describes the relationship of VM/SP HPO to its base program product, VM/SP,
- Describes the users of VM/SP HPO,
- Describes the programming enhancements, microcode assists, and additional features contained in VM/SP HPO prior to VM/SP HPO Release 6.

What Is VM/SP HPO?

VM/SP HPO is a separately orderable licensed program that can be installed and executed in conjunction with VM/SP.

The VM/SP HPO program package adds additional components for Release 6 and contains all CP components. It requires installation of VM/SP or an equivalent program. When you install and use VM/SP HPO in conjunction with the prerequisite VM/SP release, you obtain an operating system that extends the capabilities of the VM/SP with programming enhancements, support for microcode assists, and additional features.

Who Uses VM/SP HPO?

VM/SP HPO offers enhancements for large or complicated system environments. These enhancements include system management performance improvements, additional processor and I/O support, and enhanced MVS/SP and VSE support.

What Are the Main Functions of VM/SP HPO?

VM/SP HPO contains all the base functions and enhances the capabilities of VM/SP with the following:

- System management performance improvements:
  - Storage management and simplified tuning
  - System lock utilization
  - Less than 16 MB constraint relief
  - Predictable system behavior and performance
  - Hardware segment protect follow-on
  - Free storage management improvements
  - Dispatching and scheduling modifications
  - Enhanced paging subsystem support
  - Expanded Storage support
- Additional processor support:
  - Storage support
  - Performance enhancements for dyadic and dual processors
  - 3090 processor support

- Additional I/O support:
  - Paging allocation order
  - IBM 3880 Model 11 and 21 Storage Subsystems support
  - IBM 3880 Model 13 and 23 Storage Subsystems support
  - Channel support
  - Number of virtual devices supported
- Enhanced MVS/SP support:
  - Preferred machine assist support
  - Control switch assist extensions to preferred machine assist
  - 3033 Extension Feature virtual machine assist enhancement
  - Availability in the MVS/SP V=R environment
  - Single processor mode operational enhancements
- Vector Facility.

System Management Performance Improvements

Storage Management and Simplified Tuning

This support addresses performance for users in constrained storage environments. The objectives of this support are:

- Improve the storage management of large working sets, shared pages, and the less-than-16MB area. Core table scan becomes the primary method for free list replenishment. In addition, the disposable page collector is eliminated.
- Change the balance between swapping and paging so that swapping will predominate.
- Reduce overhead at queue drop and queue add. When a virtual machine drops from queue, its pages will no longer be logically swapped and trimmed.
- Simplify tuning.

Use of the System Lock

In Release 6, there is less use of the system lock and, therefore, less contention for it. This should improve system performance.

Examples of some of the functions that are now executed without the system lock are:

- Part of Diagnose X'18'
- Part of the external interrupt handler
- Most scheduler calls
- System modules that call DMKPTRAN and do not hold the system lock.

In addition, several functions use less of the system lock when executed. As a result of the command logic that builds multiple-line command responses, the VMSTKO or 'VM Stacked Output' has been fixed.

Less than 16 MB Constraint Relief

CP references certain pages above the 16MB line. This relieves constraint below the 16 MB line since CP no longer needs to copy these pages below 16MB to reference them.

Privileged operation simulation has been changed to reduce the number of times a page must be moved below the 16MB line. When a privileged operation occurs, the system now copies the privileged instruction to the VMBLOK (from wherever it was). It no longer needs to move the page containing the instruction below the line before it does the copy. Likewise, the new PSW that is the operand of the LPSW instruction can be directly obtained from above the line.

Pseudo-pages, privileged operations, LPSW operands, and I/O data used by SWPTABLE migration may be located and referenced above the 16MB line. The system no longer moves them below the 16MB line before reinforcing them.

In addition, certain pages that could not move above the 16MB line in previous releases are now moved above the line before being paged out. These include virtual page zeros and pageable system modules.

Predictable System Behavior and Performance

Changes to the scheduler provide more predictable system behavior and more predictable performance in a storage-constrained environment.

Before moving a virtual machine from the eligible list to run list, the scheduler tests for sufficient processor time as well as sufficient main storage. If processor time is not available, a virtual machine is promoted from the eligible list to run list only if it has an earlier deadline (higher priority) than the last virtual machine already on the run list.

Free Storage Management

To perform system-related functions such as building control blocks, processing I/O operations, or building save areas, the control program obtains free storage. Requests for free storage are processed by obtaining the required storage from fixed-size storage groups called *subpools*.

Free storage management improvements in VM/SP HPO process free storage requests more efficiently. These improvements make available more subpools in a greater variety of sizes. Processing required to return free storage to the system is also more efficient.

Overall, free storage improvements in VM/SP HPO reduce the amount of processing overhead associated with allocating and releasing free storage.

Dispatching and Scheduling Modifications

This support incorporates modifications that improve the efficiency of virtual machine dispatching functions. These changes are provided in the following areas.

Dispatch List

The VM/SP HPO control program scheduler maintains the run list, which contains virtual machines in either executable or nonexecutable states. Prior to this support, the dispatcher examined the run list in its search for an executable virtual machine. A new dispatch list is implemented to reduce the amount of time required to select an executable virtual machine. The dispatch list contains a subset of the virtual machines in the run list; however, only dispatchable virtual machines are placed in the dispatch list. The dispatcher now references only the dispatch list when selecting a virtual machine to run.

Using the dispatch list, the dispatcher no longer has to scan the run list in search of an eligible virtual machine. The scheduler's maintenance of the run list is unchanged.

Dispatcher Fast Path

After a virtual machine is dispatched, frequently an event occurs that stops the virtual machine's execution. An event such as an I/O interruption, a page fault, or a disk diagnose (DIAGNOSE X'18') does not change the status of the virtual machine, but the event does interrupt the virtual machine's execution. If the virtual machine is operating in basic control mode and is interrupted by one of the events described, the control program saves information regarding the status of the virtual machine. With this information, there is no need to reexamine the virtual machine's status. The virtual machine can be redispatched without resetting the required states and modes for the system. The next time the virtual machine is dispatched, it executes using the dispatcher fast path.

Basic control mode virtual machines (CMS, for example) can take advantage of the dispatcher fast path when the processor adheres to all of the following characteristics:

- Virtual machine assist is on
- The system is generated and operating in AP/MP mode
- Segment protection extension is available and used to prohibit access to all shared segments.

The dispatcher fast path results in greater processor utilization. If, upon examination of new status fields, it can be determined that the virtual machine is eligible for the dispatcher fast path, other tests for dispatchability are bypassed, reducing the time spent examining and readying an executable virtual machine.

Paging Subsystem

Enhanced paging subsystem support, including block paging support, improves paging performance. This support uses main storage as a high-speed buffer. The system groups together user pages that are likely to be used together. Single start I/O (SIO) operations swap these pages in and out of main storage. The SYSPAG macro provides for more efficient use of DASD and gives the system programmer a finer tuning capability over the paging and spooling resources.

Enhanced paging subsystem support simplifies CP code by handling Fixed Block Architecture (FBA) devices and Count Key Data (CKD) devices on pseudo-cylinder boundaries for paging and spooling devices.

Storage Support

Expanded storage is a paging assist on the 3090 processor. It is used by CP as a high-speed paging area. Only CP has access to it; virtual machines cannot access expanded storage.

The SYSPAG or SYSXSTOR macros let you specify whether you want the system to use expanded storage as a swap area, a preferred paging area, or both.

To collect information on expanded storage's effect on system performance, several MONITOR records have been expanded or enhanced.

The DASD Dump Restore program (DDR) is extended to support expanded storage in a stand-alone environment. It now prints selected pages of expanded storage in hexadecimal and EBCDIC on a printer or terminal.

Additional Processor Support

Extended Storage Support

This feature allows the VM/SP HPO control program to manage real storage on processors that have increased storage capacity. When the amount of real storage defined exceeds 16 MB, the control program uses the extended storage area (above 16MB) as an additional dynamic paging area. Though the control program itself must remain in the real storage area below 16MB, it uses the extended storage area for virtual machine pages.

Given this flexibility, you can configure the system and the extended storage area for:

- Exclusive use by the control program
- Exclusive use by the MVS/SP preferred guest
- Partitioned use by both the control program and the MVS/SP preferred guest.

With extended storage configured and online, systems previously constrained by limited availability of real storage will be able to improve system throughput.

Performance Enhancements for Dyadic and Dual Processors

The performance on dyadic and dual processors with store-in caches is improved through implementing the control block alignment, local dispatcher and free storage queues for each processor, and dispatcher queue scanning.

Frequently used control blocks are aligned on cache line boundaries. Dispatcher and free storage queues are provided for each processor. Thus, two processors will modify and reference the same storage areas less frequently. In addition, when a processor completes its dispatchable work, it scans the dispatcher queues of the other processor instead of going into a wait state. Thus, the other processor no longer has to signal the idle processor when there is a piece of work that the idle processor can perform.

IBM 3090 Processor Complex

VM/SP HPO supports the 3090 processor as a dyadic processor in System/370 mode.

The 3090 processor yields high performance and has the flexibility to meet expanding installation requirements. It combines basic support with options to reduce channel contention and to add optional storage for use by CP. Thus, it provides faster response time and increases overall system performance at an improved price-to-performance ratio.

The 3090 service processor provides I/O configuration and storage information and executes commands that modify the real-system configuration.

Additional I/O Support

Paging Allocation Order

The parameter ORDER= is added to the SYSPAG macro to improve the distribution of I/O activity over available channels and control units. The default is ORDER=SYSTEM. If you accept the default, you can add devices or reorganize your I/O configuration without reoptimizing your SYSPAG macros. The system verifies that you distributed I/O activity in a reasonable order and attempts to improve the order.

However, you may specify ORDER=USER if you want to preserve the order you specified.

IBM 3880 Model 11 and 21 Storage Subsystems

Excessive system paging demands can cause active pages to be transferred from real storage to auxiliary storage. Preferred system paging keeps frequently used pages more readily available to the system. The IBM 3880 Model 11 or 21 control unit, when operating in conjunction with 3350 direct-access storage devices, establishes a paging subsystem that the control program uses to implement preferred system paging.

If a 3880 Model 11 or 21 storage subsystem is installed and properly defined to the system, active pages are transferred into a high-speed buffer area, or *cache*. Subsequent paging requests quickly transfer the necessary pages back into real storage.

The 3880 Model 11 or 21 storage subsystem also has extended addressing capability. When you supply a real device address for the 3880, the control program automatically generates three additional addresses. Multiple addresses allows the 3880 to operate in two modes of operation:

- Direct mode uses I/O operations to format and allocate the 3350 direct access storage devices.
- Paging mode uses I/O operations to access the cache.

This flexibility improves data transfer rates by providing access to I/O devices through several paths.

The VM/SP HPO MONITOR is enhanced to collect status information for the 3880 Model 11 or 21 storage subsystem. Using MONITOR class DASTAP, you receive information regarding the configured and available size of the subsystem cache in addition to other performance-related indicators.

DMKFMT, the FORMAT/ALLOCATE program, formats 3880 Model 21 devices without dummy records.

IBM 3880 Model 13 and 23 Storage Subsystems

The 3880 Models 13 and 23 are high-performance cached DASD subsystems designed especially for nonpaging applications (application data that resides on DASD that is not defined as paging, swapping, spooling, or dump area). This support is provided to improve performance for MVS or CMS guest virtual machines running with 3380 DASDs. The performance improvement is accomplished principally by maximizing the number of read accesses that can be satisfied by accessing the cache (subsystem storage) copy. This support includes:

- Selective use of the cache by virtual device and minidisk as specified in the user's directory.
- Operator commands to enable or disable the caching function so that the cache can be taken offline. In addition, the operator commands provide the ability to select and display the devices that are eligible to use the cache.
- A directory option on the MDISK control statement to specify which minidisks on a 3880 Model 13 and 23 device can be used for caching.
- Recognition of 3880 Model 13 and 23 hardware errors.
- MONITOR records that cache usage from both storage directors.
- Dedicated guest virtual machine control of the cache.

For effective use of the caching function, frequently used data must remain in the cache and must not be displaced by infrequently used data.

Channel Support

Extended channel support allows an installation to configure its resources over 48 channels for a 3090 processor. This minimizes channel contention, improves system performance and availability, and provides an installation with room for increased work loads.

With extended channel support, three-digit real device addresses that appear in certain commands, messages, and macros (in the form cuu) are extended to four-digit addresses (ccuu).

To support the extended channels, the DASD Dump Restore (DDR) program is changed to accommodate three- or four-digit device addresses. The DDR control statements with extended device address fields are INPUT, OUTPUT, and SYSPRINT.

Two directory control statements are changed to specify three- or four-digit addresses. Both the 'ccuu' parameter of the DIRECTORY statement and the rdev parameter of the DEDICATE statement are supported.

FORMAT/ALLOCATE control statements now support extended channels by including three- or four-digit device addresses in the messages they issue.

The VM MONITOR now maintains the larger number of values produced by extended channel support. It keeps a "channel busy" count and tracks the number of I/O tasks queued on channels 16 − 31 of both processors.

The device address that the loader can manipulate is extended from X'0000' to X'1FFF'. This allows the printer or reader to be on any channel.

Number of Virtual Devices Supported

The number of virtual devices supported for an ECMODE virtual machine ranges from one to a maximum of 3277 devices.

The option MAXDEV xxxx (used with the OPTION directory control statement) allows an ECMODE virtual machine to attach the number of devices specified by MAXDEV xxxx up to 3277. This maximum depends on VDEVSIZE which is 10 doublewords. To be able to attach this maximum number, you must have the maximum device option specified on the option statement in your virtual machine's directory. In addition, there must be sufficient contiguous free storage available.

If you do not have the MAXDEV xxxx option in your directory, then you will be able to attach up to 410 devices to your virtual machine.

Enhanced MVS/SP Support

VM/SP HPO incorporates new functions that enhance the performance of an MVS/SP virtual machine operating system. These functions are:

- Preferred machine assist
- Control switch assist extensions to preferred machine assist
- 3033 Extension Feature enhancement to virtual machine assist
- Enhanced availability in the MVS/SP V = R environment
- Single processor mode operational enhancements.

MVS/SP Preferred Machine Assist

VM/SP HPO provides support for the preferred machine assist feature, a microcode processor enhancement that is standard on the 4381, 3090, and 308x processors and is separately orderable for the 3033. Preferred machine assist produces a new mode of operation for the MVS/SP V = R virtual machine. Preferred machine assist significantly reduces control program processing, allowing the MVS/SP *preferred guest* to achieve improved performance.

As a preferred guest, MVS/SP operates in supervisor state with direct control of its hardware resources and I/O operations. The preferred guest can also use extended storage (greater than 16MB), providing additional performance potential for storage-constrained MVS/SP systems.

Additionally, preferred machine assist removes restrictions for using single processor mode and dynamic transition to and from single processor mode. Users executing MVS/SP with preferred machine assist can use the single processor mode and dynamic transition functions of VM/SP HPO.

Control Switch Assist Extensions to Preferred Machine Assist

This support, available on 308x, 4381, and 3090 processors, allows the MVS/SP preferred guest to use IUCV, some DIAGNOSE instructions, and some service call instructions. It also reduces line time-out problems for such guests by letting CP reflect virtual I/O interruptions to the guest. You must install the control switch assist to obtain this support.

