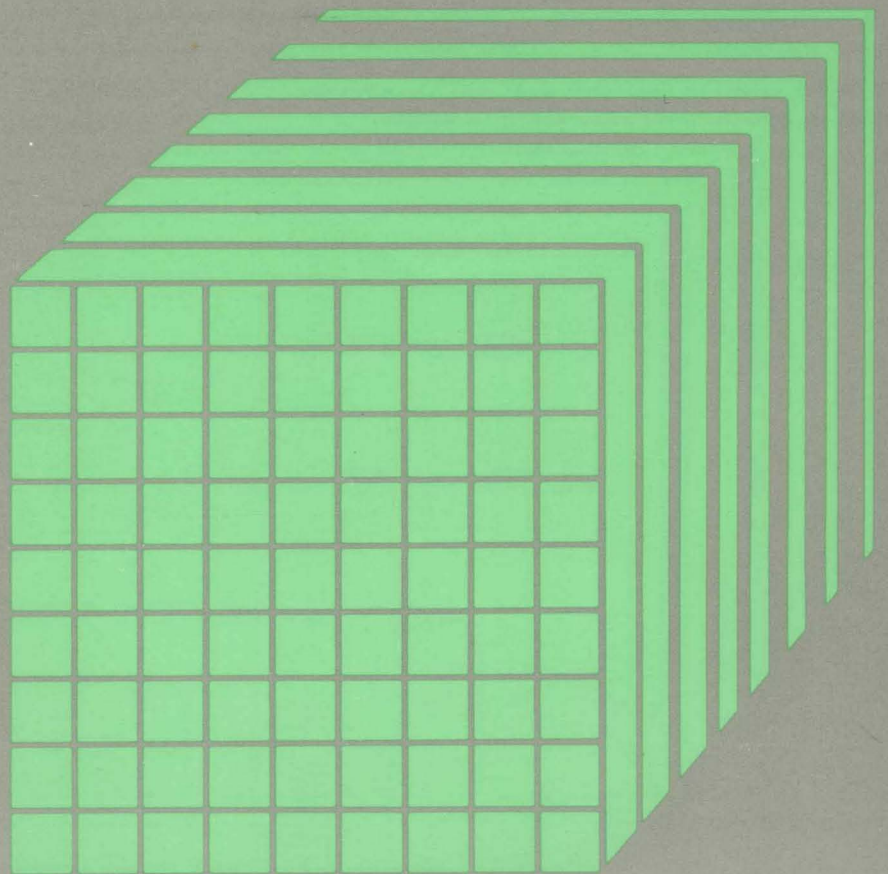


IBM

Virtual Machine/
System Product

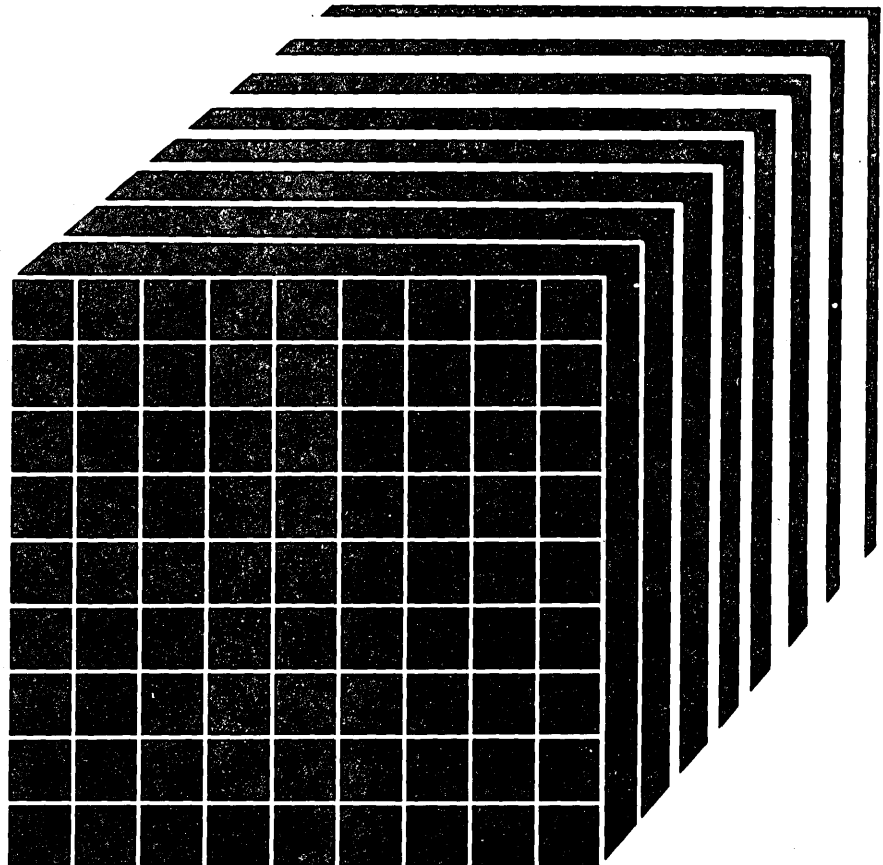
Release 3 Guide





Virtual Machine/ System Product

Release 3 Guide



First Edition (July 1983)

This edition, SC24-5240-0, applies to Release 3 of the Virtual Machine/System Product (VM/SP), program number 5664-167.

Changes are made periodically to the information herein; before using this publication in connection with the operation of IBM systems, consult the latest *IBM System/370 and 4300 Processors Bibliography*, GC20-0001, for the editions that are applicable and current.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM program product in this publication is not intended to state or imply that only IBM's program product may be used. 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 reader's comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Programming Publications, Dept. G60, P.O. Box 6, Endicott, NY U.S.A. 13760. IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Preface

This book describes the new and changed functions in Virtual Machine/System Product (VM/SP) Release 3. These changes are duplicated, where appropriate, in other books in the VM/SP library. By describing in one place the main implication of each change, this manual makes it easier for you to understand and take advantage of the changes. However, this book is not intended to replace the full VM/SP library for either new or current users.

If you are using VM/SP Release 1 or VM/370, refer to the *VM/SP General Information Manual* from Release 2 (GC20-1838-2).

Who This Book Is For

This book is specifically intended for current users of VM/SP Release 2 who plan to migrate to Release 3. It is for data processing managers, system programmers, system analysts, and other programming personnel responsible for migrating an installation from VM/SP Release 2 to VM/SP Release 3. Nevertheless, the changes to VM/SP are described in sufficient detail to enable other users, for example application programmers, to learn about the new release before it is installed.

How This Book Is Organized

“Part 1: General Information” is an overview of VM/SP Release 3. It summarizes the environmental considerations of the release and describes the reorganization of the VM/SP library.

“Part 2: New Facilities and Enhancements” describes the functional changes introduced in Release 3. Each section describes changes that affect a different aspect of VM/SP.

“Part 3: Migration” describes the general requirements and migration aids for migrating to Release 3. This will help you take advantage of the new facilities in Release 3.

“Part 4: Internal Design Changes” describes changes to the internal design of VM/SP that have performance, usability, or maintenance implications.

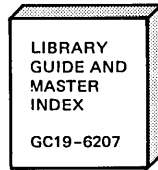
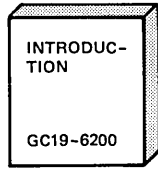
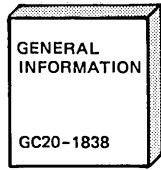
How To Use This Book

When using this book you should:

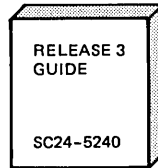
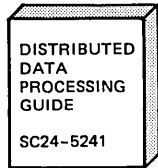
- Refer to Part 1 for a general overview of VM/SP Release 3.
- Refer to Part 2 for details of the functional changes introduced in VM/SP Release 3.
- Refer to Part 3 for details for migrating from VM/SP Release 2 to VM/SP Release 3.
- Refer to Part 4 for details of the internal design changes of VM/SP Release 3.