3033 Extension Feature Enhancement to Virtual Machine Assist

A combination of processor and microcode enhancements, the 3033 Extension Feature (#6850) takes advantage of the numerous programming enhancements implemented in MVS/SP Release 3.

When installed on the IBM 3033 processor, the 3033 Extension Feature Enhancement reduces CP overhead for dynamic address translation. This enhancement permits MVS/SP virtual machines (Version 1 Release 3 and subsequent releases) to use hardware assists in the 3033 Extension Feature to improve performance. VM/SP HPO supports this feature for 3033 and 3042 processors.

The 3033 Extension Feature enhancement to virtual machine assist improves the performance of the MVS/SP Release 3 virtual machine operating system by:

- reducing paging operations
- enhancing real-storage management functions
- increasing the performance of cross-memory services
- increasing the performance of page-fault processing.

MVS/SP Release 3 cross-memory services gain an improvement in performance on processors equipped with the 3033 Extension Feature. The extends the ability of programs to communicate among address spaces through enhanced data movement and program calling procedures.

When the dual-address-space (cross-memory) assist is active, the 3033 Extension Feature improves the performance of MVS/SP page-fault processing. VM/SP HPO support of the 3033 Extension Feature reduces the amount of time needed to handle page translation exception interrupts that occur the first time certain MVS/SP tasks reference storage acquired by a GETMAIN macro instruction.

Additionally, restrictions for using single processor mode or nondisruptive transition are eliminated. 3033 users executing MVS/SP Release 3 can use the single processor mode or nondisruptive transition functions of VM/SP HPO.

Availability in the MVS/SP V = R Environment

VM/SP HPO increases the availability of the MVS/SP V = R virtual machine by attempting to save the status of the MVS/SP virtual machine after the control program terminates with a software abend. When the control program is automatically restarted, the MVS/SP V = R virtual machine resumes execution.

When the MVS/SP virtual machine is operating with preferred machine assist active, additional recovery capability is provided. If the VM/SP HPO control program encounters a condition that would place it in a disabled wait state, the control program attempts to pass control to the MVS/SP preferred guest. If the attempt is successful, the MVS/SP preferred guest continues to operate in native state. To restore VM/SP HPO operations, the system operator must shut down MVS/SP and reinitialize (using IPL) the VM/SP HPO control program.

Single Processor Mode Operational Enhancements

Single processor mode allows your installation to restrict the VM/SP HPO control program to a single processor of an attached processor or multiprocessor system, leaving the other processor for the exclusive use of the MVS/SP virtual machine. The single processor mode enhancements in VM/SP HPO make it easy for you to switch between single processor mode and attached processor mode or multiprocessor mode. Given this flexibility, your installation can pass a second processor between the MVS/SP V=R virtual machine and the VM/SP HPO control program without disrupting the MVS/SP virtual machine.

The MVS/SP virtual machine is able to issue MP instructions (CONCS, SPX, STPX, STAP, and SIGP) before single processor mode is activated for the system. When the system operator varies the second non-IPL processor offline, it becomes available to the MVS/SP virtual machine. There is no need to reinitialize the MVS/SP virtual machine operating system.

Single processor mode users on the 3033 processor can now use a subset of the shadow table bypass assist functions. For other processors for certain releases of MVS/SP, the single processor user must also be the MVS/SP preferred guest to use this subset of the shadow table bypass assist functions.

The processor handles execution of STNSM and STOSM instructions, eliminating VM/SP HPO control program processing.

Vector Facility

Support is provided for the Vector Facility in System/370 mode configured to a 3090 processor. This support increases the vector section size from 128 on the 3090 Model E to 256 on the 3090 Model S. The Vector Facility is a synchronous vector/scalar instruction processor that can manipulate values (usually floating-point) at a high speed. Compiled engineering and scientific FORTRAN applications can use the array processing capability of the Vector Facility. VM/SP HPO supports multiple virtual machines' use of this facility.

The VM MONITOR collects CP overhead time and the number of saves and restores for each processor. In addition, the VM Monitor collects data concerning the number of saves and restores and timer values associated with each virtual machine's use of the Vector Facility.

VM/SP HPO also allows many virtual machines to use the Vector Facility. Commands are provided to enable a Vector Facility user to display and change the various sets of registers within the Vector Facility and to determine the availability of the facility and the amount of time spent using the facility.

Summary of Changes

Release 6 Technical Changes for VM/SP HPO Release 6 Guide

Changes to this release are indicated by revision bars. For an example, look to the left of this paragraph.

Summary of Changes
for SC23-0189-4
as updated June 1989
for VM/SP HPO Release 6

Shared File System

Changed: Programming Support

The Shared File System (SFS) helps CMS to manage and share files. Files stored in SFS can be shared by multiple users and across multiple systems. SFS allocates DASD space dynamically as a user creates and extends files; it similarly reclaims the space as users erase or replace files. Users can organize their files in multiple hierarchical structured directories; hence, the same files can be in multiple directories. The addition of SFS has required only minor changes to this manual. For more details on SFS, refer to *VM/SP CMS Shared File System Administration*, SC24-5367.

Enhanced Saved Segment Management

Changed: Programming Support

Improvements to the handling of saved segments allow multiple logical saved segments to be packed together and managed in a single physical saved segment. Multiple objects, such as, NLS language repositories, may reside in a single logical segment.

National Languages Supported by VM/SP HPO

Changed: Programming Support

National Language support has been expanded. In addition to German, French and Kanji, VM/SP HPO now supports Spanish. Uppercase English is now included in the product tape. REXX appears in Brazilian Portuguese.

The NLS language repositories may now reside in a single logical segment because of improvements to saved segment Management.

3990 Storage Control Models 1, 2 and 3

New: Hardware Support

VM/SP HPO now supports the 3990 Storage Controls, Models 1, 2 and 3. The 3990 Model 1 is a new storage control providing two internal paths, while the 3990 Model

2 provides four storage paths. On top of the Model 2 capabilities, the 3990 Model 3 also provides:

- A choice of cache sizes (32, 64, 128, or 256 megabytes)
- A higher data transfer rate
- Improved internal cache algorithms
- A new fast write capability for guest virtual machines
- A 4 megabyte nonvolatile storage (NVS) providing random-access storage with its own separate power region for data protection.

A new control level mechanism is used to control channel command execution for DASD attached to the 3880 Models 11, 13, 21, and 23, and the 3990 Models 1, 2, and 3.

Two new directory statements, DASDOPT and MINIOPT, are added as part of this support. (DASDOPT establishes control level authorization and MINIOPT controls caching for minidisks.)

3380 Direct Access Storage Direct Channel Attach Model CJ2

New: Hardware Support

VM/SP HPO now supports the 3380 Direct Access Storage Direct Channel Attach Model CJ2. It provides both the storage functions of a 3380 DASD and the storage control functions of a 3990 Model 1 (the two storage path model) in a single unit.

Mixed Directory Support for VM/SP HPO and VM/XA SP Systems

Changed: Programming Support

Mixed directory support lets you use a single source user directory on both VM/SP HPO and VM/XA SP systems. Rather than flagging VM/XA SP-unique statements and options as errors, the directory program checks them for syntax errors. Otherwise, the VM/XA SP-unique statements and options will be ignored and will have no effect on the VM/SP HPO object directory.

DASD/Dump Restore Streaming Support Improvements

Changed: Programming Support

DDR DUMP and RESTORE I/O transfer rates for FBA DASD and streaming tapes are improved because the internal DDR buffer is larger than the buffer previously used. Since a larger amount of data can be transferred per SIO, DDR's internal headers for FBA DASD are written to tape differently.

Support for IBM 3380 Direct Access Storage Models AJ4/BJ4 and AK4/BK4

New: Hardware Support

VM/SP HPO now supports IBM 3380 Direct Access Storage Models AJ4/BJ4 and AK4/BK4. In addition, the FORMAT/ALLOCATE program (DMKFMT) has been modified to allocate the 2655 cylinders of the 3380 AK4/BK4. The maximum size of the DASD allocation map has been increased to 4K (4096 bytes).

The FINDSYS and ADDRQRY functions have been added to the CP DIAGNOSE code X'64'—Finding, loading, and purging a named segment or querying an address—to provide information about shared pages.

Support for VM/Directory Maintenance (DIRMAINT)

New and Changed: Programming Support

This support allows a VM/SP HPO system to exist in a multiple system environment that consists of VM/SP HPO and VM/XA SP systems. DIAGNOSE code X'84' processing will now accept and ignore any VM/XA SP-unique operations and parameters. This support improves performance because:

- DIRMAINT does not need to process the entire source directory each time it makes an update, and
- I/O contention to the minidisk that contains the real object directory is reduced.

Security and Integrity Enhancements

Changed: Programming Support

The Security and Integrity Enhancements represent four enhancements made to VM/SP HPO. The first three of these enhancements help you increase the security of your system and the fourth increases the integrity of your system.

Free Storage Enhancements

Changed: Free Storage Management

This programming enhancement changes the way CP manages free storage. This support:

- Provides recovery from many free storage errors that cause abends
- Reduces the overhead associated with free storage management
- Reduces the probability that SEL007 abends will occur
- Provides an external control for swap table migration.

Paging Storage and Vector Facility Changes

Changed: Programming support

This enhancement increases the following:

- Paging Storage increment size on the 3090 Models E and S
- Vector section size on the 3090 Model S.

The Paging Storage support also changes the way you can use the SYSXSTOR macro. You can now specify the exact number of megabytes you want instead of the entire increment for your particular processor.

Miscellaneous

Changed: Programming Support

The FINDSYS and ADDRQRY functions have been added to the CP DIAGNOSE code X'64'—Finding, loading, and purging a named segment or querying an address—to provide information about shared pages.

New: CP Command

GIVE command - supports the transfer of a dedicated tape drive from one user to another

New: DIAGNOSE Codes

X'DC' - starts and stops interval recording of data to the Monitor data stream
X'EC' - supports the CPTRAP and general use QUERY functions

Changed: DIAGNOSE Codes

X'B0' - accesses diagnostic IPL information.
X'64' - loads, finds, or purges a named system or queries an address.

Documentation Changes

Minor technical and editorial changes have been made throughout this publication.

Summary of Changes
for SC23-0189-3
as updated July 1987
for VM/SP HPO Release 5

IMPROVED STORAGE MANAGEMENT AND SIMPLIFIED TUNING

Changed: Programming Support

The new support that selects pages on a systemwide "least recently used" basis aims at improving performance in several ways:

- Improving the memory management of large working sets, shared pages, and the < 16 Mb area. Core table scan becomes the primary method for free list replenishment. In addition, the disposable page collector is eliminated.
- Streamlining the QDROP and QADD processes. When a virtual machine drops from queue, its pages will no longer be logically swapped and trimmed.
- Making Expanded Storage (called *Paging Storage* in this manual) more attractive as a swapping device.

ABILITY TO CREATE UP TO 9900 SPOOL FILES PER USER

Changed: Programming Support

The former limit of 9900 spool files on a system has been removed. With this change, there may exist up to 9900 spool files *for each user.*

Spool files will now have a user-unique spool ID. Reader spool file blocks (SFBLOKs) will now be kept in the virtual storage of a special userid, SYSSPOOL. Printer and punch SFBLOKs will remain in FREE storage.

In conjunction with these enhancements, the checkpoint/force start process has been improved to reconstruct the spool files more rapidly.

SCHEDULER ENHANCEMENT

Changed: Programming Support

With this enhancement, virtual machines are moved from the eligible list to the run list only if processor time and sufficient main storage are available. This will help eliminate storage overcommitment and reduce response times.

NEW 'NOQ2' OPTION ON THE 'SET QDROP OFF' COMMAND

New: Programming Support

Specify the SET QDROP OFF NOQ2 command for service virtual machines (like GCS/VSCS) that use system resources in small, frequent bursts. This command will keep that virtual machine in Q1, improving performance for the users of that service.

EXPANDED STORAGE (PAGING STORAGE) ENHANCEMENTS

New: Programming Support

A new macro, SYSXSTOR, is added. This macro controls the allocation of Paging Storage. You may continue to use SYSPAG to allocate Paging Storage, but SYSXSTOR offers these advantages:

- It is easier to use than SYSPAG
- For the 3090 Model 400, it allows you to generate Paging Storage greater than 64 increments.

IMPROVED PAGING ALLOCATION ORDER

New: Programming Support

A new parameter is added to the SYSPAG macro, ORDER=SYSTEM/USER. If ORDER=SYSTEM (the default) is coded, HPO will automatically order the devices allocated on the SYSPAG macro to distribute I/O activity over the available channels and control units. This will decrease the I/O contention by evenly distributing the I/O activity over the available I/O paths. If you specify ORDER=USER, you will preserve the order you specified.

IMPROVED SYSTEM LOCK UTILIZATION

Changed: Programming Support

There is less usage of the system lock in Release 5 and, therefore, less contention. This should improve system performance.

LESS THAN 16 MEGABYTE CONSTRAINT RELIEF

Changed: Programming Support

CP will now reference certain pages above the 16 megabyte line. This relieves constraint below the 16 megabyte line since CP no longer needs to copy these pages below 16 Mb to reference them.

ERROR RECORDING ENHANCEMENTS

New: Programming Support

For a 3090 Processor, the channel check handler and machine check handler will now provide more information in the error records. Specifically, these records will now tell you whether or not the hardware error actually affected system performance.

NEW 'NOVF' PARAMETER ON THE 'OPTION' DIRECTORY ENTRY

New: Programming Support

Specifying the NOVF parameter on a user's OPTION control statement in the directory will deny that user access to the Vector Facility.

4381 PROCESSOR COMPLEX MODELS 11, 12, 13, AND 14

Changed: Hardware Support

The 4381 Processor Complex Models 1, 2, and 3 are replaced and extended by the Models 11, 12, 13, and 14.

NEW MODELS OF THE 3090 PROCESSOR COMPLEX

Changed: Hardware Support

In addition to supporting the 3090 Processor Complex Model 200, VM/SP HPO now supports the 3090 Processor Complex Models 150, 150E, 180, 180E, 200E, 400, and 400E (the 400 and 400E are supported in partitioned processing mode only). VM/SP HPO does **not** support the 300E and 600E.

LOGICAL DEVICE HOST LIMIT RELIEF

Changed: Programming Support

This support removes the restriction that no more than 8 virtual machines can create and use logical devices concurrently. Now, any number of virtual machines can create up to 512 logical devices as long as the number of logical devices in the system does not exceed 4096.

3480 VOLUME SERIAL ERROR RECORDING

Changed: Programming Support

Users can now keep track of the error frequency for 3480 tapes by examining the volume serial in 3480 outboard records (OBR) or miscellaneous data records (MDR).

3422 MAGNETIC TAPE SUBSYSTEM

Changed: Programming Support

VM/SP HPO provides programming support for the 3422 Magnetic Tape Subsystem.

TRANSPARENT SERVICES ACCESS FACILITY (TSAF)

New: Programming Support

The transparent services access facility lets users connect to and communicate with local or remote virtual machines within a group of systems. This facility consists of the TSAF virtual machine component, APPC/VM, and two CP system services. APPC/VM is a modified subset of IUCV. With the TSAF virtual machine, it provides services within a single system and throughout a group of systems, unlike IUCV, which provides services only within a single system. The TSAF virtual machine component handles communication between systems by letting APPC/VM paths span more than one system.

QUERY CPTRAP Command

New: Programming Support

A new command, QUERY CPTRAP, is added.

NATIONAL LANGUAGE SUPPORT

New: Programming Support

VM/SP HPO now supports a variety of national languages. Updates have been made to modules and data areas providing this support, specifically, those handling CP messages.

ALTERNATE NUCLEUS SUPPORT

New: Programming Support

Alternate nucleus support makes it easier to create and IPL backup copies of the CP nucleus when the primary nucleus is damaged or unavailable.

PRINTER SUPPORT ENHANCEMENTS

Changed: Programming Support

The printer support enhancements include the addition of a SPOOL system service facility which provides support for a printer subsystem. The DESTination option allows you to select a specific printer or punch to process your print, punch, or console file. Two new DIAGNOSE codes allow a user to specify additional information about a print file. The CMS PRINT command has been enhanced to support an OVersize option and a special carriage control character to allow a longer data line.

LOGON/LOGOFF ENHANCEMENTS

Changed: Programming Support

The LOGON/LOGOFF enhancements improve system availability to users and resolve the problem of conflicting messages during LOGOFF processing.

ERROR LOGGING SYSTEM SERVICE

Changed: Programming Support

The error logging system service, a new CP system service, allows a virtual machine to receive a copy of all records currently written to the CP error recording area.

SPOOL FILE COMPRESSION SUPPORT ENHANCEMENT

Changed: Programming Support

An enhancement to SPOOL File Compression Support improves the reliability of transmitting spooled data between systems.

ASCII ENHANCEMENTS

Changed: Programming Support

Various enhancements have been made to the support of ASCII devices.

DOCUMENTATION CHANGES

Minor editorial and technical changes have been made throughout this publication.

Summary of Changes
for SC23-0189-2
as updated December 17, 1985
for VM/SP HPO Release 4.2

VECTOR FACILITY

New: Hardware Support

Support is provided for the Vector Facility in System/370 mode configured to a 3090 processor. The Vector Facility is a synchronous vector/scalar instruction processor that can manipulate values (usually floating-point) at a high speed. Compiled engineering and scientific FORTRAN applications can use the array processing capability of the Vector Facility. VM/HPO supports multiple virtual machines′ use of this facility.

VM/SP HPO provides user commands to display and change the various sets of registers in the Vector Facility. Commands are changed to provide monitoring and control over Vector Facility functions.

PAGE MIGRATION

Changed: Programming support

Page migration is changed to select pages (rather than segments) for migration on a reference basis instead of by time-stamp (age basis). Also, pages are migrated down the demand page hierarchy, instead of being migrated directly to the pre-allocated migration area. This improves the time required to retrieve those pages that become active in the near future.

Because migration of swap tables is sometimes necessary even when page migration is not actively moving pages, swap table migration is now invoked independently of page migration (rather than after page migration). Swap table migration is further improved by migrating swap tables regardless of whether all the pages in the segment have been migrated.

Because of this change, the installation should retune its free storage requirement.

Commands and monitor are enhanced to support the changed migration algorithms.

AUTO-DEACTIVATION OF RESTRICTED PASSWORDS AND DIRECTORY ENHANCEMENTS

New: Programming Support

This support enhances system integrity by minimizing the exposure of unauthorized system access through the use of restricted passwords. The directory enhancements remove the restriction on the number of USER entries that can be defined in the directory. Also, directory PROFILE support provides a means by which installations can optimize the number of commonly repeated control statements in USER entries in the source directory.

SECURITY ENHANCEMENT

New: Programming Support

The Resource Access Control Facility (RACF) is now called to authorize the STCP and LINK commands. If the user is not authorized to issue the STCP command, an error message is generated. Otherwise, the command is completed and processing continues as normal. If an unauthorized user tries to link to a disk using the LINK command, RACF denies access, the request fails, and an error message is issued. If your installation does not have the RACF feature, normal authorization checking is done.

If the JOURNAL operand of the SYSJRL macro is YES, you can set the maximum number of invalid password attempts and the delay time until next logon, or accept the default of 10 attempts and 60 minutes.

Accounting cards are generated for all CP directory links as well as other links currently recorded

ACCESS VERIFICATION ROUTINES

New: Programming Support

While VM/SP HPO provides many security functions, added support for access verification routines provides a standard interface to the RACF/VM Support PRPQ or user-written routines that can provide a higher level of security. Although the access verification routines support does not by itself provide security functions, it allows you to install software that does.

For example, to increase security of minidisk accesses, logon passwords, and movement of spool files, you can install access verification routines with the Resource Access Control Facility (RACF) (Program Number 5740-XXH) and RACF/VM Support PRPQ (Program Number 5767-002).

IX/370 HANDSHAKING SUPPORT

New: Programming Support

Support is added to improve the performance of authorized IX/370 virtual machines by accelerating supervisor call instructions (SVC) 9, 10, and 11.

3380 DIRECT ACCESS STORAGE DEVICE MODELS AE4/BE4

New: Hardware support

VM/SP HPO now supports the 3380 DASD Models AE4/BE4. The 3380 Models AE4/BE4 are count-key-data (CKD) devices that attach to high-speed channels only, via the 3880 Control Unit. The 3880 can attach up to 16 physical spindles (32 logical devices) of 3380 Models AE4 and BE4 directly to data streaming channels. Strings of different 3380 device models may be intermixed at the control unit level.

MONITOR DEVICE SUPPORT

New: Software support

You can now monitor as many DASD and tape devices as you can put on your system.

Glossary

This glossary defines new terms related to VM/SP HPO. This glossary is especially oriented for readers of *VM/SP HPO Release 6 Guide*. Therefore, some terms already defined in the *VM/SP HPO Glossary* do not appear here or may be defined slightly differently. Another glossary you may want to reference is the *IBM Dictionary of Computing.*

A

A disk. In CMS, the primary user disk that is allocated to a CMS user. This read/write disk is used to store files created under CMS; such files are retained until deleted by the user. Synonymous with *primary user disk*. See also *D disk, CMS system disk, virtual disk, S disk*, and *Y disk*, which have special uses; all other disks (B,C, E-R, T-X, and Z) are optional user disks.

abend. (1) Abnormal end of task. (2) Synonym for *abnormal termination.*

abend dump. The contents of main storage, or of part of main storage, written to an external medium for the purpose of debugging an error condition that resulted in the termination of a task prior to its normal completion.

abnormal end of task (abend). Termination of a task before its completion because of an error condition that cannot be resolved by recovery facilities while the task is executing.

abnormal termination. The ending of processing before planned termination. Synonymous with *abend.*

absolute address. The address assigned to a main storage location. An absolute address is used for a storage access without any transformations performed on it.

accept. Allowing a connection to the user's virtual machine from another virtual machine or from the user's own virtual machine.

ACF/SSP. Advanced Communications Function for Systems Support Programs.

ACF/VTAM. Advanced Communications Function for Virtual Telecommunications Access Method.

active disk table (ADT). A table residing in the user's copy of the CMS nucleus that contains an entry for each valid file mode letter; that is, for each disk that has been accessed.

active file table (AFT). A table residing in the user's copy of the CMS nucleus that contains an entry for each disk file that is currently open.

active work unit. A work unit that has uncommitted work associated with it. A request was made on the work unit (other than an atomic request) and no commit or rollback has occurred.

address translation. In VM/SP HPO, the process of changing the address of an item of data or an instruction from its virtual storage address to its real storage address. See also *dynamic address translation.*

ADT. Active disk table.

Advanced Communications Function for Systems Support Programs (ACF/SSP). An IBM program product made up of a collection of utilities and small programs. SSP is required for operation of the NCP.

Advanced Communications Function for Virtual Telecommunications Access Method (ACF/VTAM). An IBM licensed program that controls communications and flow of data in an SNA network. It provides single-domain, multiple-domain, and interconnected network capability.

Advanced Program-to-Program Communications (APPC). The inter-program communication service within SNA LU 6.2 on which the APPC/VM interface is based.

Advanced Program-to-Program Communication/VM (APPC/VM). An application program interface (API) for communicating between two virtual machines that is mappable to the SNA LU 6.2 APPC interface and is based on IUCV functions. Along with the TSAF virtual machine, APPC/VM provides this communication within a single system and throughout a collection of systems.

AF. Access Facility.

affinity. The tendency, either specified or implied, of a virtual machine always to be dispatched on the same processor in a dyadic, dual, or multiprocessor system.

AFT. Active file table.

alias. A pointer to a base file. An alias can be in the same directory as the base file or in a different directory. There must always be a base file for the alias to point to. The alias references the same data as the base file. Data is not moved or duplicated.

alphameric. Synonym for *alphanumeric.*

alphanumeric. Pertaining to a character set that contains letters, digits, and usually other characters, such as punctuation marks. Synonymous with *alphameric*.

American National Standard Code for Information Interchange (ASCII). The standard code, using a coded character set consisting of 7-bit coded characters (8 bits including parity check), used for information interchange among data processing systems, data communication systems, and associated equipment. The ASCII set consists of control characters and graphic characters.

AP. Attached processor.

APAR. Authorized program analysis report.

APAR number. The number that IBM assigns to an APAR and to the change resulting from it.

APPC. Advanced Program-to-Program Communications.

APPC/VM. Advanced Program-to-Program Communications/VM.

APPC/VM VTAM Support (AVS). A component of VM/SP that lets application programs using APPC/VM communicate with programs anywhere in a network defined by IBM's SNA. AVS transforms APPC/VM into APPC/VTAM protocol.

application program. A program written for or by a user that applies to the user's work, such as a program that does inventory control or payroll.

apply. In reference to installation and service, to load down program temporary fix (PTF) files from the tape, reassemble or rename as needed, and produce runnable (executable) code. The PTF may have been loaded down in a previous step. If that is the case, apply means to reassemble or rename if needed and place the files on the right staging disk for the build step to use, then produce the runnable code.

AP/MP mode. A mode of VM/SP HPO used when running in an attached processor or multiprocessor system. VM/SP HPO controls the execution of virtual machines on both processors.

area. This term is acceptable for storage space when there is no need to differentiate between DASD space on count-key-data devices and FB-512 devices. See also *DASD space.*

ASCII. American National Standard Code for Information Interchange.

assembler language. A source language that includes symbolic machine language statements in which there is a one-to-one correspondence with instruction formats and data formats of the computer.

attached processor (AP). A processor that has no I/O capability. An attached processor is always linked to the processor initialized for I/O handling. Note that VM/SP High Performance Option also supports true multiprocessor (MP) configurations.

attention interrupt. An input/output interrupt caused by a terminal user's pressing the attention key (or equivalent). See also *signaling attention, attention key.*

attention key (ATTN key). A key on some terminals that, when pressed, causes an I/O interrupt in the main processing unit. Also referred to as the ATTN key. See also *signaling attention.*

authority. In SFS, the permission to access a file or directory. You can have read authority or write authority (which includes read authority). You can also have file pool administration authority, which is the highest level of authority in a file pool.

authorized program analysis report (APAR). An official request to the responsible IBM Change Team to look into a suspected problem with IBM code. An APAR describes a problem, giving conditions of failure, error messages, abend codes, or other identifiers. It also contains a problem summary and resolution when applicable. See also *program temporary fix (PTF).*

auxiliary directory. In CMS, an extension of the CMS file directory, which contains the names and locations of certain CMS modules that are not included in the CMS file directory.

auxiliary file. In CMS, a file that contains a list of file types of update files to be applied to a particular source file. See also *control file, preferred auxiliary file.*

auxiliary storage. Data storage other than main storage; in VM/SP HPO, auxiliary storage may be a DASD or Paging Storage.

AVS. APPC/VM VTAM Support.

AVS virtual machine. The virtual machine that manages a gateway that allows communication among VM systems and an SNA network.

B

base file. The first occurrence of a file. It remains the base for the life of the file, even if the file has been renamed. Aliases point to base files.

basic control (BC) mode. A mode which the features of System/360 computing system and additional System/370 features, such as new machine instructions, are operational on a System/370 computing system. See also *extended control (EC) mode.*

basic conversation. A conversation where data is sent in APPC-defined logical record formats. Programs must be coded to consider the amount of data sent. Contrast with *mapped conversation.*

batch facility. See *CMS Batch Facility.*

binary digit. Either of the digits 0 or 1 when used in the pure binary numeration system. Synonymous with *bit.*

bit. (1) Either of the binary digits 0 or 1. See *byte.* (2) Synonym for *binary digit.*

block. A unit of DASD space on FB-512 devices. For example, FB-512 devices can be the IBM 9335, 9332, 9313, 3370, and 3310 DASD using fixed-block architecture. Contrast with *cylinder.*

bpi. Bits per inch.

BPI. Bytes per inch.

buffer. An area of storage, temporarily reserved for performing input or output, into which data is read, or from which data is written.

build. In reference to installation and service, to perform the steps necessary to take derived files and produce runnable code or systems. This is often referred to as the build process.

build minidisk/directory. A minidisk or SFS directory where executable versions of objects serviced are built.

built-in function. A specialized function, invoked by a keyword, that has been built into the system program code because it is commonly required by many users.

byte. A unit of storage, consisting of eight adjacent binary digits that are operated on as a unit and constitute the smallest addressable unit in the system.

C

#CP. Synonym for *escape to CP.*

cache. (1) In a processing unit, a high-speed buffer that is continually updated to contain recently accessed contents of main storage. Its purpose is to reduce access time. (2) In the 3880 Storage Control Models 11, 13, 21, and 23, a high-performance, random-access, electronic storage device. The cache is one level of the two-level storage arrangement of the 3880 Models 11, 13, 21, and 23; the second level is the DASDs. (3) In the 3990 Storage Control Model 3, basic caching operations are equivalent to those provided by the 3880 Storage Control Models 13 and 23. In addition, cache fast write capability is provided for guest virtual machines. See also *cache fast write.*

cache directory. A directory in a cached device that contains entries that identify the address of the DASD track combined with the address of associated DASD as seen by the storage director.

cache fast write. A caching function provided with the 3990 Storage Control Model 3 for guests. It permits write operations to be performed at cache access speed, eliminating the need to write data to DASD immediately. Cache fast write is designed for use with special kinds of data, such as sort work files, that can easily be reconstructed if lost.

cache-line. A block of bytes of central storage that maps to a specific area in the cache or high-speed-buffer. See also *high-speed-buffer.*

caching function. The function that allows the cache of a cache device to be used.

callable services library (CSL). A package of CMS assembler routines that can be stored as an entity and made available to application programs.

catalog storage group. The storage group in a file pool that contains information about the objects (such as files and directories) and authorizations that exist in the file pool. See *file pool catalog.*

CCW. Channel command word.