The VM/SP Library

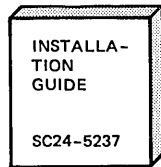
Evaluation



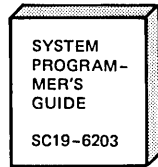
Planning



Installation



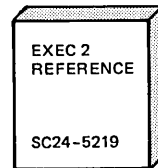
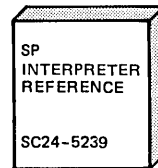
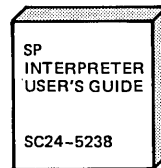
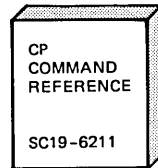
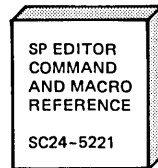
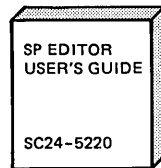
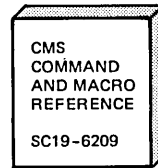
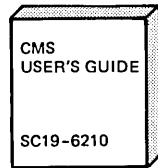
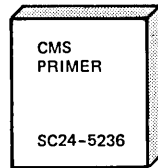
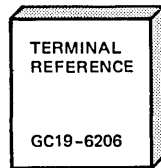
Administration



Operation



End Use



Reference Summaries

To order all the Reference Summaries, use order number SBOF 3820.

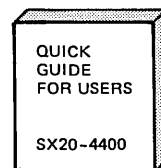
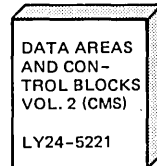
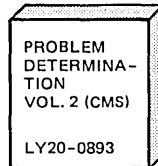
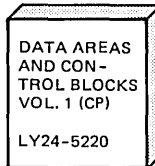
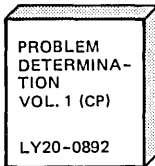
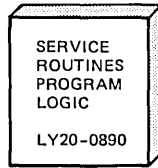
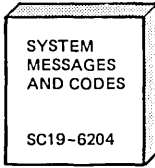
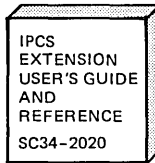


Figure 1 (Part 1 of 2). The VM/SP Library

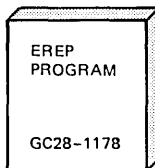
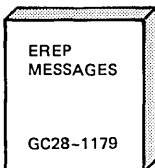
Program Service



Auxiliary Service Support

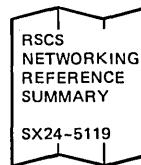
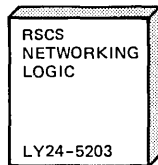
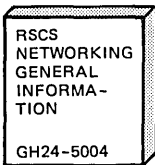


Device Support Facilities
IPCS Extension 5748-SA1

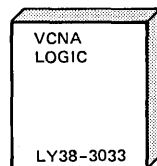
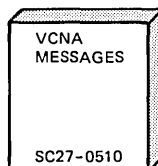
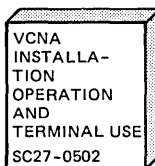
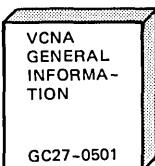


Environmental Recording
Editing and Printing
(EREP)

Auxiliary Communication Support



RSCS Networking
5748-XP1



VTAM Communications
Networking Application
(VCNA) 5735-RC5

Figure 1 (Part 2 of 2). The VM/SP Library

Contents

Part One: General Information	1
An Overview of VM/SP Release 3	2
Significant Additions	3
Support for Structured Query Language/Data System (SQL/DS)	3
Addition of the System Product Interpreter	3
Enhancements to the Programmable Operator Facility	4
Enhancements to the System Product Editor	4
Summary of Other Facilities and Enhancements	5
Environmental Considerations	7
Programming Requirements	8
VSE/VSAM Release 3	8
EREP Release 2.3	8
IPCS/E Release 2	8
SQL/DS Release 2	8
Statement of Direction	9
System Product Interpreter	9
System Product Editor	9
Statement of System Integrity	10
Responsibilities of the Installation	11
The VM/SP Library	12
The VM/SP Library Chart	13
New Look	15
Cover Design	15
Bill of Forms for Easy Ordering	16
New Books	18
VM/SP Distributed Data Processing Guide, SC24-5241	18
VM/SP Installation Guide, SC24-5237	18
VM/SP Release 3 Guide, SC24-5240	18
VM/SP System Product Interpreter User's Guide, SC24-5238	18
VM/SP System Product Interpreter Reference, SC24-5239	18
Restructured books	19
VM/SP CMS User's Guide, SC19-6210	19
VM/SP Data Areas and Control Block Logic	19
VM/SP General Information, GC20-1838	19
VM/SP Introduction, GC19-6200	19
VM/SP Planning Guide and Reference, SC19-6201	19
VM/SP Terminal Reference, GC19-6206	20
Part Two: New Facilities and Enhancements	21
Support for SQL/DS	22
SQL/DS Release 2	22
VM/SP Support for SQL/DS	23
Addition of the VM/SP System Product Interpreter	24
Advantages of the System Product Interpreter	24
Examples of System Product Interpreter EXECs	25
Enhancements to the Programmable Operator Facility	27
Message Routing with Nicknames	27
Availability Enhancements	28
Enhanced Text Comparison	28
EXEC Action Routines	28
Enhancements to Message Recording and Error Handling (logging)	28
Enhancements to the System Product Editor	30
Color and Extended Highlighting	30
Prefix Macro Support	31
Prefix Macros	31
Writing Prefix Macros	31
LPREFIX Subcommand	32

Selective Line Editing	33
ALL Macro	33
SET SELECT, SET DISPLAY, SET SCOPE, SET SHADOW	33
Screen Control Enhancements	34
Vertical Screen Split	34
Message Line	36
Middle of Screen	36
Command Line	36
PF Key Changes	36
XEDIT PF Key Changes	36
CMS Commands	37
Summary of PF Key Changes	38
Additional New Functions	38
Modifications to Existing Functions	39
Enhanced Serviceability	43
Support of System/370 Program Event Recording (PER)	43
Symptom Record (VM/IPCS/E)	44
DMSDOS Split	45
DMSQRY Split	45
CP Enhancements	47
Enhancements for Application Programs	47
Enhancements to the Inter-User Communication Vehicle (IUCV)	47
Addition of a DASD Block I/O System Service	49
Enhancements for Improved CP Performance	49
Nucleus Reduction	49
Enhancement to the SET QDROP Command	50
Support for Larger-than-4K Buffer Terminals	50
Other CP Enhancements	50
DIAGNOSE Code X'00'	50
DIAGNOSE Code X'8C'	50
DIAGNOSE Code X'14'	51
SET LOGMSG Command	51
CMS Enhancements	52
Enhancements for Application Programs	52
CMS Support of IUCV	52
Wait on ECB	53
End-of-Command Nucleus Extensions	53
ABEND Exit	54
Reading Directly from the Virtual Machine Console	54
Creating Your Own Immediate Commands	54
QSAM Tape End-of-Volume Exit	55
Support for CP DASD Block I/O System Service	55
Enhancements to the CMS File System	56
512-byte Blocksize Minidisk	56
RESERVE Command	56
Update-in-Place	57
Enhancements for Improved CMS Performance	57
LISTFILE and RENAME Migration	57
Reduction of the SCBLOCK Look-Aside Buffer	57
Removal of the CMSSEG Segment	57
Enhancement to GLOBALV	58
Other CMS Enhancements	58
Enhancements to TELL and DEFAULTS Commands	58
EXECOS Command	58
SET EXECRAC Command	59
Enhanced Support for VSE/VSAM	60
Support for VSE/VSAM Release 3	60
CATCHECK Command	61
Support for VSAM Assembler Language Macros	61
Support for VSE/VSAM Macros	61
Support for OS/VSAM Macros	61
VSEVSAM Command	62
CMS/VSAM Support of the 3380 Direct Access Storage	62

Previously Announced Functions	63
Support of the Missing Interrupt Handler	63
Support of the Speed Matching Buffer for the IBM 3375	64
Hardware Support	65
Printer Support	65
3262 Printer, Model 5	65
3800 Printing Subsystem Models 3 and 8	65
4245 Line Printer	65
4250 Printer	65
Tape Support	66
3430 Magnetic Tape Subsystem	66
Channel Communication Unit	66
3088 Multisystem Channel Communication Unit	66
Part Three: Migration	69
Migration Aids	70
SPOOL File	70
Programmable Operator Routing Tables	70
AP/MP Processor Unit	70
Migration Compatibilities/Incompatibilities	71
Migration Compatibilities	71
Migration Incompatibilities	71
Removal of the EDGAR Support	71
IPCS Incompatibilities with VM/SP	71
RSCS Incompatibilities with VM/SP	72
System Product Editor Migration Considerations	72
Performance	76
Part Four: Internal Design Changes	77
Module, Control Block, and Macro Changes	78
Abend Exit	78
CATCHCHECK Command	78
CMS File System Enhancement	79
CMSSEG Segment, Removal of	79
Communication Unit, 3088 Multisystem Channel Support	79
CP DASD BLOCK I/O System Service	80
CP, Other Enhancements to	80
DASD Block I/O System Service, CMS Support for	80
DEFAULTS Command	80
Diagnose Code X'00'	80
Diagnose Code X'8C'	81
Diagnose Code X'14'	81
DMSDOS Split	81
DMSQRY Split	81
End-of-Command Nucleus Extensions	81
EXECOS command	82
GLOVALV Enhancement	82
Immediate Commands, Creating Your Own	82
IUCV, CMS Support of	82
IUCV, CP Enhancements to	82
Larger-than-4K-Buffer Terminals, Support for	82
LISTFILE and RENAME Migration	83
Nucleus Reduction	83
Previously Announced Functions	83
Printer Support	83
Program Event Recording (PER)	83
Programmable Operator	84
QSAM Tape End-of-Volume Exit	84
Reading Directly from Your Virtual Machine Console	84
SCBLOCK Look-Aside Buffer, Reduction of	84
SET EXECRAC Command	84
SET LOGMSG Command	85
SET QDROP Command	85

SQL/DS	85
Symptom Record	85
System Product Editor	85
System Product Interpreter	86
Tape Support	86
TELL Command	86
VSAM Support of the 3380 Direct Access Storage	87
VSE/VSAM Release 3 Support	87
VSE/VSAM, Support for Assembler Language Macros	87
Wait on ECB	87
General Changes Follow For:	88
CMS Module Splits	88
CMS Deleted Assemble Files	88
CMS Deleted EXECs	88
CMS Deleted Modules	88
CMS Deleted XEDIT Files	88
Files that are now Mode 1 instead of Mode 2	88
New CMS Modules	89
CP Module Splits	89
New CP Copy Files	89
New CP Macro	89
New CP Modules	89
Index	90

Figures

1. The VM/SP Library	iv
2. The VM/SP Library Chart	13
3. Cover Design	15
4. PF Key Changes	38
5. PF Key Changes	73
6. CP Module Splits	89

This part of the book introduces you to VM/SP Release 3. Its three sections:

- Give an **overview** of the new facilities and enhancements
- Explain the **environmental considerations** that must be understood to install VM/SP Release 3
- Describe how the **VM/SP library** has changed.

An Overview of VM/SP Release 3

This section gives you an overview of VM/SP Release 3. It:

- Introduces **significant additions** to VM/SP Release 3
 - Summarizes the **other changes and enhancements** to the facilities in VM/SP Release 3.
-

Significant Additions

Significant additions to VM/SP Release 3 are:

- Support for **Structured Query Language/Data System (SQL/DS)**
 - Addition of the **System Product Interpreter**
 - Enhancements to the **Programmable Operator Facility**
 - Enhancements to the **System Product Editor**
-

The VM/SP Release 3 licensed program is a new release of the VM/System Product (VM/SP). Release 3 contains all the functions currently available in Release 2 plus many new or enhanced capabilities.

Significant additions to the basic functions provided by VM/SP include the following new and enhanced facilities:

Support for Structured Query Language/Data System (SQL/DS)

Structured Query Language/Data System (SQL/DS) is a data base licensed program that provides relational data base facilities for customers running on VM/SP Release 3 or VSE/Advanced Functions. CP and CMS enhancements have been included to allow SQL/DS Release 2 to operate on VM/SP Release 3 without a guest VSE operating system and without VSE/VSAM. These enhancements have been provided in a generalized fashion so that other applications may take advantage of them.

SQL/DS supports many different application and user needs. Its interactive query, report writing, and application development facilities are usable by both DP professionals and novice end-users. It addresses environments such as planning and prototyping where data structure and application requirements are subject to frequent change. It provides data base management facilities such as data sharing, locking, security, authorization, and recovery.

You can find more information in "Support for SQL/DS" on page 22.

Addition of the System Product Interpreter

The System Product Interpreter is a new and powerful command and macro processor. It can be used for:

- writing personal applications
- processing and generating commands
- creating macros such as those used by the System Product Editor
- solving mathematical problems as a desk calculator
- solving any other problem for which an interpreted simple-to-use language is suitable.

The syntax of its language is a high-level form that is easy to learn.

You can find more information in “Addition of the VM/SP System Product Interpreter” on page 24.

Enhancements to the Programmable Operator Facility

The Programmable Operator Facility is designed to help increase the efficiency of system operation and to allow remote operation of systems in a distributed data processing environment.

The Release 3 programmable operator, as in Release 2, can operate in either end of the network: the distributed end or the host end. In Release 2 the programmable operator was not able to recognize that communication with another system was failing. In Release 3, there are means to detect and correct this system failure.

The Release 3 programmable operator has been enhanced to:

- route messages to different users in addition to the logical operator
- alert the host operator of the inability to communicate with a remote system and vice versa
- be more flexible in comparing the text of messages in the routing table entries
- accept EXEC action routines written in languages other than the Basic Assembler language
- provide additional message recording and error handling (logging).

You can find more detailed information in “Enhancements to the Programmable Operator Facility” on page 27.

Enhancements to the System Product Editor

The System Product Editor has numerous new or improved functions. The primary emphasis of this added support is to:

- add color and extended highlighting
- tailor the screen with more flexibility
- control screen layout
- split the screen vertically
- extend use of the prefix area
- improve flexibility in editing selected lines.

You can find more detailed information in “Enhancements to the System Product Editor” on page 30.

Other enhancements to VM/SP are as follows:

Summary of Other Facilities and Enhancements

This list summarizes the other facilities and enhancements.

- Enhanced Serviceability - RAS
 - Support of System/370 Program Event Recording (PER)
 - Symptom record
 - DMSDOS split
 - DMSQRY split
- CP Enhancements
 - Enhancements for Application Programs
 - Enhancements to the Inter-User Communication Vehicle (IUCV)
 - Addition of CP DASD Block I/O System Service
 - Enhancements for Improved CP Performance
 - Nucleus reduction
 - Enhancement to SET QDROP
 - Support for larger-than-4K buffer terminals
 - Other CP Enhancements
 - DIAGNOSE code X'00'
 - DIAGNOSE code X'8C'
 - DIAGNOSE code X'14'
- CMS Enhancements
 - Enhancements for Application Programs
 - CMS support of IUCV
 - Wait on ECB
 - End-of-command nucleus extensions
 - ABEND exit
 - Reading directly from virtual machine console
 - Creating your own immediate commands
 - QSAM tape end-of-volume exit
 - Support of CP DASD Block I/O System Service
 - Enhancements to CMS File System
 - 512-byte blocksize minidisks
 - RESERVE command
 - Update-in-place
 - Enhancements for Improved CMS Performance
 - LISTFILE and RENAME migration
 - Reduction of SCBLOCK look-aside buffer
 - Removal of the CMSSEG segment
 - Enhancement of GLOBALV
 - Other CMS Enhancements
 - Enhancements to TELL and DEFAULTS commands
 - EXECOS command
 - SET EXECRAC command

- **Enhanced Support of VSAM**
 - Support of VSE/VSAM Release 3
 - CATCHCHECK command
 - Support for VSAM Assembler language macros
 - Support for VSE/VSAM macros
 - Support for OS/VSAM macros
 - VSEVSAM command
 - Support for 3380 Direct Access Storage

- **Hardware Support**
 - Printer support
 - 3262 Printer, Model 5
 - 3800 Printing Subsystem, Models 3 and 8
 - 4245 Line printer
 - 4250 Printer
 - Tape support
 - 3430 Magnetic Tape Subsystem
 - Communications Unit
 - 3088 Multisystem Channel Communication Unit

- **Previously Available Functions**
 - Support for the missing interrupt handler
 - Support for the speed matching buffer for the IBM 3375

You can find additional information about these enhancements in “Part Two: New Facilities and Enhancements” on page 21.