CE. (1) Channel end. (2) Customer Engineer.

changes. In reference to installation and service, IBM and original equipment manufacturer (OEM) supplied service for their programs (PTFs, APARs, etc.) and user modifications to those programs.

channel. A path in a system that connects a processor and main storage with an I/O device.

channel command word (CCW). A doubleword at the location in main storage specified by the channel address word. One or more CCWs make up the channel program that directs data channel operations.

channel status word (CSW). An area in storage that provides information about the termination of I/O.

channel-check handler (CCH). In System/370, a feature that records information about channel errors and issues appropriate messages to the operator.

checkpoint. (1) An SFS internal file pool server operation during which the changes recorded on the log minidisks are permanently made to the file pool. (2) A spool file operation that saves the status of a spool file for a checkpoint start.

checkpoint (CKPT) start. A system restart that attempts to recover information about closed spool files previously stored on the checkpoint cylinders. The spool file chains are reconstructed, but the original sequence of spool files is lost. Unlike warm start, CP accounting and system message information is also lost. Contrast with *cold start, force start,* and *warm start.*

circumventive service. Information that IBM supplies over the phone or on a tape to circumvent a problem by disabling a failing function until a PTF is available to be shipped as a corrective service fix. See *patch* and *zap.*

CKD. Count-key-data.

class. The IBM-defined privilege class when used to identify a command or DIAGNOSE code as used in the phrase "the class A command QUERY."

class Any. A subset of CP commands used to log on to VM/SP HPO, have the terminal logically connected to a multiple-access virtual machine, or send messages to the operator or another user.

class override file. A file containing control statements defining changes in the privilege classes of CP commands and/or diagnose codes. The file is used by the override program, DMKOVR, to establish a new class structure of commands under user class restructure (UCR).

CMS. Conversational Monitor System.

CMS batch facility. A facility that allows a user to run time-consuming or noninteractive CMS jobs in another CMS virtual machine dedicated to that purpose, thus freeing this user's terminal and virtual machine for other work.

CMS/DOS. Refers to the functions of CMS that become available when you issue the command SET DOS ON. CMS/DOS is a part of the normal CMS system and is not a separate system. Users who do not use CMS/DOS are sometimes referred to as OS users, since they use the OS simulation functions of CMS.

CMS EXEC. An EXEC procedure or EDIT macro written in the CMS EXEC language and processed by the CMS EXEC processor. EXECs may be written in EXEC2 or REXX (System Product Interpreter) language. Synonymous with *CMS program.* See also *EXEC.*

CMS EXEC language. A general purpose, high-level programming language, particularly suitable for EXEC procedures and edit macros. (The language is documented in the *VM/SP CMS Command and Macro Reference*, SC24-6209.) Procedures and macros (programs) written in this language are executed by the CMS EXEC processor.

CMS file directory. A directory on each CMS disk that contains the name, format, size, and location of each of the CMS files on that disk. When a disk is accessed using the ACCESS command, its directory is read into virtual storage and identified with any letter from A through Z. Synonymous with *master file directory block.*

CMS files. This term refers exclusively to files that are in the fixed-block format used by CMS file system commands. VSAM and OS data sets and DOS files are not compatible with the CMS file format and cannot be manipulated using CMS file system commands. Note that "disk", "minidisk", and "virtual disk" are used interchangeably to indicate disks that are in your CMS virtual machine configuration. Where necessary, a distinction is made between CMS-formatted disks and disks that are in OS or DOS format.

CMS file system. A way to create files in the CMS system. CMS files are created by using an identifier consisting of three fields: file name, file type, and file mode. These files are unique to the CMS system and cannot be read or written using other operating systems.

CMS nucleus. The portion of CMS that is resident in the user's virtual storage whenever CMS is executing. Each CMS user receives a copy of the CMS nucleus when he initial program loads (IPLs) CMS. See also *saved system, shared segment.*

CMS program. Synonym for *CMS EXEC.*

CMS system disk. The virtual disk (file mode S) that contains the CMS nucleus and the disk-resident CMS commands. The CMS system disk can have extensions; usually as file mode Y.

CMSSERV. The command that starts a CMS router in the Enhanced Connectivity Facilities environment of VM.

cold start. A system restart that ignores previous data areas and accounting information in main storage, and the contents of paging and spool files on CP-owned disks. Contrast with *warm start, checkpoint start, force start.*

collection. A group of up to 8 VM operating systems that can share resources. Each system within the collection must have the transparent services access facilities (TSAF) virtual machine installed and running.

command. A request from a user at a terminal for the execution of a particular CP, CMS, RSCS, or IPCS function. A CMS command may also be the name of a CMS file with a filetype of EXEC or MODULE. See also *subcommand, user-written CMS command.*

command line. The line at the bottom of display panels that lets a user enter commands or panel selections. It is prefixed by an arrow (= = = >).

commit. Permanently changing a resource (such as a file or a data base object).

Common Programming Interface (CPI) Communications. A set of program-to-program communication routines that let applications written in REXX and other high-level languages access APPC/VM functions. These routines are part of IBM's Systems Application Architecture.

common storage. A shared segment of reentrant code that contains free storage space, the GCS supervisor, control blocks, and data that all members of a virtual machine group share.

communications directory. A CMS facility that lets APPC/VM applications connect to a resource using symbolic destination names and special NAMES files.

compile. To translate a program written in a high-level programming language into a machine language program.

component. A collection of elements that together form a separate functional unit. A product may contain many components (VM/SP HPO for example has components of CP, CMS, AVS, GCS, TSAF,and IPCS). A component may be part of many products (CMS spans both the VM/SP and VM/SP HPO products).

component override. Synonym for *component parameter override.*

component override area. An area of the product parameter file or of a product parameter override file that contains one or more component parameter overrides. Synonymous with *override area.*

component parameter override. A component parameter, defined in a component override area, that updates or replaces a component parameter defined in a component area of the product parameter file. Synonymous with *component override* and *override.*

computer word. A contiguous set of bits or characters suitable for processing by a given computer, usually treated as a unit.

concurrently. Pertaining to a mode of operation that includes the performance of two or more operations within a given interval of time.

connect. Establish a path to communicate with another virtual machine or with the user's own virtual machine.

console. A device used for communications between the operator or maintenance engineer and the computer.

console stack. A two-part storage area. Console stack is a general term used to refer collectively to the program stack and the terminal input buffer. The term console stack is used when there is no need to distinguish between the two areas. See also *program stack, terminal input buffer.*

contention. The situation in which two LUs try to allocate a conversation over the same session at the same time.

control block. A storage area that a computer program uses to hold control information.

control data. In reference to a file pool, the data that controls the DASD space and objects within a file pool. Control data consists of the POOLDEF file, the control minidisk, and all minidisks allocated to storage group 1.

control file. (1) In CMS, the file that contains records that identify the updates to be applied and the macro libraries, if any, that are needed to assemble that source program. See also *auxiliary file.* (2) A CMS file that is interpreted and used to direct the flow of a certain process through a series of specific steps. For example, the control file could contain installation steps, default addresses, and PTF prerequisite lists as well as many other items that are necessary.

control level authorization. Control level authorization allows the installation to affect operational control over use of CCWs by virtual machine programs. This control is used to manage subsystem or device-related functions and resources. It is valid for virtual devices defined on DASD attached to 3880 Models 11, 13, 21, and 23, or any model 3990.

control minidisk. In a file pool, the minidisk that tracks the physical DASD blocks allocated to the file pool.

control program. A computer program designed to schedule and to supervise the execution of programs in a computer system. See also *Control Program (CP).*

Control Program (CP). The component of VM/SP HPO that manages the resources of a single computer in such a way that multiple computing systems appear to exist. Each virtual machine is the functional equivalent of an operating system.

control statement. A statement that controls or affects the execution of a program in a data processing system.

control unit. A device that controls input/output operations at one or more devices.

control unit terminal (CUT). An operating mode that allows one logical terminal session. Contrast with *distributed functional terminal.*

conversation. A connection between two transaction programs over an LU-LU session that lets them communicate with each other while processing some transaction. The programs establish a conversation, send and receive data in the conversation, and then terminate the conversation.

copy file. A file having file type COPY that contains nonexecutable real storage definitions that are referred to by macros and assemble files.

Conversational Monitor System (CMS). A component of VM that is a conversational operating system. CMS provides general interactive time-sharing, problem solving, and program development capabilities. It operates only under the control of the VM/SP or VM/SP HPO Control Program (CP).

copy function. In VM, the function initiated by a Program Function (PF) key to copy the contents of a display screen onto an associated hard copy printer. A remote display terminal copies the entire contents of the screen onto a printer attached to the same control unit. A local display terminal copies all information from the screen, except the screen status information, onto any printer attached to any local display control unit.

corrective service. The process of loading changes from tape to minidisks and then applying the requested changes. The last step of corrective service is to do the build process (see *build*).

corrective service tape. A tape supplied by IBM at the user's request containing a fix for a specific problem.

count-key-data (CKD) device. A disk storage device that stores data formatted with a count field, usually followed by a key field, followed by the actual data of a record. The count field contains the cylinder number, head number, record number, and the length of the data. The key field contains the record's key (search argument).

CP. Control Program.

CP assist. A hardware function available only on a processor that has Extended Control-Program Support (ECPS), that reduces CP overhead by performing the most frequently used tasks of CP routines.

CP command. A request from the terminal user for the execution of programming that controls the user's virtual machine. The VM control program commands are called CP commands. The subset of CP commands that perform console simulation are called console functions.

CP directory. Synonym for *VM directory*.

CPI Communications. Common Programming Interface Communications.

CP read. The situation in which the control program (CP) is waiting for a response or request for work from the user. On a typewriter terminal, the keyboard is unlocked; on a display terminal, the screen status area indicates CP READ.

CP trace table. A table used for debugging VM; its size is a multiple of 4096 bytes and is dependent on the size of real storage. This table contains the chronological occurrences of events that take place in the real machine, recorded in a wraparound fashion within the trace table.

CPTRAP. A CP debugging tool that creates a reader spool file of selected trace table entries, CP data, and virtual machine data in the order that they happen. The IPCS commands can help the user access and print this collected data.

CPU. Central processing unit.

cross memory service. This MVS/SP™ facility is a standard hardware feature on the IBM 3081 processor complex. It increases communication efficiency between address spaces. To implement cross memory services on the IBM 3033 processor, the processor must have virtual machine assist with the 3033 extension feature (#6850) as well as the virtual machine assist RPQ. Synonymous with *dual address space assist*.

CSL. Callable services library.

CSW. Channel status word.

CUT. Control unit terminal.

cylinder. A term used to describe specific space on count-key-data direct access storage devices.

D

D disk. In CMS, the disk that becomes a user disk with a mode letter of D if the user logs on and a virtual disk at address 192 is defined in the virtual machine configuration.

DASD. Direct access storage device.

DASD Dump Restore (DDR) program. In VM, a service program used to copy all or part of a minidisk onto tape, or to load the contents of a tape onto a minidisk.

DASD space. (1) Area allocated to DASD units on count-key-data devices. (2) Area allocated to DASD units on FB-512 devices. Note that *DASD space* is synonymous with *cylinder* when there is no need to differentiate between count-key-data devices and FB-512 devices. This term applies to VM/370, as well as to the VM/SP and VM/SP HPO licensed programs.

MVS/SP is a trademark of the International Business Machines Corporation.

data area. A storage area used by a program or device to hold information.

data control block (DCB). A control block access method routines use to store and retrieve data.

data stream. A set of logical records sent one after the other.

DBCS. Double-byte character set.

DCB. Data control block.

DDR. See *DASD Dump Restore (DDR) program.*

DDR program. In VM, refers to the DASD Dump Restore program.

Device Level Selection (DLS) Support Mode. A feature of the 3990 Storage Control Models 1, 2, and 3 which permits two single-path storage directors to access data in the DASD subsystem. In DLS mode, there is a one-to-one relationship between the storage director and the storage path.

DFT. Distributed function terminal.

DIAGNOSE interface. Under VM, a programming mechanism that allows any virtual machine, including CMS, to communicate directly with CP through the DIAGNOSE instruction. Specific interface codes allow a virtual machine to request specific CP services efficiently.

direct access storage device (DASD). A storage device in which the access time is effectively independent of the location of the data.

directory. See *auxiliary directory, CMS file directory,* or *VM directory.*

directory identifier (dirid). A fully-qualified directory name (in which the file pool ID and userid can be allowed to default), a file mode letter, or plus (+) or minus (-) file mode syntax (used in commands).

directory name (dirname). A fully-qualified directory name that can incorporate a period (.) to indicate the user's own top directory (used in commands).

dirid. Directory identifier.

dirname. Directory name.

discontiguous system. One or more 64K segment(s) of storage that was previously changed and saved and assigned a unique name. The segment(s) can be shared among virtual machines. If the segment(s) contains reentrant code, it may be shared and CP will check for any modifications. If the segment(s) contains reentrant code and is protected, CP insures that the segment(s) cannot be loaded.

disk. Refers to a disk that is in your CMS virtual machine configuration. Also referred to as a virtual disk.

disk operating system (DOS). An operating system for computer systems that use disks and diskettes for auxiliary storage of programs and data.

Disk Operating System/Virtual Storage Extended (DOS/VSE). An operating system that is an extension of DOS/VS. A VSE system consists of:

1. Licensed VSE/Advanced Functions support, and
2. Any IBM-supplied and user-written programs required to meet the data processing needs of a user.

VSE and the hardware it controls form a complete computing system.

dispatch list. In VM, a list of those virtual machines that are executable and currently competing for a time slice of processor resources. The virtual machine currently executing is called the runuser. Dispatch lists are a subset of the run list. On an AP/MP system, there are two dispatch lists for each run list. See also *eligible list* and *run list.*

dispatcher. The program in CP that places jobs or tasks into execution. The dispatcher selects the next virtual machine to run and prepares the virtual machine for problem state execution.

dispatcher queues. All the queues that the dispatcher references (IOBLOK/TRQBLOK, CPEXBLOK, and dispatch list queues).

dispatching. The starting of virtual machine execution.

display mode. A type of editing at a display terminal in which an entire screen of data is displayed at once and in which the user can access data through commands or by using a cursor. Contrast with *line mode.*

display terminal. A terminal with a component that can display information on a viewing surface such as a CRT or gas panel.

Distributed Data Processing (DDP). The operation of a network of interconnected processors/systems each capable of independent operation. These processors/systems are linked so it is not necessary to store all data and programs at every site. Authorized users of such a network can use the facilities of any part of this network. Such use can either be intentional or transparent to the user; that is, the user might intentionally request service from a remote system or a user's request for data or a program might automatically be routed to the appropriate part of the network by that system.

distributed function terminal (DFT). An operational mode that allows multiple concurrent logical terminal sessions. Contrast with *control unit terminal (CUT)*.

Document Composition Facility (DCF). A text processing program; its main component is the text formatter, called SCRIPT/VS. See *SCRIPT/VS*.

DOS. Disk operating system.

DOS/VSE. Disk Operating System/Virtual Storage Extended.

double-byte character set (DBCS). A character set for languages, such as Kanji, that require two bytes to uniquely define each character.

dual. A processor complex comprising two processors in one unit. Both processors share central storage, are controlled by a single operating system, and communicate directly with each other. A dual configuration differs from a dyadic configuration because the channels in the dual configuration are attached directly to each processor and channel set switching is not provided. The 4381 Model 3 is an example of a dual processor.

dual address space assist. Synonym for *cross memory service*.

dual processor. A processor complex comprising two processors in one unit. Both processors share central storage, are controlled by a single operating system and communicate directly with each other. A dual configuration differs from a dyadic configuration because the channels in the dual configuration are attached directly to each processor and channel set switching is not provided. Contrast with *dyadic processor*.

dump (DU) area. An area of CP-owned DASDs as specified by the SYSPAG macro.

dyadic. A system having two processors that cannot be configured into two independent uniprocessors that use separate control programs. For example, the 3081 Processor Complex contains two processing units that share central storage.

dump. To write the contents of part or all of main storage, or part or all of a minidisk, to auxiliary storage or a printer. See *abend dump*.

dyadic processor (DY). A processor complex comprising two processors in one unit. Both processors share central storage, are controlled by a single operating system, communicate directly with each other, execute I/O operations through a common element, and can run with one central processor if the other is removed from the configuration because of an error. A dyadic processor cannot be configured into two independent uniprocessor units. Note that each processor has access to its own assigned channel set. Contrast with *dual processor*.

dynamic address translation (DAT). In System/370 virtual storage systems, the change of a virtual address to a real storage address during execution of an instruction.

dynamic paging area (DPA). The area of real storage that is used by CP for the temporary storage of pages when paging occurs.