Environmental Considerations

This section describes the **environmental considerations** that must be satisfied to install VM/SP Release 3:

- **Programming Requirements**
 - **Statement of direction for the System Product Interpreter**
 - **Statement of direction for the System Product Editor**
 - **Integrity Statement**
-

Programming Requirements

VM/SP Release 3 requires certain release levels of related programs. Discussion follows for:

- **VSE/VSAM Release 3**
 - **EREP Release 2.3**
 - **IPCS/E Release 2**
 - **SQL/DS Release 2**
-

VSE/VSAM Release 3

VM/SP Release 3 supports VSE/VSAM Release 3. VSE/VSAM Release 2 is not supported by VM/SP Release 3.

EREP Release 2.3

Environmental Recording, Editing and Printing Program 2.3 (EREP)

VM/SP Release 3 contains changes that support Release 2.3 of the Environmental Recording, Editing and Printing Program. The new CPEREP commands are described in *VM/SP OLTSEP and Error Recording Guide*.

EREP 2.3 is no longer shipped on the VM/SP product tape. EREP 2.3 is a unique product offering shipped on a separate tape.

IPCS/E Release 2

VM/Interactive Problem Control System Extension (IPCS/E) Release 2

VM/SP Release 3 continues to support the Release 2 level of VM/Interactive Problem Control System Extension program product. VM/SP Release 3 is required to use the enhanced symptom record. Refer to the VM/SP Release 3 Program Directory for additional information.

This is a separate program product and is shipped on a separate tape.

SQL/DS Release 2

Structured Query Language/Data System Release 2

Structured Query Language/Data System (SQL/DS) Release 2 operates on VM/SP Release 3. SQL/DS can operate on VM/SP Release 3 without a guest VSE operating system and without VSE/VSAM.

Statement of Direction

There is a special direction for two products:

- **System Product Interpreter**
 - **System Product Editor**
-

System Product Interpreter

The System Product Interpreter will be the primary interpretive command and macro processor for VM/SP in the future. It is IBM's intent to provide all future VM/SP interpretive command and macro processing enhancements to the new System Product Interpreter. EXEC and EXEC 2 will continue to be supported and coexist with the System Product Interpreter.

System Product Editor

The System Product Editor (invoked by the XEDIT command) is the primary VM/SP editor. It is IBM's intent to provide all future VM/SP editing enhancements to this editor. The System Product Editor has attempted to provide the most useful capabilities available in various editors.

A compatibility function for Display Editing System (EDGAR) was provided in previous VM/SP releases to assist EDGAR users in the transition to the System Product Editor. Its primary purpose was for migration only. A subset of the EDGAR functions was provided to aid in migration. **This EDGAR compatibility support is not included in VM/SP Release 3.** All those using the EDGAR migration support **must** migrate to the System Product Editor before installing VM/SP Release 3. In VM/SP Release 3, System Product Editor users cannot use the EDGAR migration functions.

Statement of System Integrity

System Integrity

An operating system is said to have system integrity when it is designed, implemented, and maintained to protect itself against unauthorized access. It does this to the extent that security controls specified for that system cannot be compromised. Specifically for VM/SP, system integrity is defined as the inability of any program running in a virtual machine not authorized by a VM/SP Control Program (CP) mechanism under the customer's control, and/or a guest operating system mechanism under the customer's control, to:

1. circumvent or disable main or secondary storage protection provided by the control program (CP)
2. access a resource that is password-protected by the control program (CP)
3. obtain control in real supervisor state, or with privilege class authority or directory capabilities greater than those it was assigned
4. circumvent the system integrity of any guest operating system which itself has system integrity (i.e., MVS or VM/SP) as a result of an operation by any VM/SP control program facility.

The following definitions apply to the above:

main storage protection: refers to the isolation of one virtual machine from another achieved by the control program (CP) through hardware dynamic address translation (DAT).

secondary storage protection: refers to the disk extent isolation implemented for minidisks/virtual disks for the control program (CP) through channel program translation.

password protected resource: refers to resources protected by control program (CP) logon passwords and minidisk passwords.

directory capabilities: refers to those directory options that control the use of functions intended to be restricted by specific assignment, for example, options such as those that permit system integrity controls to be bypassed or those not intended to be generally granted to users.

guest operating system: refers to a system control program that operates under the VM/SP control program.

VM/SP system integrity applies to the following environments for MVS guest machines only:

- V=R with the NOTRANS option
- V=R with the shadow-table-bypass SET command option
- the preferred machine assist
- the single processor mode

However, when any of these facilities is used within an MVS guest machine, an MVS user or program that has been given authority to bypass MVS system integrity controls may also be able to bypass the system integrity controls built into VM/SP. In these circumstances, it is the installation's responsibility to assure that the required MVS system integrity controls are installed, and that programs and users authorized for the MVS guest machine are properly controlled.

VM/SP control program system integrity does not include the protection of data between multiple users of a single CMS batch system, nor does it apply to virtual machines using non-disruptive transition (NDT) support.

Responsibilities of the Installation

Protection of the installation's resources remains the responsibility of the installation itself. For system integrity to be meaningful, proper use of security controls is essential.

Some areas for which effective controls should be implemented are:

- password protection
- assignment of appropriate privilege classes
- assignment of directory options
- set up and authorization of guest virtual machines

Particular actions and restrictions may vary, depending on the system configuration or environment. The installation is responsible for the selection, application, adequacy, and implementation of these actions and restrictions, and for appropriate application controls.

Additional information pertaining to the use of security controls can be found by consulting the *VM/SP Library Guide and Master Index*.

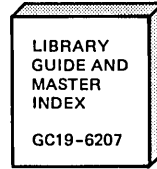
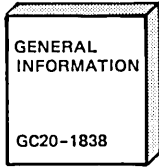
The VM/SP Library

The VM/SP library has been revised for this release. The library has a new organization and a new look. This section describes the new organization, the new look, the new books, and the revised books in the library.

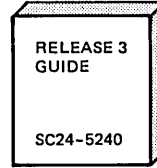
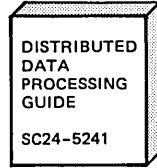
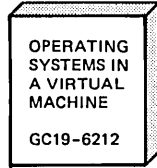
- **The VM/SP library chart**
 - **New look**
 - **New books**
 - **Revised and restructured books**
-

The VM/SP Library

Evaluation



Planning



Installation



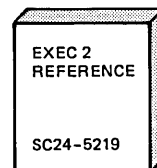
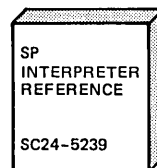
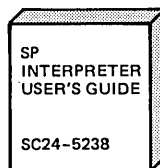
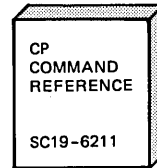
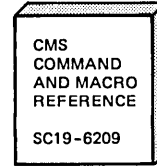
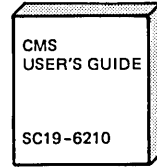
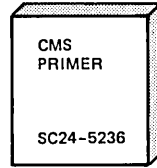
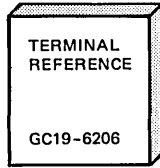
Administration



Operation



End Use



Reference Summaries

To order all the Reference Summaries, use order number SBOF 3820.

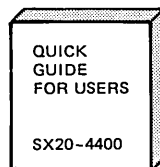
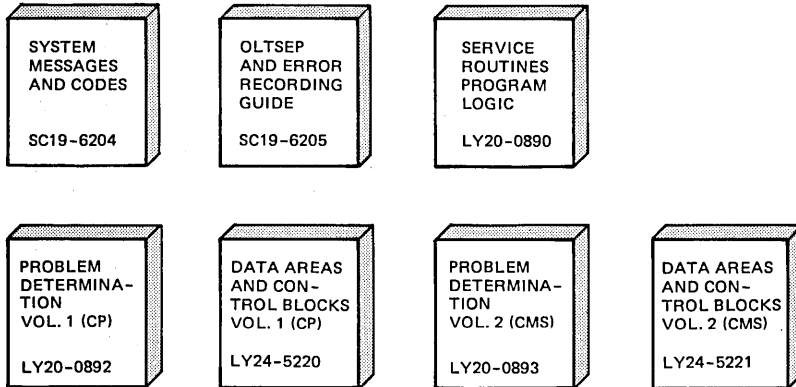
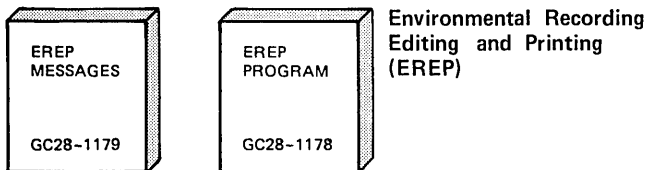
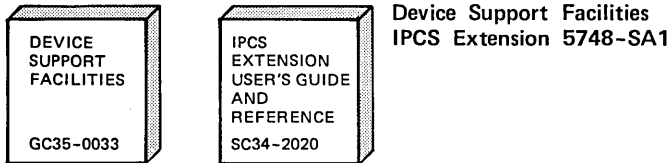


Figure 2 (Part 1 of 2). The VM/SP Library Chart

Program Service



Auxiliary Service Support



Auxiliary Communication Support

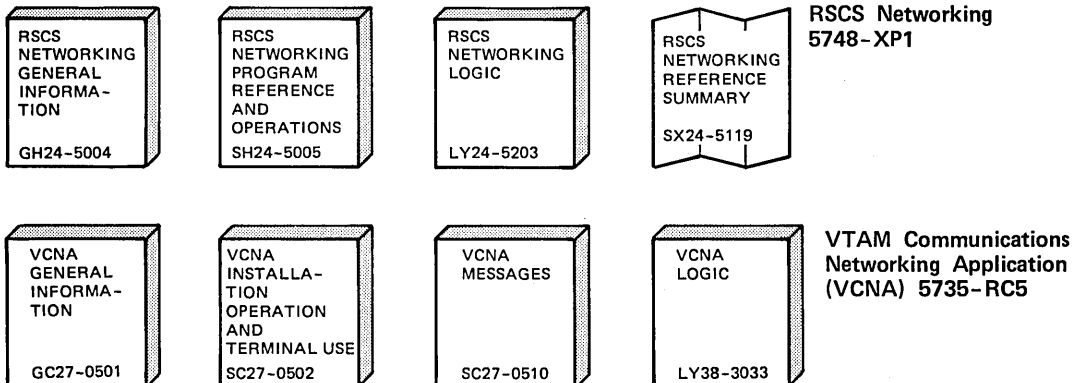


Figure 2 (Part 2 of 2). The VM/SP Library Chart

New Look

The VM/SP library has a new look.

Beginning with Release 3, the VM/SP library has a new appearance:

- The books have a new **cover design**, with a graphic that is printed in various colors according to the major task associated with a particular book.
- New, coordinating **binders** are available. Binders and labels of book titles that suggest a library organization can be ordered with a set of manuals (using a **Bill of Forms**), or the binders and labels can be ordered separately. See the following page for order numbers.

Cover Design

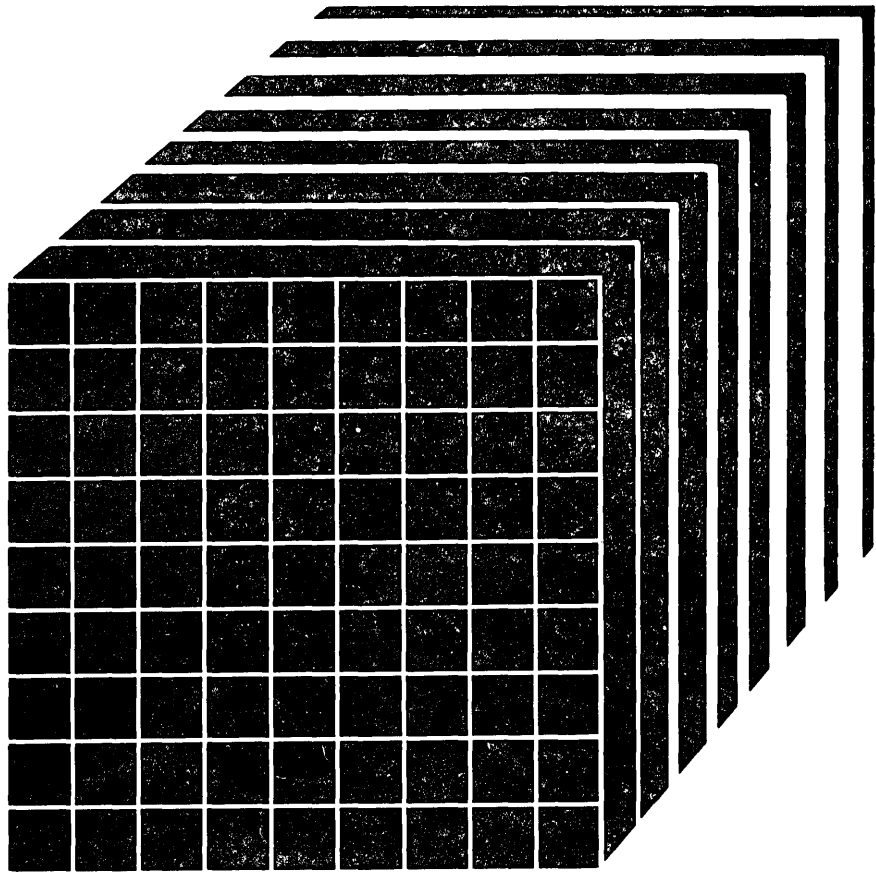


Figure 3. Cover Design

The graphic design on the covers and binders represents the system image of VM. The grid represents multiple virtual machines within a real machine. The radiating lines symbolize VM as an expanding system that offers virtually unlimited resources and uses.

The design is printed in different colors on the book covers. The color represents the major task associated with the publication. Tasks are described in the *VM/SP Library Guide and Master Index*. The colors and tasks are as follows:

Blue	Evaluation
Green	Planning
Red	Installation
Purple	Diagnosis/Program Service
Orange	Customization
Yellow	Operation
Cranberry	End Use
Turquoise	Administration
Pink	Application Programming

Bill of Forms for Easy Ordering

For easy ordering there is one Bill of Forms number, SBOF-3204, needed to order the entire set of:

eight binders
one set of labels
21 publications

Or, they can be ordered separately:

- SX24-5129 - one binder
- SX24-5131 - one set of labels
- Publications:
 1. GC19-6200 - *VM/SP Introduction*
 2. GC19-6206 - *VM/SP Terminal Reference*
 3. GC19-6207 - *VM/SP Library Guide and Master Index*
 4. SC19-6201 - *VM/SP Planning Guide and Reference*
 5. SC19-6202 - *VM/SP Operator's Guide*
 6. SC19-6203 - *VM/SP System Programmer's Guide*
 7. SC19-6204 - *VM/SP System Messages and Codes*
 8. SC19-6205 - *VM/SP OLTSEP and Error Recording Guide*
 9. SC19-6209 - *VM/SP CMS Command and Macro Reference*
 10. SC19-6210 - *VM/SP CMS User's Guide*

11. SC19-6211 - *VM/SP CP Command Reference for General Users*
12. GC19-6212 - *VM/SP Operating Systems in a Virtual Machine*
13. SC24-5219 - *VM/SP EXEC 2 Reference*
14. SC24-5220 - *VM/SP System Product Editor User's Guide*
15. SC24-5221 - *VM/SP System Product Editor Command and Macro Reference*
16. SC24-5236 - *VM/SP CMS Primer*
17. SC24-5237 - *VM/SP Installation Guide*
18. SC24-5238 - *VM/SP System Product Interpreter User's Guide*
19. SC24-5239 - *VM/SP System Product Interpreter Reference*
20. SC24-5240 - *VM/SP Release 3 Guide*
21. SC24-5241 - *VM/SP Distributed Data Processing Guide*

For additional VM/SP library information, including a description of tasks, see the *VM/SP Library Guide and Master Index*.