E

EBCDIC. Extended binary-coded decimal interchange code.

ECB. Event Control Block.

EC level. Engineering change level.

EC mode. Extended control mode.

ECMODE. A mode in which all of the features of a System/370 computing system, including dynamic address translation, are operational.

ECPS:VM/370. See *Extended Control-Program Support*.

EDF. Enhanced disk format.

edit. To make changes, additions, or deletions to a file that is on a disk, and to make these changes interactively. The edit function is also used to generate information in a file that did not previously exist.

eligible list. In VM/SP HPO, a list of virtual machines that are potentially executable, but are not placed on the dispatch list to compete for processor resources because of the current load on the system. See also *dispatch list*.

emulation. The use of programming techniques and special machine features to permit a computing system to execute programs written for another system.

enhanced connectivity. The strategy for sharing services and resources in a heterogeneous network.

enhanced disk format (EDF). A CMS file storage format that supports files consisting of 512-, 1K-, 2K-, or 4K-byte CMS blocks.

entitled. Entitled manuals are those manuals in the VM/SP HPO library that are shipped automatically with the product tape. Any updates (new releases, technical newsletters, and revisions between releases) of entitled manuals are sent to customers automatically as soon as they are available from IBM.

entry point. An address or label of an instruction performed upon entering a computer program, a routine, or a subroutine. A program may have several different entry points, each corresponding to a different function or purpose.

Environmental Record Editing and Printing Program (EREP). A program that makes the data contained in the system recorder file available for further analysis.

EREP. Environmental Record Editing and Printing Program.

error recording area. This term refers to the DASD space that the system programmer defines during system generation on the system residence volume that CP uses to record formatted outboard error recordings, machine check records, and channel check records. For count-key-data devices, this area is between 2 and 9 contiguous cylinders in size; for FB-512 devices, the size of this area can be any number of contiguous pages.

escape to CP. Under VM, a transfer of control to CP when either the terminal user or the machine stops virtual machine operation. This can be accomplished by a CP command (such as #CP), by invoking a DIAGNOSE function, or by signaling attention. Synonymous with *#CP*. See also *DIAGNOSE interface, signaling attention, attention interrupt.*

event control block (ECB). A control block that represents the status of an event.

EXEC. An EXEC written using the System Product Interpreter, EXEC2, or CMS EXEC language. See also *CMS EXEC.*

EXEC 2 language. A general purpose, high-level programming language, particularly suitable for EXEC procedures and XEDIT macros. (The language is documented in the *VM/SP EXEC 2 Reference*, SC24-5219.) Procedures and XEDIT macros (programs) written in this language are executed by the EXEC 2 processor. Contrast with *CMS EXEC language* and *Restructured Extended Executor (REXX) language.*

Expanded Storage. An optional paging assist of the 3090 processor that provides additional storage for paging or swapping. See also *Paging Storage.*

expanded virtual machine assist. A hardware assist function, available only on a processor that has Extended Control-Program Support (ECPS), that handles many privileged instructions not handled by virtual machine assist, and extends the level of support of certain privileged instructions beyond that provided by virtual machine assist.

explicit lock. A lock on a file or directory that a user explicitly created by entering a CREATE LOCK command or executing a DMSCRLOC CSL routine.

extended binary-coded decimal interchange code (EBCDIC). A set of 256 characters, each represented by 8 bits.

extended channel support. Support for a one byte channel address. This allows a device address range of 0 through 1FFF.

extended communications. A set of program-to-program communication routines that are an extension to CPI-Communications routines. These routines *are not* part of IBM's Systems Application Architecture.

extended control (EC) mode. A mode in which all the features of a System/370 computing system, including dynamic address translation, are operational. See also basic control (BC) mode.

Extended Control-Program Support (ECPS:VM/370). A hardware assist feature available on certain processors, that improves the performance of CP by reducing CP overhead. ECPS:VM/370 consists of CP assist, expanded virtual machine assist, and virtual interval timer assist.

extended PLIST (untokenized parameter list). This is an untokenized parameter list. It consists of four addresses that indicate the extended form of a command as it was entered at a terminal.

extended storage. Storage above the 16 megabyte line.

external security manager. A program that either augments or completely replaces the authorization checking done by file pool server processing.

F

FB-512. Refers to the IBM 3370 and 3310 Direct Access Storage Device.

FBA. Fixed block architecture.

file access mode. A filemode number that designates whether the file can be used as a read-only or read/write file by a user. See also *filemode*. The following are the filemode numbers available to the VM/SP HPO user:

0 Limits access to a file to only those other users who have read/write access to the disk. Files having filemode 0 are not listed for another user who links to a disk in read-only mode and requests a list of files on the disk.

1 Allows general read/write use of the file; this is the default.

2 Allows general read/write use of the file. Filemode 2 is usually used to group together files on a common disk, such as the system disk.

3 Causes the file to be erased after it is read.

4 Causes the file to be written in OS simulated data set format.

5 Allows general read/write use of the file. Filemode 5 is used to group together files so they can be manipulated as a group.

6 Causes existing records of a file to be written back to their previous location on the disk rather than to a new location. Filemode 6 eliminates the need for the dual directory scheme of block location on a disk, and reduces the possibilities of errors when one user links to another user's virtual disk.

file pool. A collection of minidisks managed by SFS. It contains user files and directories and associated control information. Many users' files and directories can be contained in a single file pool.

file pool catalog. The part of a file pool that contains information about the objects stored in the file pool and the authorizations granted on those objects. See *catalog storage group.*

file pool ID. The name of a file pool. It is part of a fully-qualified directory name, identifying where the directory and all files in it are located. It has up to eight characters, followed by a colon (:).

file pool server machine. A virtual machine that is properly configured to manage a file pool. (Its VM system directory entries must, for example, contain the MDISK statements for a file pool.)

file space. A user's allocation of space within a file pool.

file system control block (FSCB). A collection of information about a CMS file, used by CMS OS simulation commands and user programs. A file system control block is established for a file by the FILEDEF command or the FSCB macro instruction.

fileid. A CMS file identifier that consists of a filename, filetype, and filemode. The fileid is associated with a particular file when the file is created, defined, or renamed under CMS. See also *filename, filetype, filemode.*

filemode. A 2-character CMS file identifier field comprising the filemode letter (A through Z) followed by the filemode number (0 through 6). The filemode letter indicates the CMS file directory on which the file resides and whether or not the disk is a user virtual disk or a CMS system disk. The filemode number indicates the access mode of the disk. See also *file access mode.*

filename. A 1- to 8-character alphameric field, comprised of A-Z, 0-9, and special characters $ # @ + - (hyphen) : (colon) _ (underscore), that is part of the CMS file identifier and serves to identify the file for the user.

filetype. A 1- to 8-character alphameric field, comprised of A-Z, 0-9, and special characters $ # @ + - (hyphen) : (colon) _ (underscore), that is used as a descriptor or as a qualifier of the filename field in the CMS file identifier. See also *reserved filetypes.*

fixed-block architecture (FBA). Those DASD devices whose architecture uses fixed blocks or records of 512 bytes.

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

FMH5. Function Management Header 5.

force start. A VM system restart that attempts to recover information about closed spool files that was previously stored on the checkpoint cylinders. All unreadable or invalid spool file information is ignored. Contrast with *checkpoint start, warm start, cold start.*

free list. A list maintained by CP that points to a set of pages above or below the 16 MB line that can be allocated to satisfy both virtual machine and system page requests. To satisfy system requests, pages below the 16 MB line must be used.

free storage. Storage not allocated. The blocks of memory available for temporary use by programs or by the system.

FSCB. File system control block.

fullword. Synonym for *computer word.*

Function Management Header 5 (FMH5). A field at the beginning of an application request that carries control information for the target LU in an SNA network.

G

gateway. The LU name of a TSAF collection that is a source for communications to an SNA-defined network or the target of communications from an SNA-defined network.

GCS. Group control system.

general register. In CMS, a register that does operations such as binary addition, subtraction, multiplication, and division. General registers primarily compute and modify integers and addresses in a program.

global resource. A resource accessible from anywhere within a TSAF collection and whose identity is known throughout the collection. A shared file system file pool is an example of a global resource. Contrast with *local resource* and *private resource.*

global resource manager. An application that runs in a server virtual machine and identifies itself to the TSAF collection as a global resource owner using *IDENT. Contrast with *local resource manager* and *private resource manager.*

group. Synonym for *virtual machine group.*

Group Control System (GCS). An operating environment that provides a problem state OS subtasking environment with common storage access for members of a virtual machine group.

guest virtual machine (GVM). A virtual machine in which an operating system is running.

H

HELP. An online tool for supplying reference information on commands and messages for VM components.

high-speed-buffer. A set of logically partitioned blocks that provides significantly faster access to instructions and data than central storage.

host system. A data processing system that prepares programs and the operating environments for use by another computer or controller.

HPO. See *IBM Virtual Machine/System Product High Performance Option.*

I

IBM Virtual Machine/System Product (VM/SP). A licensed program that manages the resources of a single computing system so that multiple computing systems (virtual machines) appear to exist. VM/SP consists of a Control Program (CP), a Conversational Monitor System (CMS), a Group Control System (GCS), an Interactive Problem Control System (IPCS), Advanced Program-to-Program Communications/Virtual Machine Virtual Telecommunications Method (AVS) support, and a Transparent Services Access Facility (TSAF). Note that former VM/370 users continue to have a Remote Spooling Communications Subsystem (RSCS), which spools files to and from remote work stations.

IBM Virtual Machine/System Product High Performance Option (VM/SP HPO). VM/SP HPO is a separately orderable licensed program that can be installed and executed in conjunction with VM/SP. When you install and use VM/SP HPO in conjunction with the prerequisite VM/SP release, you obtain an operating system that extends the capabilities of VM/SP. VM/SP HPO offers enhancements for large system environments. These enhancements include system management performance improvements, additional processor and I/O support, and enhanced MVS/SP support.

inactive work unit. A work unit on which no requests have yet been made, or an atomic request was made, or requests were made and have been committed or rolled back; that is, an inactive work unit has no uncommitted work associated with it.

indicator. A 1-byte area of storage that contains either the character "1" to denote a true condition or the character "0" to denote a false condition.

initial program load (IPL). The initialization procedure that causes an operating system to begin operation. A virtual machine user must IPL the specific operating system into the virtual machine that will be used to control his work. Each virtual machine may be loaded with a different operating system.

initialize. To set counters, switches, addresses, or contents of storage to starting values.

input/output (I/O). (1) Pertaining to a device whose parts can do an input process and an output process at the same time. (2) Pertaining to a functional unit or channel involved in an input process, output process, or both, concurrently or not, and to the data involved in such a process.

Inter-User Communication Vehicle (IUCV). A VM generalized CP interface that facilitates the transfer of messages either among virtual machines or between CP and a virtual machine.

interactive. (1) An application where each user entry calls forth a response from a system or program. (2) A user's virtual machine is classified as interactive if it is in queue 1 or it's first queue 2 time slice. See also *queue 1, queue 2.* Contrast with *noninteractive.*

Interactive Problem Control System (IPCS). A component of VM that permits online problem management, interactive problem diagnosis, online debugging for disk-related CP abend dumps, problem tracking, and problem reporting.

interface. A shared boundary between two or more entities. An interface might be a hardware or software component that links two devices or programs together.

internal trace table. See *CP trace table.*

interrupt. A suspension of a process, such as execution of a computer program, caused by an external event and done in such a way that the process can be resumed.

Inter-User Communication Vehicle (IUCV). A VM/SP or VM/SP HPO generalized CP interface that helps the transfer of messages either among virtual machines or between CP and a virtual machine.

invoke. To start a command, procedure, or program.

I/O. Input/output.

IPCS. Interactive Problem Control System.

IPL. Initial Program Load.

IPL processor. In an attached processor (AP) or multiprocessor (MP) system, the processor on which the control program was first initialized during system generation. Note that both the IPL and the non-IPL processors in a real MP configuration have I/O capabilities.

IUCV. Inter-User Communication Vehicle.

K

K. Kilobyte.

L

library object. A named resource stored in a host library system.

line mode. The mode of operation of a display terminal that is equivalent to using a typewriter-like terminal. When the CMS editor is used, the terminal displays a chronological log of the CMS EDIT subcommands entered, the lines affected by the editing (unless that is suppressed), and the system responses. When the System Product Editor (XEDIT) is used, full screen editing is the norm but line mode can be used instead. Contrast with *display mode*.

link. (1) In RSCS, a connection, or ability to communicate, between two adjacent nodes in a network. (2) In TSAF, the physical connection between two systems.

load. In reference to installation and service, to move files from tape to disk.

loader. A routine, commonly a computer program, that reads data into main storage.

local. Two entities (for example, a user and a server) are said to be local to each other if they belong to the same system within a collection or to the same node within a SNA system.

local resource. A resource accessible from only within a single VM system and whose identity is known only within a single VM system in the TSAF collection. Contrast with *global resource* and *private resource*.

local resource manager. An application that runs in a virtual machine and identifies itself to the local system in the TSAF collection as a local resource owner by *IDENT. Contrast with *global resource manager* and *private resource manager*.

local service. Changes manually applied to a product or component (that is, not using the program update service or corrective service procedures). See *circumventive service* and *user modification*.

lock. (1) A tool for controlling concurrent usage of SFS objects. Implicit locks are acquired and automatically released when you run CMS commands and program functions in SFS. Explicit locks let you control the type and duration of the lock. (2) In AP/MP mode, CP also has a variety of locks to limit access to system resources to a single processor at a time.

log data. Information that a communications program can send to its partner to help diagnose errors.

logical saved segment. A portion of a physical saved segment that CMS can manipulate. Each logical segment can contain different types of program objects, such as modules, text files, execs, callable services libraries, language repositories, user-defined objects, or a single minidisk directory. A system segment identification file (SYSTEM SEGID) associates a logical saved segment to the physical saved segment in which it resides. See *physical saved segment* and *saved segment*.

logical unit (LU). An entity addressable within an SNA-defined network, similar to a node within a VM network. LUs are categorized by the types of communication they support. A TSAF collection in an SNA network is viewed as one or more LUs.

log minidisks. In a file pool, two duplicate minidisks that contain information about changes made to the file pool. File pool servers use the log minidisks to help protect the integrity of the file pool if a system failure occurs.

logoff. The procedure by which a user ends a terminal session.

logon. The procedure by which a user begins a terminal session.

LU. Logical unit.

M

M. Megabyte.

machine. A synonym for a virtual machine running under the control of VM/370 or VM/SP HPO.

machine ID. A 2-byte field that uniquely defines a virtual machine within a virtual machine group. Machine ID is sometimes combined with task ID to uniquely identify a task within the virtual machine group.