New Books

Several new books have been added to the VM/SP library.

VM/SP Distributed Data Processing Guide, SC24-5241

This book defines the concept of distributed data processing and describes its functional requirements. It lists the advantages of distributed data processing and describes the products used with VM/SP to accomplish it.

VM/SP Installation Guide, SC24-5237

This manual provides procedural information about the VM/SP installation process. It is published expressly to satisfy the requirements associated with the installation and service application tasks. It is based on the information contained in Parts 3, 4, and 5 of the former *VM/SP Planning and System Generation* book (VM/SP Release 2 version), the *System IPO/Extended Planning Guide*, GC20-1874, and the *Universal Program Directory for VM/SP System IPO/E*. This publication provides one central source of installation and servicing instructions for all VM/SP users. It includes procedures for updating VM/SP. It contains flowcharts showing the installation steps to be followed, based on the user's current system level and the target system desired. It also includes procedures for installing the Emulation Program for the 3704/3705 communications controller.

VM/SP Release 3 Guide, SC24-5240

This book provides current users of VM/SP Release 2 with a synopsis of the functional enhancements offered by the new release. It will improve the installation time of Release 3 and improve customer productivity by describing new functional enhancements, defining the related user interfaces, and giving examples of their use. It includes details for migrating from VM/SP Release 2 to VM/SP Release 3 and identifies new and changed modules.

VM/SP System Product Interpreter User's Guide, SC24-5238

This manual is a step-by-step guide to using the System Product Interpreter, the new and powerful interpretive command and macro language. It is 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 that has been added to the *VM/SP CMS Primer* for prerequisite knowledge.

VM/SP System Product Interpreter Reference, SC24-5239

This book 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.)

Restructured books

Several books have been revised or restructured.

VM/SP CMS User's Guide, SC19-6210

This book contains a new chapter on communicating with other computer users. The discussion on what you can do with CMS commands describes several tasks and lists the commands used to perform these tasks.

The chapters have been reordered. Usage information on the CMS Editor and the CMS EXEC Facility is now in the appendix. Examples on using the System Product Editor and the System Product Interpreter are provided.

VM/SP Data Areas and Control Block Logic

This book has been split into two volumes to make information easier to find. The original number was LY20-0891. The two new books are:

- *VM/SP Data Areas and Control Block Logic, Volume 1 (CP)*, LY24-5220
- *VM/SP Data Areas and Control Block Logic, Volume 2 (CMS)*, LY24-5221.

VM/SP General Information, GC20-1838

This manual, previously titled *General Information Manual*, GC20-1838, has been generalized to remove release-sensitive material now found in the *VM/SP Release 3 Guide*. This manual introduces and describes the features of VM/SP, and provides customer management and technical staff with information needed to evaluate the applicability of VM/SP to their installations. It contains information needed for a basic understanding of using, programming, and installing VM/SP.

VM/SP Introduction, GC19-6200

This book has been revised to serve as an extension to 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 book 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. Topics are discussed only to a depth needed to introduce VM/SP. This book is suggested as a prerequisite before reading the more detailed reference books in the VM/SP library.

VM/SP Planning Guide and Reference, SC19-6201

This manual, previously titled *VM/SP Planning and System Generation Guide*, SC19-6201, has been revised to better meet the requirements of the planning task. Conceptual information was removed and now appears in *VM/SP Introduction*. Parts 3, 4, and 5 were removed and now appear in *VM/SP Installation Guide*.

VM/SP Terminal Reference, GC19-6206

This book, previously titled *VM/SP Terminal User's Guide*, is completely reorganized and revised for ease of use. It is intended for those who plan to use terminals with VM/SP in their operations. It discusses the characteristics of terminals in general and the physical characteristics of different terminals and consoles. It contains a typical session that gives working examples from logon to logoff.

This part of the book describes the **functional changes** introduced in Release 3. Each section describes **details** for one functional change. The details are given for:

- SQL/DS
- VM/SP System Product Interpreter
- Programmable Operator Facility
- System Product Editor
- Enhanced Serviceability
- CP Enhancements
- CMS Enhancements
- Enhanced VSE/VSAM Support
- Previously Available Functions
- Hardware Support

Structured Query Language/Data System (SQL/DS) is a data retrieval and data handling facility that provides access to data from interactive and programming environments.

SQL/DS Release 2

Structured Query Language/Data System (SQL/DS) is a full-function data base management system with a relational data base at its core. While SQL/DS provides end-user query and report writing capabilities, its facilities for managing data include functions normally found in a data base management system. This is especially useful for application programmers.

Access to data stored in SQL/DS is done through a data language called the Structured Query Language (SQL). SQL is a simple, easy-to-use query language with English commands. It is especially useful for end-users. Because of its simplicity and the automatic access features provided by SQL/DS, no advanced planning is required for retrieving data from a SQL/DS data base. SQL commands can be entered directly from display terminals or embedded in application programs written in COBOL, PL/I, VS FORTRAN, or Assembler language.

Specifically, SQL/DS supports:

1. **A logical data model**
SQL/DS supports a *relational* model of data. This logical data model support insulates applications and users from the physical storage of the data. Data storage can be reorganized without affecting applications that access the data.
2. **Sharing data among multiple users and applications**
SQL/DS allows multiple users to access the same data at the same time. SQL/DS has locking mechanisms that prevent conflicting updates from different users.
3. **Application programming support**
SQL/DS supports application program access to data managed by SQL/DS. Such programs may be written in PL/I, COBOL, VS FORTRAN, or Assembler language.
4. **Data Base Services (DBS) Utility**
The Data Base Services (DBS) Utility supports a variety of data functions useful in an interactive environment. It can be used for file maintenance and reporting operations. This includes loading tables from interactive files, or unloading tables to interactive files for use with other tools (such as editors or other data processing programs). It can also reorganize data, extend the data base, and recover the data base. The DBS Utility is especially useful for system programmers.

5. Data recovery
SQL/DS can recover data if there is an abnormal ending of a program, a DASD failure, or a system failure. These recovery facilities include dynamic backout, logging, and archiving.
6. Data security
Data stored under SQL/DS can be protected from unauthorized access. SQL/DS provides security facilities to control access to SQL/DS and to data managed by SQL/DS.
7. Data integrity
SQL/DS protects data from inadvertent or intentional damage by other users.
8. System catalogs
SQL/DS automatically maintains data definitions, relationships between data, control information, user descriptions, and other information and statistics in a set of system catalogs. These catalogs are stored as data base tables.
9. Interactive SQL (ISQL)
Authorized CMS users can access data stored in a data base using the Interactive SQL (ISQL) component of SQL/DS. ISQL runs as a CMS application in multiple user mode.

VM/SP Support for SQL/DS

CP and CMS changes and enhancements have been included in VM/SP Release 3 to allow SQL/DS to operate on VM/SP without either a guest VSE operating system or VSE/VSAM. These enhancements are provided in a general manner so that other applications may take advantage of them.

“CP Enhancements” on page 47 and “CMS Enhancements” on page 52 describe this support in detail.

Addition of the VM/SP System Product Interpreter

The **System Product Interpreter** is an interpretive command and macro processor. It may help you increase productivity with the creation and execution of EXEC-type command procedures and programs. Its functions are easy to learn and are written in a high-level language, known as the Restructured Extended Executor (REXX) language.

The System Product Interpreter may be used advantageously by everyone from new users to experienced programmers. Novices may use the System Product Interpreter to easily create and execute EXEC-type command procedures and programs. Experienced programmers may use the System Product Interpreter to increase their productivity.