MACLIB. Macro library.

MACLIB library. A library that contains macros, copy files, or source program statements for use under CMS.

macro. Synonym for *macrodefinition* and *macroinstruction.*

macrodefinition. A set of statements that defines the name of, format of, and conditions for generating a sequence of assembler language statements from a single source statement. Synonymous with *macro.*

macroinstruction. In assembler language programming, an assembler language statement that causes the assembler to process a predefined set of statements called a macrodefinition. The statements usually produced from the macrodefinition replace the macroinstruction in the program. Synonymous with *macro.*

macro library (MACLIB). A library of macrodefinitions and/or copy files. See also *MACLIB library.*

map. In CMS, the file that contains a CMS output listing, such as:

1. A list of macros in the MACLIB library, including macro size and location within the library.
2. A listing of the directory entries for the DOS/VS system or private source, relocatable, and/or core image libraries.
3. A linkage editor map for CMS/DOS programs.
4. A module map containing entry point locations.

mapped conversation. A conversation where data is sent in arbitrary length buffers. Programs do not have to be concerned with the format of data being sent. Contrast with *basic conversation.*

master file directory block. Synonym for *CMS file directory.*

MB. Megabyte.

MDISK. Another name for minidisk. It is also the user directory used to describe a user's storage space.

megabyte (MB). 1,048,576 bytes.

memo-to-users. A file provided on a service tape that contains specific service information for a product.

merge. When receiving files from a service tape using the VMFREC EXEC, the process of moving existing service files from each minidisk or SFS directory in the target string (as defined by the MERGE tag in the product parameter file) to the minidisk or SFS directory that contains the previous service level. The result is that the primary target minidisk or directory is left empty and ready to receive the latest service from the tape.

message. Data sent from a source application to a target application program in a conversation.

minidisk. Synonym for *virtual disk.*

minidisk directory. Synonym for *CMS file directory.*

module. A file whose external references have been resolved.

MP. Multiprocessor.

Multiple Virtual Storage (MVS). An alternative name for OS/VS2.

multiprocessor (MP). A computer using two or more processing units under integrated control.

Multisystem Communications Unit (MCU). An input/output device; e.g. the 3088 Multisystem Communications Unit, that interconnects up to eight systems using block multiplexer channels.

multitasking. Providing services for many tasks that are active at the same time.

MVS. Multiple Virtual Storage.

N

named system. A system that has an entry in the CP system name table (DMKSNTBL). The entry in the system name table includes the system name and other pertinent data so that the system can later be saved. See also *saved system.*

namedef. A temporary name that represents either: (1) a file name and file type, or (2) an SFS directory name. Namedefs are used in program functions so that it is not necessary to directly code a file name and file type or directory name in the program.

National Language Support (NLS). A feature of VM/SP HPO which allows the designation of languages other than that shipped with the product tape to be used as the system national language or as an additional language with which users can interact with VM.

NCCF. Network Communication Control Facility.

NCP. Network control program.

netdata. The name of the format used to send a file when the NEW option of the CMS SENDFILE command is specified.

network. Any set of two or more computers, workstations, or printers linked in such a way as to let data be transmitted between them.

Network Communication Control Facility (NCCF). An IBM product that can control a VM/SP HPO system through the Programmable Operator Facility in a mixed environment.

network control program (NCP). An IBM licensed program that provides communication controller support for single-domain, multiple-domain, and interconnected network capability.

NLS. National Language Support.

nondisruptive transition. A facility that enables an installation to transfer control of an operating system from VM/SP HPO virtual machine mode to native mode (and vice versa) without having to shut down and reinitialize the operating system.

noninteractive. The classification given to a virtual machine depending on this virtual machine's processing characteristics. The scheduler classifies a virtual machine as noninteractive if it is in queue 2 (but not in first queue slice) or in queue 3. Contrast with *interactive*.

non-IPL processor. In an AP or a MP system, the attached or second processor initialized at system generation time. Note that both the IPL processor and the non-IPL processor in a real MP configuration have I/O capabilities.

nonpaging mode. See *OS/VS1 nonpaging mode*.

nonvolatile storage (NVS). For the 3990 Storage Control Model 3, 4 megabytes of storage provided for new data that is to be saved but has not yet been stored on DASD, and for dual copy logging information.

nucleus. That part of the CP or CMS that is resident in main storage.

NUCON. The nucleus constant area of CMS which is invoked through the NUCON macro.

NVS. Nonvolatile storage.

O

object. (1) In computer security, anything to which access is controlled; for example, a file, a program, or an area of main storage. (2) A collection of data referred to by a single name. See also *library object*.

object code. Compiler or assembler output that is itself executable machine code or is suitable for further processing to produce executable machine code.

object directory. The output from the directory program.

OLTSEP. On Line Test Stand-alone Executive Program.

online test stand-alone executive program (OLTSEP). A program IBM uses for I/O maintenance.

operand. Information entered with a command name to define the data on which a command processor operates and to control the execution of the command processor.

Operating System/Virtual Storage (OS/VS). A family of operating systems that control IBM System/360 and System/370 computing systems. OS/VS includes VS1, VS2, MVS/SP, and MVS/XA.

OS. Operating system.

OS/VS. Operating System/Virtual Storage.

OS/VS1. A virtual storage operating system that is an extension of OS/MFT. Synonymous with *nonpaging mode*.

OS/VS1 nonpaging mode. If OS/VS1 executes under the control of a VM/SP HPO system that supports the VM/VS Handshaking feature and if the OS/VS1 address space is equal to the size of its VM/SP HPO virtual machine, OS/VS1 executes in nonpaging mode. When OS/VS1 executes in nonpaging mode, it uses fewer privileged instructions and avoids duplicate paging since paging is done only by CP.

overhead. The additional processor time charged to each virtual machine for the CP functions needed to simulate the virtual machine environment and for paging and scheduling time.

overlay. The technique of repeatedly using the same areas of internal storage during different stages of a program.

override. Synonym for *component parameter override*.

override area. Synonym for *component override area*.

override file. See *class override file*.

P

pack. A set of flat, circular recording surfaces that a disk storage device uses. A disk pack.

page. A fixed-length block that has a virtual address and can be transferred between real storage and auxiliary storage.

page allocation table. A table initialized when you IPL CMS. It is created at a location determined by the size of the virtual machine. CMS cannot reserve a segment space at the location of this table.

page frame. A block of 4096 bytes of real storage that holds a page of virtual storage.

page migration. Movement of inactive pages from paging areas higher in the paging hierarchy to paging areas lower in the paging hierarchy. Also movement of swap sets from SW paging storage areas to SW DASD areas.

paging. Transferring pages between real storage and external page storage.

paging area. An area of direct access storage (and an associated area of real storage) that is used by CP for the temporary storage of pages when paging occurs.

Paging Storage. The VM/SP HPO name used in reference to the hardware Expanded Storage assist. See also *Expanded Storage*.

parameter. A variable that is given a constant value for a specified application and that may denote the application.

parameter list (PLIST). In CMS, a string of 8-byte arguments that are used to call a CMS command or function. The first argument must be the name of the command or function to be called. General register 1 points to the beginning of the parameter list.

parent directory. The directory for a CMS disk that has a disk extension defined for it by the ACCESS command.

part. A CMS file provided on a product tape or service tape as input to the build process. See *build.* A part is the smallest serviceable unit of a component.

partitioned processing mode. A mode in which a multiprocessor is reconfigured into two separate and independent dyadic processors, each capable of executing an operating system of unique type or version.

password. In VM, a 1- to 8-character symbol that the user is required to supply at the time he logs on to identify himself. The password is normally protected from inadvertent disclosure to unauthorized personnel by not displaying the password or by masking the password as it is keyed in. A password may also be assigned to a virtual disk to control or limit access to that disk.

patch. A circumventive service change applied directly to object code in a text deck in a nucleus.

path. In APPC/VM or IUCV, a connection between two application programs that are on the same or different systems. Paths have names assigned to them.

PC. Personal computer.

PEL. Picture element.

performance option. One or more functions that can be assigned to a virtual machine to improve its performance, response time (if terminal-oriented) and/or throughput under VM/SP HPO.

personal computer (PC). A properly configured IBM Personal Computer or 3270 Personal Computer that allows communication between enhanced connectivity programs. These communicating programs reside on both a personal computer and host system.

PF key. Program function key.

PFnn keys. Special keys available on display terminals that can be defined as elementary cataloged procedures equivalent to one or more CP or CMS command lines, or input data for a virtual machine. A Program Function key can also be assigned a tabbing function for such terminals. When the PFnn key is pressed, the function or functions are performed, and the full command line can be displayed on the screen. Synonymous with *program function (PFnn) keys.*

physical saved segment. One or more pages of storage that have been named and retained on a CP-owned volume (DASD). Once created, it can be loaded within a virtual machine's address space or outside a virtual machine's address space. Multiple users can load the same copy. A physical saved segment can contain one or more logical saved segments. A system segment identification file (SYSTEM SEGID) associates a physical saved segment to its logical saved segments. See *logical saved segment* and *saved segment.*

physical screen. Synonym for *screen.*

preferred guest. An MVS/SP virtual machine that runs with preferred machine assist under VM/SP HPO.

POOLDEF file. A CMS file with a file type of POOLDEF that contains information that a file pool server uses to locate all the minidisks in the file pool.

pop. To remove an item from the top of a pushdown list. Contrast with *push.*

preferred auxiliary file. In CMS, an auxiliary file that applies to a particular version of a source module to be updated, if multiple versions of the module exist.

preferred machine assist. A hardware feature available on certain processors and with certain releases of MVS/SP and VSE that improves MVS/SP V=R virtual machine performance. The MVS/SP guest virtual machine operates in supervisor state with direct control of its own I/O operations under VM/SP High Performance Option. Note that preferred machine assist, is an extension of virtual machine assist, which eliminates CP simulation of certain instructions and interruptions.

preferred paging area. A special area of auxiliary storage where frequently used pages are paged out. It provides high-speed paging.

preferred virtual machine. A particular virtual machine that has one or more of the performance options assigned to it.

prefix storage area (PSA). A storage area where the normal low core IPL, logout, PSW information, the processor model, type, and features, BALR and FREE areas, monitor and trace data, and the needed linkage to virtual machines, real devices, and spool files are stored.

preventive service. The process of loading the contents of a program update tape (PUT) to minidisks , and applying all changes. The last step of preventive service is to perform the build process (see *build*).

primary user disk. Synonym for *A disk*.

Print Services Facility/VM (PSF/VM). The access method that supports the 3800 Printing Subsystem Models 3 and 8. PSF can interface either directly to a user's application program or indirectly through the JES of MVS.

private resource. A resource accessible from anywhere within a TSAF collection or SNA network and whose identity is known only within a single virtual machine. Contrast with *global resource* and *local resource*.

private resource manager. An application that runs in a server virtual machine and provides a service for connecting programs, but that does not identify itself to the TSAF collection. Contrast with *global resource manager* and *local resource manager*.

privilege class. One or more classes assigned to a virtual machine user in the directory entry; each privilege class specified allows a user to access a logical subset of the CP commands. There are eight IBM-defined privilege classes that correspond to specific administrative functions. They are:

Class A - operations
Class B - resource
Class C - programmer
Class D - spooling
Class E - analyst
Class F - service
Class G - general
Class H - reserved for IBM use.

The privilege classes can be changed to meet an installation's needs. See also *class authority and user class restructure*.

privileged instruction simulation. The CP-incurred overhead to handle privileged instructions for virtual machine operating systems that execute as if they were in supervisor state but which are executing in problem state. See also *virtual machine assist*.

problem state. A state during which the central processing unit cannot execute I/O and other privileged instructions. VM/SP HPO runs all virtual machines in problem state. See also *privileged instruction simulation*. Contrast with *supervisor state*.

Procedures Language/VM. A component of VM/SP. It contains the VM/SP System Product Interpreter, which processes the REXX language. This component contains the VM/SP implementation of the Systems Application Architecture™ Procedures Language in addition to the VM/SP System Product Interpreter function available in VM/SP Releases 3, 4, and 5. Procedures Language/VM provides a single source base for the VM/SP System Product Interpreter in both the CMS and GCS components.

process. A systematic sequence of operations to produce a specified result. A process is usually logical, not physical.

product. Any separately installable software program, whether supplied by IBM or otherwise, that is distinct from others and is recognizable by a unique identification code. Common examples of software products include:

5664-173 - Virtual Machine/System Product High Performance Option

5748-F03 - VS/FORTRAN Licensed Program

5748-RC1 - VM/Pass-through Product

The product identification code is unique to a given product, but does not identify the release level of that product.

Trademark of International Business Machines Corporation

processor-local queues. Those queues that are owned primarily by one processor.

product parameter file. A file containing installation and service parameters for a product: control options, minidisk and SFS directory assignments, and component part type/function lists.

product parameter override file. A file containing one or more component override areas.

PROFILE EXEC. A special EXEC procedure with a filename of PROFILE which can be created by a user. The procedure is normally executed immediately after CMS is loaded into a virtual machine.

program function (PFnn) keys. Synonym for *PFnn keys.*

program stack. Part of the console stack; it is a storage area used for temporary storage of lines or files being exchanged by programs executing under CMS. See also *console stack.*

program temporary fix (PTF). Code changes needed to correct a problem reported in an APAR. The corrected code is included in later releases. It includes element replacements (for object code) or element updates (for source code) for elements changed by the fix. It also defines limitations on which the PTF can be included. Each PTF is unique to a given release of a product. If the same problem occurs in multiple releases of a product, a separate PTF is defined for each release. A PTF defines only one replacement or update for each element, regardless of how many APARs are fixed.

Programmable Operator Facility. This facility enables automatic filtering and routing of messages from a specified virtual machine (for example the operator) to a logical operator virtual machine in a local distributed or mixed environment. The Programmable Operator Facility also permits installation defined actions to be performed automatically.

program status word (PSW). An area in storage that indicates the order in which instructions are executed, and to hold and indicate the status of the computer system.

program update service. Receiving the contents of a PUT, applying all or some of the changes, and rebuilding the serviced parts. See also *preventive service* and *selective preventive service.*

program update tape (PUT). A tape containing a customized collection of service tapes (preventive service) to match the products listed in a customer's ISD (IBM Software Distribution) profile. Each PUT contains cumulative service for the customer's products back to earlier release levels of the product still supported. The tape is distributed to authorized customers of the products at scheduled intervals or on request.

prompt. A displayed message that describes required input or gives operational information.

protocol. A set of rules for communication that are mutually understood and followed by two communicating stations or processes. The protocol specifies actions that can be taken by a station when it receives a transmission or detects an error condition.

PR/SM. A hardware function of VM/SP HPO which provides flexible partitioning of the IBM 3090 Processor Complex Enhanced Models into logical partitions.

PSA. Prefix storage area.

pseudonumber. A special order number assigned to a current or back-level edition of a publication so that it can continue to be ordered after a new edition is available.

PSF/VM. Print Services Facility/VM.

PSW. Program status word.

PTF. Program temporary fix.

PTF number. A number assigned by service organizations that uniquely identifies a PTF; for example, IBM uses UV*nnnnn* for a VM-unique product, and UP*nnnnn* for a cross-system product. PTFs for different products or different releases of a product have different numbers.