Advantages of the System Product Interpreter

Some advantages of using the System Product Interpreter are:

- It is easy to learn and use.
- It uses a general-purpose, high-level language, the Restructured Extended Executor (REXX) language, much like that used by PL/I and other high-level programming languages.
- It supports structured programming concepts such as:
 - If-Then-Else
 - Select-When-Otherwise-End
 - Do-Iterative-End
 - Iterate-and-Leave
- It has many built-in functions that you can use in your programs. These include character manipulation, conversion, and information about functions. These are similar to the built-in functions in PL/I, such as INDEX, which returns the position of one string in another.
- It accepts programs written in mixed case (which makes them easier to read.)
- It has extensive mathematical capabilities, such as decimal, exponential, and scientific arithmetic. (You can even use it as a desk calculator.)
- It uses a full set of arithmetic, character, and logical operators with algebraic precedence and parentheses.
- It accepts programs written in free format which allow:
 - multiple lines per statement
 - multiple statements per line
 - comments on a statement line
- It imposes no limit (except your virtual storage size) to the length of variable data. Symbols may be up to 250 characters in length.

- It has built-in string parsing instructions. The instructions ARG, PARSE, and PULL have the capability to parse (split up) a selected string into variables.
- It supports interactive debugging with the TRACE instruction to see how a particular instruction is interpreted.
- It enhances the maintainability of XEDIT macros and system EXECs. It easily handles subroutine calls to other EXECs, modules, or internal routines.

The System Product Interpreter coexists with the CMS EXEC and EXEC 2 processors. It is functionally a superset of CMS EXEC and EXEC 2, but uses a completely different language and syntax.

Examples of System Product Interpreter EXECs

Example One

```

/* Sample desk calculator for expressions in REXX format          */
    parse arg expression                                         /* Get REXX expression */
    signal on syntax                                             /* Set up for possible */
                                                                /* syntax error       */
    interpret 'answer =' expression                             /* Evaluate expression */
    say 'Answer =' answer                                       /* Display answer     */
    exit
/* Control passes to here if the expression causes a syntax error*/
SYNTAX:
    say "'expression'" is not quite valid syntax. Try again!'
    exit
/* Sample input:  2 ** 3           Sample answer:  8           */

```

Example Two

```

/* This REXX EXEC checks console input and returns a response.  */
say 'Please answer yes or no.'                                  /* Asks for input     */
do until flag='done'                                         /* Main loop          */
    parse pull answer                                         /* Gets input         */
    if (answer='yes' | answer='no') then                       /* Checks for valid answer */
        do
            say 'OK'                                           /* Answer is OK      */
            flag='done'
        end
    else
        say 'I don't know. Try again.'                         /* Answer isn't OK   */
end
exit

```

Example Three

```
/* This REXX EXEC is a prime number generator. */
parse arg stop /* Any limit given? */
if stop='' then stop=100 /* No, just do it 100 times */
count=0
flags.=1
do i=0 while count<stop /* OK, loop until we have enough */
  if ~flags.i then iterate /* Have we seen this one yet? */
  prime=i+i+3 /* No, then I+I+3 is a prime */
  /* number */
  say prime /* Type it out */
  do k=i+prime to stop*5 by prime /* Eliminate some numbers */
    flags.k=0
  end
  count=count+1 /* Bump the count of numbers */
end i
say count 'primes' /* And say how many we have */
exit /* before...we say farewell.... */
```

You can find complete information about the System Product Interpreter and REXX in the *VM/SP System Product Interpreter User's Guide* and *VM/SP System Product Interpreter Reference*.

Enhancements to the Programmable Operator Facility

Several enhancements have been added to the **Programmable Operator Facility**. Among them are:

- **Message routing with nicknames**
 - **Availability enhancements**
 - **Enhanced text comparison**
 - **EXEC action routines**
 - **Enhancements to message recording and error handling (logging)**
-

The Programmable Operator Facility is designed to increase the efficiency of system operation and to allow remote operation for systems in a distributed data processing environment. It intercepts all messages/requests directed to its virtual machine and handles them according to preprogrammed actions. It determines whether a message is recorded for future reference, and whether the message is acted upon or sent on to the logical operator to handle.

To ensure that the programmable operator will function properly, a user (a virtual machine other than the programmable operator virtual machine on the local system or in the distributed system) is identified to the programmable operator to receive messages. This user is called the **logical operator**, as opposed to the CP system operator. This logical operator may be in the same system as the programmable operator or in a separate system, called a **HOST** system.

Changes have been made in VM/SP Release 3 to provide:

- Greater flexibility in the routing of messages
- Reduced system programmer time for programmable operator installation
- Improved usability for a host operator with multiple distributed systems
- Improved availability of the programmable operator
- Recognition of the inability to communicate between host and distributed systems.

Message Routing with Nicknames

The programmable operator routes messages to different users, other than the logical operator, or to more than one user at a time, via the CMS TELL command. For example, this provides the ability to have SPOOL and/or TAPE operators at the remote site besides the system operator at the host site.

Messages are routed to users specified by a nickname from the “userid NAMES” file or by a **userid**. If a user is on another system, identification must be through a nickname. A message longer than 94 characters will be split and sent as multiple messages.

Note: To use the CMS TELL command, the user must have access to the SYSTEM NETID file.

Availability Enhancements

The programmable operator allows periodic signaling between a host and the distributed system. It alerts the host operator of the inability to communicate with a remote system and vice versa. The operator at the host site also has the capability to send commands to the programmable operator facility to control its operation, as well as commands to execute on the distributed system.

If a CMS ABEND occurs while the programmable operator facility is running, all files will be closed and ABEND error messages will be sent to the logical operator. A dump of the virtual machine storage will be taken using the CP VMDUMP command and the last system or device that was IPLed will be re-IPLed. If a CMS ABEND occurs while an action routine is running, ABEND error messages will be sent to the logical operator and the requester (if any). Control will be returned to the point in the programmable operator facility immediately following the action routine call.

Enhanced Text Comparison

The programmable operator routing table is a CMS file that contains the information used to control the operation of the programmable operator. The routing table enables the programmable operator to recognize a message as a command, to determine the action to take when a message comes in, and to recognize the authorized users of programmable operator functions.

Enhanced text comparison allows multiple text comparison strings in the text field of a single routing table entry. If the text field is left blank, any text compared with this field is considered a match. When multiple texts are specified, intervening blank or non-blank characters in messages can be skipped. Also, logical operations may be done on those texts. Therefore, it is easier to construct routing table entries and to match incoming messages, and thus tailor the programmable operator to meet the needs of the installation.

EXEC Action Routines

Action routine writing is made easier with EXECs. In Release 2, action routines had to be written in Basic Assembler language. Presently, you may write action routines as EXECs in System Product Interpreter, EXEC 2, and CMS EXEC languages.

Enhancements to Message Recording and Error Handling (logging)

- Provides additional information in the log recording as a result of log commands.

SET LOGGING provides the message sender with the ability to control the logging level: no logging, logging incoming messages and special PROP messages, or incoming messages plus response messages. LOGGING now provides additional information and is an alternative to spooling your console.

- Allows the installation to handle error conditions which occur when writing to the log, for example DISK FULL.

If the disk is full it may not be possible to write another record to the programmable operator facility log file. If this happens, a user-written EXEC can be invoked to do the recovery. The user EXEC must have the filename of PROPLGER. After the error action is taken, the programmable operator facility will continue to operate.

Note: The routing table format is changed from the format used by the programmable operator facility in VM/SP Release 2. The new format makes the specifications easier and the information clearer. The Release 2 format is not compatible with the Release 3 format. You **must** convert the routing table to reflect this change. IBM provides a conversion EXEC, PROPRTCV, to assist you in the conversion. The PROPRTCV EXEC converts Release 2 format routing tables to Release 3 format routing tables. Using a Release 2 RTABLE as input, PROPRTCV creates a new RTABLE, leaving the Release 2 one unchanged. If you choose not to use the conversion EXEC, you must regenerate the routing table by hand.

You can find additional information about the programmable operator facility in the *VM/SP Operator's Guide* and *VM/SP System Programmer's Guide*.