PTF parts list. A file that identifies the file name and file type of each part required to install a PTF. A PTF parts list is shipped on the service tape for each PTF on the tape.

push. To add an item to the top of a pushdown list. Contrast with *pop.*

PUT. Program update tape.

PVM. VM/Pass-Through Facility.

Q

queue-add. The action by the system scheduler, DMKSCH, of placing a runnable virtual machine on the run list.

queue-drop. The action by the system scheduler, DMKSCH, or removing a runnable virtual machine from the run list.

queue 1 (Q1). A queue of interactive users from which the dispatcher selects users to run. The virtual machines may or may not be immediately runnable. The size of queue 1 is dependent upon the system's multiprogramming level. The time slice given to a Q1 user is significantly shorter than that given to a Q2 or Q3 user, but the Q1 user's virtual machine is dispatched more frequently. Queue 1 is filled from the eligible list. See also *interactive, noninteractive, queue 2.*

queue 2 (Q2). A queue of interactive or noninteractive users from which the dispatcher selects users to run. The virtual machines may or may not be runnable. The time slice given to a Q2 user is significantly longer than that given to a Q1 user, but the Q2 user is dispatched less often since runnable users in Q1 are selected first. See also *interactive, noninteractive, queue 1.*

Q1. Queue 1.

Q2. Queue 2.

R

RACF. Resource Access Control Facility.

read. See *CP read, VM read.*

read authority. The authority to read the contents of a file without being able to change them. For a directory, read authority lets the user view the names of the objects in the directory.

real machine. The actual processor, channels, storage, and input/output devices required for operation of VM/SP HPO.

real storage. Synonymous with *absolute address* except for the effects of prefixing. Prefixing converts real storage addresses to absolute storage addresses.

receive. (1) Bringing into the specified buffer data sent to the user's virtual machine from another virtual machine or from the user's own virtual machine. (2) To load service files from a service tape.

records. A term used to describe a spool file generated to represent physical card decks.

register. See *general register.*

remote. Two entities (for example, a user and a server) are said to be remote to each other if they belong to different systems within a collection, or to different nodes within a SNA network.

Remote Spooling Communications Subsystem (RSCS). A virtual machine subsystem of VM that provides for the transfer of spool files between VM users, remote stations, and remote and local batch systems in SNA, non-SNA mixed networks.

requester. The program that relays a request to another computer through the server-requester programming interface (SRPI). Contrast with *server.*

reserved filetypes. (1) Filetypes recognized by the CMS editors (EDIT and XEDIT) as having specific default attributes that include: record size, tab settings, truncation column, and uppercase or lowercase characters associated with that particular filetype. The CMS Editor creates a file according to these attributes. (2) Filetypes recognized by CMS commands; that is, commands that only search for and use particular filetypes, or create one or more files with a particular filetype.

resource. A program, a data file, a specific set of files, a device, or any other entity or a set of entities that the user can uniquely identify for application program processing in a VM system.

Resource Access Control Facility (RACF). A licensed program that provides for access control by identifying and verifying users to the system, authorizing access to DASD data sets, logging detected unauthorized attempts to enter the system, and logging detected accesses to protected data sets.

resource. A program, a data file, a specific set of files, a device or any other entity or a set of entities that you might want to uniquely identify for purposes of application program processing in a VM system. A resource can be identified by up to eight characters.

resource ID. A 1- to 8-character name used to identify a resource.

resource manager. An application running in a server virtual machine that directly controls one or more VM resources. There are three categories of VM resource managers: global, local, and private.

response time. (1) The time between the submission of an item of work to a computing system and the return of results. (2) In systems with time sharing, the time between the end of a block or line-end character of terminal input and the display of the first character of system response at the terminal.

restricted passwords. Commonly published passwords that are not permitted in the object directory. A user who supplies a restricted password is denied access to the system. These commonly published or restricted passwords are contained in a file called RPWLIST DATA.

Restructured Extended Executor (REXX) language. A general purpose, high-level programming language, particularly suitable for EXEC procedures, XEDIT macros, or programs for personal computing. (The language is documented in the *VM/SP System Product Interpreter Reference*, SC24-5239.) Procedures, XEDIT macros, and programs written in this language are interpreted and executed by the System Product Interpreter.

REXX. Restructured Extended Executor (REXX) language.

REXX program. Synonym for *REXX EXEC*.

RSCS. Remote Spooling Communications Subsystem.

run list. A physical list in storage that is sorted by deadline priority. It contains interactive and noninteractive virtual machines who are competing for processor resources and whose storage and processor requirements can be accommodated. These virtual machines are considered in-queue and are either runnable or nonrunnable. When a virtual machine is dropped from the run list, replacement is made from the eligible list. See also *eligible list*, *dispatch list*.

S

saved segment. A segment of storage outside the address range of a virtual machine that has been saved and assigned a name. The saved segments can be physical saved segments that CP recognizes or logical saved segments that CMS recognizes. The segments can be loaded and shared among virtual machines, which helps use real storage more efficiently, or a private, nonshared copy can be loaded into a virtual machine. See also *logical saved segment* and *physical saved segment*.

saved system. A special nonrelocatable copy of a virtual machine's virtual storage and associated registers kept on a CP-owned disk and loaded by name instead of by I/O device address. Loading a saved system by name substantially reduces the time it takes to IPL the system in a virtual machine. In addition, a saved system such as CMS can also share one or more 64K segments of reenterable code in real storage among virtual machines. This reduces the cumulative real main storage requirements and paging demands of such virtual machines.

scale. A line on the System Product Editor's (XEDIT) full-screen display, used for column reference.

S disk. See *CMS system disk*.

SCP. System control programming.

screen. An illuminated display surface; for example, the display surface of a CRT. Synonymous with *physical screen*.

SCRIPT. See *SCRIPT/VS*.

SCRIPT/VS. A component of the IBM Document Composition Facility licensed program (available from IBM for a license fee). For additional information on SCRIPT/VS usage, see *Document Composition Facility: User's Guide*, SH20-9161.

section size. The number of elements of a vector register; the dimension of a vector.

segment. (1) A contiguous 64KB area of virtual storage (not necessarily contiguous in real storage) that is allocated to a job or system task.

segment number. The part of a virtual storage address needed to refer to a segment.

segment protect. A hardware feature, available on certain processors, that provides protection for shared segments at the hardware level.

selective preventive service. The selective application of PTFs from the PUT. Contrast with *preventive service*.

server. A program or set of programs executing in a virtual machine and managing access to one or more VM resources; also called a resource manager. Contrast with *requester*.

server-requester programming interface (SRPI).

1. A protocol between requesters and servers in an enhanced connectivity network. Includes the protocol to define cooperative processing subsystem.
2. The interface that enables enhanced connectivity between requesters and servers in a network.

service. Changing a product after installation. See *corrective service, local service,* and *program update service*.

service routines. CP or CMS routines used for addressing and updating directories; formatting or initializing disks; or performing disk, tape, or terminal input/output functions.

service tape. The service for individual products. The tape, which is distributed at scheduled intervals, contains a collection of PTFs, for selected program products. The tape contains service for those IBM-supplied software products for which the customer is licensed. The service tape is cumulative back to the earliest release level still available from IBM Software Distribution (ISD).

session. The SNA term for a connection between two LUs. The LUs involved allocate conversations across sessions.

sever. Ending communication with another virtual machine or with the user's own virtual machine.

SFS. Shared file system.

SFS directory. A group of files. SFS directories can be arranged to form a hierarchy in which one directory can contain one or more subdirectories as well as files.

shared file system (SFS). A part of CMS that lets users organize their files into groups known as *directories* and to selectively share those files and directories with other users.

shared segment. A feature of a saved system that allows one or more segments of reentrant code in real storage to be shared among many virtual machines. For example, if a saved CMS system was generated, segment 1 of the CMS nucleus is shared in real storage among all CMS virtual machines loaded by name; that is, every CMS machine's segment of virtual storage maps to the same 64K of real storage. See also *saved system, discontiguous system.*

shared segment protection. A feature that ensures that one virtual machine does not access a shared segment that another virtual machine has changed. CP does not allow any user to change the storage keys on the protected page, thus preventing other users from accessing the information on that page.

simultaneous peripheral operations online (SPOOL). (1) (Noun) An area of auxiliary storage defined to temporarily hold data during its transfer between peripheral equipment and the processor. (2) (Verb) To use auxiliary storage as a buffer storage to reduce processing delays when transferring data between peripheral equipment and the processing storage of a computer.

signaling attention. Pressing a key or keying in a CP command to present an attention interruption to CP or to a user's virtual machine.

single processor mode. In tightly coupled multiprocessing (MP) or attached processor (AP) systems, single processor mode allows an installation to dedicate a processor to an MVS V = R virtual machine. In single processor mode, VM/SP HPO runs in uniprocessor mode in the main processor, and the MVS V = R virtual machine runs under VM/SP HPO in the main processor and has the exclusive use of the other processor for MP or AP operations. However, other virtual machines can operate under VM/SP HPO concurrently with the MVS V = R virtual machine in single processor mode. With certain releases of MVS/SP, the single processor user must also be the preferred guest. Single processor mode should not be confused with VM/SP HPO uniprocessor mode.

SIO. Start I/O.

SNA. Systems Network Architecture.

source code. A program that has not yet been compiled or assembled to produce the corresponding object code.

source file. A file that contains source statements for such items as high-level language programs and data description specifications.

source directory. The directory control statements that constitute the input to the directory program.

SPOOL. Simultaneous peripheral operations online.

spool file. Data from a virtual machine stored in a virtual card reader, printer, or punch.

spoolid. A spool file identification number that is assigned to each user's spool files. Each virtual machine user may have up to 9900 spool files, with the spoolid numbers ranging from 0001 to 9900.

spooling. The processing of files created by or intended for virtual readers, punches, and printers. The spool files can be sent from one virtual device to another, from one virtual machine to another, and to real devices. See also *virtual console spooling.*

spooling devices. Input/output devices (card readers, punches, printers, DASD) used for reading input and writing output.

SRPI. Server-requester programming interface.

SSBLOK. Swap set block.

SSP. System service program.

stack. See *console stack; program stack.*

stand-alone. Pertaining to an operation independent of another device, program, or system.

starter system. A very basic VM/SP HPO system that an installation can use to build a production VM/SP HPO system.

starter system tape. The tape used to start when installing a system "from scratch." Starter system tapes are DASD-type specific; for instance, a 3350 starter system cannot be used on any DASD type other than a 3350. Once stored, the starter system is a very basic VM system that can be used to build a production system.

storage group. A subset of minidisks within a file pool. Each storage group is identified by a number.

storage key. A four-bit control field associated with either 2KB or 4KB blocks of real storage.

subcommand. The commands of processors such as EDIT or System Product Editor (XEDIT) and DEBUG that run under CMS.

subdirectory. Any directory below a user's top directory. The CREATE DIRECTORY command creates subdirectories. There can be up to eight levels of subdirectories with no limit on the number of them at each level, other than overall DASD space limits. Each level of a subdirectory is an additional identifier of up to 16 characters that is appended to next higher level subdirectory.

subpools. Fixed-size storage groups that the system uses to satisfy free-storage requests.

superseded. In reference to service, a change which is no longer needed (some later fix has replaced it) is called superseded.

supervisor call instruction (SVC). An instruction that interrupts a program being executed and passes control to the supervisor so that it can do a specific service indicated by the instruction.

supervisor state. A state during which the central processing unit can execute input/output and other privileged instructions. In VM/SP HPO, MVS/SP with preferred machine assist and CP can execute in the supervisor state; all virtual machine operating systems execute in problem state. Contrast with *problem state*.

SVC. Supervisor call instruction.

swap area. An area on a paging media allocated for swapping by the SW option on the SYSPAG and SYSXSTOR macros.

swap set block (SSBLOK). The control block that describes each swap set of a virtual machine. It contains information about each page in the swap set, and queuing and indirectly (through SCBLOK) VMBLOK pointers.

swapping. The process of moving swap set pages to and from a swap area.

synonym. In CMS, an alternative command name defined by the user as equivalent to an existing CMS command name. Synonyms are entries in a CMS file with a filetype of SYNONYM. Issuing the SYNONYM command allows use of those synonyms until that terminal session ends or until the use of synonyms is revoked by issuing the SYNONYM command with no operands.

syntax. The rules for the construction of a command or program.

SYSPAG macro. A macro in DMKSYS used to define how the CP-owned volumes and Paging Storage are to be allocated for swapping, paging, spooling, page migration, and dump. Using this macro, you define Paging Storage in terms of increments.

system administrator. The person responsible for maintaining a computer system.

system national language. The language automatically set for all virtual machines once VM is completely installed from the product tape — when users log on, they receive messages, see panels and can enter CMS commands in this language. VM is shipped with mixed-case American English as the national language, although the product tape also contains files for uppercase English. You can also order other languages for your VM system. When you order another language, you receive a feature tape with files that contain translated information that allows users to interact with VM in this new language.

system control programming (SCP). IBM-supplied programming fundamental to the operation and maintenance of the system. It serves as an interface with IBM licensed programs and user programs and is available without additional charge.

system integrity. The property of a system that is designed, implemented, and maintained to protect itself from unauthorized access.

System Product Editor. The CMS facility, comprising the XEDIT command and XEDIT subcommands and macros, that allows a user to create, modify, and manipulate CMS disk files.

System Product Interpreter. The component that processes procedures, XEDIT macros, and programs written in the Restructured Extended Executor (REXX) Language.

system profile. An EXEC (SYSPROF) that resides in a saved system or on a system disk and called by CMS initialization. It contains some initialization functions, and provides a means for installations to override the default CMS environment by tailoring the EXEC to suit the installation.

system service program (SSP). In ACF/TCAM, an IBM-supplied or user-supplied program that does system-oriented auxiliary functions in support of the message control program. System service programs run under control of the initiator as attached subtasks.

System/370. System/370 processors and the 4341, 4381, 303x, 308x, and 3090 processors.

Systems Application Architecture™. A defined set of interfaces, conventions, and protocols that can be used across various IBM systems.

systems network architecture (SNA). The description of the logical structure, formats, protocols, and operational

Systems Application Architecture is a trademark of the International Business Machines Corporation.

sequences for transmitting information units through and controlling the configuration and operation of networks.

SYSXSTOR macro. A macro in DMKSYS used to define how Paging Storage is to be allocated for swapping and paging. Using this macro, you define Paging Storage in terms of megabytes.

T

T-disk. See *temporary disk.*

target. One of several ways to identify a line to be searched for by the System Product Editor. A target may be specified as an absolute line number, a relative displacement from the current line, a line name, or a string expression.

temporary disk. In VM, an area on a direct access storage device available to the user for his newly created or stored files until he logs off, at which time the area is released. Temporary disk space is allocated to the user when he logs on or when he issues the CP DEFINE command.

terminal. A device, usually equipped with a keyboard and a display, capable of sending and receiving information over a communications channel. With VM, the terminal is used to communicate with the system.

terminal input buffer. Part of the console stack; this buffer holds lines entered at your terminal until CMS processes them.

terminal user. Anyone who uses a terminal to log on to VM/SP HPO.

text deck. An object code file that must be additionally processed to produce executable machine code.

time sharing. A method of using a computing system that allows a number of users to execute programs concurrently and to interact with the programs during execution.

time stamp. A record containing the TOD clock value stored in its internal 32-bit binary format.

token. An 8-character symbol created by the CMS EXEC processor when it scans an EXEC procedure or edit macro statements. Symbols longer than 8 characters are truncated to 8 characters.

tokenized parameter list (PLIST). A parameter list in which each item in the list is tokenized; that is, each item is truncated or padded on the right with blanks to fit into an eight-character token. See also *token, untokenized parameter list.*

tokenized PLIST (parameter list). A string of doubleword aligned parameters occupying successive doublewords.

top directory. The directory created for a user when the user is enrolled in a file pool. The name of the top directory is the same as the person's user ID.

trace table. See *CP trace table.*

transient program area. In CMS, the virtual storage area occupying locations X'E000' to X'10000'. Some CMS commands and user programs may execute in this area of CMS storage.

transparent. An application-to-server interface is said to be transparent if it is identical for local and remote servers.

transparent services access facility (TSAF). A facility that lets users connect to and communicate with local or remote virtual machines within a group of systems. With TSAF, a user can connect to a program by specifying a name that the program has made known, instead of specifying a userid and nodeid.

TRAPRED. This command accesses the CPTRAP reader file and the data collected in the file.

TSAF. See *transparent services access facility.*

TSAF collection. A group of VM processors, each with a TSAF virtual machine, connected by CTC, binary synchronous lines, or LANs.

TSAF virtual machine. The virtual machine that lets user programs connect to and communicate with virtual machines on different VM systems.

TSAF virtual machine component. A component in VM that handles communication between systems by letting APPC/VM paths span more than one system.

U

uniprocessor (UP). A computer configuration that consists of a single processor or that uses only one processor of an attached processor (AP) or multiprocessor (MP) system.

untokenized parameter list. A parameter list in which no restrictions are placed on the structure of the items in the list. See also *token, tokenized parameter list.*

UP. Uniprocessor.

update file. The actual changed code to be used by the assembler or compiler to produce the object code. This file exists only in source-maintained products.

update service. Servicing a part by applying a change to a source file statement, then assembling or compiling the source file to produce a new object file.

user. A program accessing one or more resources. In a VM system, a user executes in a virtual machine and is identified by a userid.

user data. In reference to a file pool, any data that resides in storage groups 2 through 32767.

user ID (user identification). A 1- to 8-character alphameric symbol identifying each terminal user. *user ID* refers to a human user's identifier. When shown in italics (*userid*), this term denotes a variable to be specified; for example, in a command syntax diagram. Synonymous with *userid.* See *vmid.*

user modification. Any change that a user originates for a product or component.

user-written CMS command. Any CMS file created by a user that has a file type of MODULE or EXEC. Such a file can be executed as if it were a CMS command by issuing its file name, followed by any operands or options expected by the program or EXEC procedure.

userid. See *user ID (user identification).*

V

V=R. Synonym for *virtual=real.*

VCNA. VTAM Communications Network Application.

vector. A quantity that has magnitude, direction, and sense, and that is commonly represented by a directed line segment whose length represents the magnitude and whose orientation in space represents the direction. When used with the Vector Facility, a vector consists of a nx1 element array, each element of which contains 32 bits.

virtual address. An address that refers to virtual storage or a virtual I/O device address, and that must, therefore, be translated into a real storage or I/O device address when it is used.

virtual console. A 3210, 3215, 1052, or 3270 system console simulated on a communications terminal (such as a 2741 or 3278) by CP. The virtual device type and I/O address are defined in the VM directory entry for that virtual machine.

virtual console spooling. The writing of console input/output on disk as a printer spool file instead of, or in addition to, having it typed or displayed at the virtual machine console. The console data includes messages, responses, commands, and data from or to CP and the virtual machine operating system. The user can invoke or terminate console spooling at any time and as often as he likes. When the console spool file is closed, it becomes a printer spool file.

virtual device. A device simulated for a virtual machine by CP. The MAXDEV *xxxx* option on the OPTION directory control statement allows you to attach up to 3277 devices to your virtual machine when VDEVSIZE is 10 doublewords. Without the MAXDEV option, you can attach 410 devices to your virtual machine.

virtual disk. A logical subdivision (or all) of a physical disk pack that has its own virtual device address, consecutive virtual cylinders (starting with virtual cylinder zero), and a volume table of contents (VTOC) or disk label identifier. Each user virtual disk is preallocated and defined through a VM directory entry as belonging to some user. Synonymous with *minidisk.*

virtual interval timer assist. A hardware assist function, available only on a processor that has Extended Control-Program Support, that provides, if desired, a hardware updating of each virtual machine's interval timer at location X'50'.

virtual machine (VM). A functional equivalent of a real machine.

virtual machine assist (VMA). A hardware feature available on certain VM/SP HPO-supported System/370 models, that causes a significant reduction in the real supervisor state time used by VM/SP HPO to control the operation of virtual storage systems such as VSE, DOS/VS and OS/VS and, to a lesser extent, CMS, DOS, and OS when executing under VM. VM/SP HPO supervisor state time is reduced because the virtual machine assist feature, instead of VM/SP HPO, intercepts and handles interruptions caused by supervisor call instructions (SVCs), other than SVC 76, and certain privileged instructions. See also *Extended Control-Program Support, CP assist, expanded virtual machine assist, virtual interval timer assist.*

virtual machine communication facility (VMCF). A CP function that provides a method of communication and data transfer between virtual machines operating under the same VM/SP HPO system.

virtual machine control block (VMBLOK). The CP control block that contains, for each virtual machine, the following types of information: the dispatch and priority level of the virtual machine, the virtual machine's processor registers, preferred virtual machine options currently in effect, and information concerning all other significant activities.

virtual machine group. The concept in the group control system of two or more virtual machines associated with each other through the same named system (e.g. IPL GCS1). Virtual machines in a group share common read/write storage and can communicate with one another through facilities provided by the group control system. Synonymous with *group.*

Virtual Machine/System Product (VM/SP). A licensed program that controls virtual machines. See *IBM Virtual Machine/System Product (VM/SP).*

Virtual Machine/System Product High Performance Option (VM/SP HPO). A licensed program that enhances VM/SP for large system environments. See *IBM Virtual Machine/System Product High Performance Option (VM/SP HPO).*

virtual = real option. A VM performance option that allows a virtual machine to run in VM's virtual = real area. This option eliminates CP paging and, optionally, CCW translation for this virtual machine.

virtual storage. Storage space that can be regarded as addressable main storage by the user of a computer system in which virtual addresses are mapped into real addresses. The size of virtual storage is limited by the addressing scheme of the computing system and by the amount of auxiliary storage available, and not by the actual number of main storage locations.

virtual storage access method (VSAM). An access method for direct or sequential processing of fixed and variable-length records on direct access devices. The records in a VSAM data set or file can be organized in logical sequence by a key field (key sequence), in the physical sequence in which they are written on the data set or file (entry-sequence), or by relative-record number.

virtual storage extended (VSE). The generalized term that indicates the combination of the DOS/VSE system control program and the VSE/Advanced Functions program product. Note that in certain cases, the term DOS is still used as a generic term; for example, disk packs initialized for use with VSE or any predecessor DOS or DOS/VS system are sometimes called DOS disks. Also note that the DOS-like simulation environment provided under the VM/SP CMS component and CMS/DOS exists on VM/SP and VM/SP HPO program products and continues to be called CMS/DOS.

Virtual Telecommunications Access Method (VTAM). An IBM licensed program that controls communication and the flow of data in a computer network. It provides single-domain, multiple-domain, and multiple-network capability. VTAM runs under MVS, OS/VS1, VM/SP, and VSE.

VM. Virtual machine.

VM directory. A CP disk file that defines each virtual machine's normal configuration; the userid, password, normal and maximum allowable virtual storage, CP command privilege class or classes allowed, dispatching priority, logical editing symbols to be used, account number, and CP options desired.

VM read. The situation in which the user's virtual machine is not executing, but is waiting for a response or a request for work from the user. On a typewriter terminal, the keyboard is unlocked; on a display terminal, the screen status area indicates VM READ.

VMA. Virtual machine assist.

VMBLOK. Virtual machine control block.

VMCF. Virtual Machine Communication Facility.

vmid. *Vmid* refers to an abstract user's identifier (the RSCS service machine, for example). When shown in italics (*vmid*), this term denotes a variable to be specified; for example, in a command syntax diagram. See *user ID (user identification).*

VMLIB. The name of the CSL supplied with VM/SP that contains routines to do various VM functions.

VM/Pass-Through Facility. A facility that lets VM users interactively access remote system and processor nodes. These can be remote IBM 4300 processors, other VM systems, with or without this facility installed, or System/370-compatible non-VM systems.

VMSIZE. The high storage boundary of a virtual machine.

VM/SP. See *IBM Virtual Machine/System Product (VM/SP).*

VM/SP HPO. See *IBM Virtual Machine/System Product High Performance Option (VM/SP HPO).*

VM/SP HPO product tape. A tape that contains only those modules and macros added or altered since VM/370 Release 6.

VM/XA SP. VM/Extended Architecture System Product operating system.

V = R. Synonym for *virtual = real option.*

VSAM. Virtual storage access method.

VSCS. VTAM SNA Console Support.

VSE. Virtual storage extended.

VSM. This term refers to the VTAM Service Machine which is a virtual machine. See *VTAM service machine.*

VTAM. Virtual Telecommunications Access Method.

VTAM service machine (VSM). A virtual machine that contains an operating system (OS/VS1 or DOS/VSE), an access method (ACF/VTAM or ACF/VTAME), and VM/VCNA. VSM forms the interface for SNA communication in VM/SP HPO.

VTAM SNA Console Support (VSCS). A component of ACF/VTAM that lets SNA terminals function as virtual machine consoles.

W

warm start. (1) The result of an IPL that does not erase previous system data. (2) The automatic reinitialization of the VM/SP HPO control program that occurs if the control program cannot continue processing. Closed spool files and the VM/SP HPO accounting information are not lost. Contrast with *cold start, checkpoint start, force start.*

window. In VM, an area on the physical screen where virtual screen data can be displayed.

windowing. In VM, a set of functions which allow the user to view and manipulate data in user-defined areas of the physical screen called "windows." Windowing support allows the user to define, position, and overlay windows; scroll backward and forward through data; and write data into virtual screens.

work station. See *personal computer.*

work unit. In CMS, a group of related operations that can be either committed or rolled back as a unit. When the operations associated with a work unit are committed or rolled back, new operations can be associated with the same work unit. These operations can also be committed or rolled back. (The work unit is, in a sense, reusable.) Multiple work units may be active. See *active work unit* and *inactive work unit.*

work unit ID. A nonnegative integer value, unique within a virtual machine, that identifies a work unit.

write authority. The authority to read or change the contents of a file or directory. Write authority implies read authority.

write verification. Optional read-back checking for count-key data DASD performed by the CP Format/Allocate service program.

X

XEDIT. See *System Product Editor.*

Y

Y disk. An extension of the CMS system disk.

Y-STAT. This is a block of storage that contains the file status tables (FSTs) associated with the Y disk. The FSTs are sorted so that a binary search can be used to search for files. The Y-STAT usually resides in the CMS nucleus so it can be shared. Only files with filemode of 2 will have their associated FSTs in the Y-STAT.

Z

ZAP. A CMS command that changes or dumps MODULE, LOADLIB, or TXTLIB files. It may be used to change either fixed or variable length MODULE files. It is for use by system support personnel only.

Numerics

2305. Refers to the IBM 2305 Fixed Head Storage Device, Models 1 and 2.

2741. Refers to the IBM 2741 Terminal. Information on the 2741 also applies to the IBM 3767 Terminal, unless otherwise noted.

3033. Refers to the IBM 3033 Processor.

3081. Refers to the IBM 3081 Processor Unit Model D16.

3088. Refers to the IBM 3088 Multisystem Communications

3090. Refers to the IBM 3090 Processor.

3270. Refers to a series of IBM display devices; for example, the IBM 3275, 3276 Controller Display Station, 3277, 3278, and 3279 Display Stations, the 3290 Information Panel, and the 3287 and 3286 printers. A specific device type is used only when a distinction is required between device types. Information about display terminal usage also refers to the IBM 3138, 3148, and 3158 Display Consoles when used in display mode, unless otherwise noted.

3289. Refers to the IBM 3289 Model 4 Printer.

3310. Refers to the IBM 3310 Direct Access Storage Device.

3330. Refers to the IBM 3330 Disk Storage Device.

3340. Refers to the IBM 3340 Direct Access Storage Device.

3350. Refers to the IBM 3350 Direct Access Storage Device when used in native mode.

3370. Refers to the IBM 3370 Direct Access Storage Device.

3375. Refers to the IBM 3375 Direct Access Storage Device.

3380. Refers to the IBM 3380 Direct Access Storage Device.

3422. Refers to the IBM 3422 Magnetic Tape Subsystem.

3480. Refers to the IBM 3480 Magnetic Tape Subsystem.

3725. Refers to the IBM 3725 Communication Controllers.

3800. Refers to the IBM 3800 Printing Subsystems. A specific device type is used only when a distinction is required between device types.

3880. Refers to the IBM 3880 Storage Control Units.

3990. Refers to the IBM 3990 Direct Access Storage Control.

4245. Refers to the IBM 4245 Printer.

4248. Refers to the IBM 4248 Printer.

4341. Refers to the IBM 4341 Processor.

4381. Refers to the IBM 4381 Processor.

6262. Refers to the IBM 6262 Printer.

9313. Refers to the IBM 9313 Direct Access Storage Device.

Index

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

Virtual Machine/
System Product
High Performance Option

READER'S
COMMENT
FORM

Release 6 Guide

Order No. SC23-0189-4

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you. Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate.

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.*

How did you use this publication?

[ ] As an introduction
[ ] As a reference manual
[ ] For another purpose (explain) ______________________

____

[ ] As a text (student)
[ ] As a text (instructor)

Is there anything you especially like or dislike about the organization, presentation, or writing in this manual? Helpful comments include general usefulness of the book; possible additions, deletions, and clarifications; specific errors and omissions.

Page Number: Comment:

What is your occupation? ______________________

Newsletter number of latest Technical Newsletter (if any) concerning this publication: __________

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

____

____

IBM branch office serving you ______________________

Thank you for your cooperation. No postage stamp 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.)

SC23-0189-4

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 52Q MS 511
Neighborhood Road
Kingston, New York 12401

Fold and Tape Please Do Not Staple Fold and Tape

PRINTED IN U.S.A.

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

Virtual Machine/
System Product
High Performance Option

READER'S
COMMENT
FORM

Release 6 Guide

Order No. SC23-0189-4

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you. Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate.

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.*

How did you use this publication?

[ ] As an introduction

[ ] As a reference manual

[ ] For another purpose (explain) ________________________

[ ] As a text (student)

[ ] As a text (instructor)

Is there anything you especially like or dislike about the organization, presentation, or writing in this manual? Helpful comments include general usefulness of the book; possible additions, deletions, and clarifications; specific errors and omissions.

Page Number: Comment:

What is your occupation? ________________________

Newsletter number of latest Technical Newsletter (if any) concerning this publication: ________

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

IBM branch office serving you ________________________

Thank you for your cooperation. No postage stamp 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.)

SC23-0189-4

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 52Q MS 511
Neighborhood Road
Kingston, New York 12401

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

IBM®

PRINTED IN U.S.A.

IBM®

Program Number
5664-173

File Number
S370-20

SC23-0189-4

Printed in U.S.A.