Enhancements to the System Product Editor

Enhancements to the **System Product Editor** provide new or improved support in the following areas:

- **Color** and extended highlighting
 - **Prefix macros**
 - **Selective line** editing
 - **Screen layout** control, including vertical screen split
 - **PF key** changes
 - Several new functions
 - Modifications to miscellaneous existing functions
-

The primary emphasis of these **System Product Editor** (invoked by the XEDIT command) enhancements is to increase flexibility in tailoring the screen by adding color and extended highlighting. Editing flexibility is enhanced by controlling screen layout and capitalizing on the use of the prefix area by supporting prefix macros.

Color and Extended Highlighting

The new **COLOR** option on the SET subcommand allows you to associate various colors and extended highlighting features (for example, blinking, underlining, and reverse video) with fields on the XEDIT screen.

The format of the SET COLOR subcommand is:

```
[SET] COLOR field [color] [exthi] [High|Nohigh]
          *
```

where:

field can be any of the following fields:

Arrow	Scale	SHadow	Cmdline
CUrline	Tabline	STatarea	Pending
PRefix	Idline	TOfeof	Msgline
Filearea			

color can be any of the following colors:

Blue	Green	White	Yellow
Red	Turquoise	Pink	Default

exthi can be any of the following features:

BLInk	Underline
REVvideo	NONE - no extended highlighting

For example:

```
SET COLOR ARROW BLUE BLINK
```

displays a blinking blue command line arrow.

Although any valid operands are accepted, the result depends on what features are available on your terminal.

The current **COLOR** setting can be displayed by **QUERY** and returned (to a macro) by **EXTRACT**, a new subcommand (described under “Additional New Functions” on page 38.)

The **SET RESERVED** and **SET CTLCHAR** subcommands are also enhanced to allow you to specify colors and extended highlighting for reserved lines and parts of a reserved line.

Prefix Macro Support

Prefix macro support greatly extends the use of the prefix area. Four **XEDIT** prefix macros are added to this release, as well as the ability to write your own prefix macros.

Prefix Macros

The following prefix macros are provided:

- | | |
|---------------------------|---|
| < (Shift Left) | Shift a line(s) to the left for one or more columns. This is functionally equivalent to the SHIFT LEFT subcommand. |
| > (Shift Right) | Shift a line(s) to the right for one or more columns. This is functionally equivalent to the SHIFT RIGHT subcommand. |
| X (Exclude) | Exclude line(s) from display and from the scope of editing subcommands (see also “Selective Line Editing” following). |
| S (Show) | Re-display excluded lines. |

Writing Prefix Macros

You can also write your own prefix macros for a variety of purposes, from performing a function from the prefix area that is normally accomplished by entering a subcommand on the command line, to creating an entirely new function.

PENDING is a new **SET** option added in support of prefix macros. It is used to control the execution of a prefix macro and the status of the screen while the prefix macro is being executed. Designed to be issued from a macro, **SET PENDING** can be used to display a prefix subcommand or macro in the prefix area of the current line, or to indicate that a prefix subcommand or macro was the beginning or end of a block. In both cases, the screen is placed in a pending status. **SET PENDING** can also be used to notify the user that a prefix macro was entered incorrectly.

The pending condition is indicated by the following message in the status area on display terminals:

```
'value' pending...
```

where “value” is the name of the prefix subcommand or macro.

The format of the SET PENDING subcommand is:

```
[SET] PENDING ON|BLOCK|ERROR string  
OFF
```

For example:

```
SET PENDING ON A
```

adds an entry to the “pending list” (list of prefix subcommands and macros that have not yet been executed), displays A==== in the prefix area of the current line, and displays the notice 'A' pending... in the status area.

A new operand on the SET PREFIX subcommand allows you to QUERY PREFIX SYNONYM to find out what you have already SET. SET PREFIX SYNONYM subcommands would normally be added to the PROFILE XEDIT file.

Synonyms can be assigned for the following reasons:

- A prefix macro filename can be up to eight characters long, but the prefix area is only five positions long. You can use SET PREFIX SYNONYM to assign a synonym that is up to five characters long.
- The synonym can be a special character that is not permitted as part of a CMS filename. For example, the filename for the XEDIT prefix macro > is PRFSHIFT.
- A macro can perform different functions, depending on how it is entered. Different synonyms can signify different functions to the macro. For example, the XEDIT prefix macro shifts the screen right if > is entered and left if < is entered. The “block” function is performed if << or >> is entered. The synonyms assigned to this macro are:

```
SET PREFIX SYNONYM > PRFSHIFT  
SET PREFIX SYNONYM < PRFSHIFT  
SET PREFIX SYNONYM >> PRFSHIFT  
SET PREFIX SYNONYM << PRFSHIFT
```

To determine what prefix macro synonyms are in effect, use the QUERY PREFIX SYNONYM subcommand:

```
Query PREFIX Synonym name|*
```

For information on writing prefix macros, refer to *VM/SP System Product Editor User's Guide*, Chapter 7. See also SET/QUERY/EXTRACT PENDING, SET PREFIX, and “Section 4: Prefix Subcommands and Macros” in the *VM/SP System Product Editor Command and Macro Reference*.

LPREFIX Subcommand

The LPREFIX subcommand allows you to simulate writing in the prefix area of the current line on a display terminal. It can be used on typewriter terminals to use some of the features of prefix subcommands and macros, as well as on display terminals.

The format of the LPREFIX subcommand is:

```
LPrefix [text]
```

where “text” specifies up to five characters.

Selective Line Editing

“Selective line editing” is the ability to edit a specified collection of lines, while excluding other lines from the display.

The following new macro and SET subcommand options are used to perform selective line editing.

- ALL macro
- SET SELECT
- SET DISPLAY
- SET SCOPE
- SET SHADOW

The X prefix macro (see “Prefix Macros” on page 31) is also used for selective line editing.

ALL Macro

The ALL macro is used to display a specified collection of lines for editing, while excluding other lines from display. The collection is specified by a target that is repeatedly applied to the entire file, starting at the top of the file.

The format of the ALL macro is:

```
ALL [rtarget]
```

where “rtarget” (repeated target) is a target that defines which lines are displayed. The target is “repeated,” that is, it is applied from the top of the file for as many times as necessary to collect all the lines in the file that correspond to it.

For example:

```
ALL/DAN/
```

displays all lines in the file that contain the string “DAN”.

```
ALL +2
```

displays every other line in the file.

Issuing ALL with no operand displays the entire file.

SET SELECT, SET DISPLAY, SET SCOPE, SET SHADOW

The SET subcommand options SELECT, DISPLAY, SCOPE, and SHADOW can be used in a macro to perform selective line editing.

These four SET options work together in the following manner. You can use SELECT to assign a selection level to one or more lines. Lines having the same selection level are logically grouped. DISPLAY is used to specify which selection levels of lines are displayed. With DISPLAY, you can specify that only one selection level of lines be displayed (DISPLAY n1), or you can specify that a range

of selection levels (DISPLAY n1 n2) be displayed. SCOPE defines the set of lines that the editor can act upon. It can be used to specify that editing operations affect the entire file (SCOPE ALL), or it can be used to restrict editing operations to only the displayed lines (SCOPE DISPLAY). SHADOW is used to specify whether or not a "shadow line," or notice, is displayed to indicate how many lines have been excluded from the display.

For examples of how these options are used in a macro, you can examine the ALL macro (ALL XEDIT) and the X prefix macro (PREFIX XEDIT).

- **SET SELECT**
designates a "selection level" for the specified lines. A selection level is a value assigned to a line in a file. Various lines in a file may be grouped logically by assigning them the same selection level.

The format of the SET SELECT subcommand is:

```
[SET] SElect [+|-] n [target]
```

- **SET DISPLAY**
specifies which selection levels of lines (as defined by SET SELECT) are displayed.

The format of the SET DISPLAY subcommand is:

```
[SET] DISPlay n1 [n2|*]
```

A range of selection levels is specified by using n1 and n2.

- **SET SCOPE**
specifies the set of lines that the editor operates on. The operand (ALL or DISPLAY) indicates if the collection includes all lines in the file or only the displayed lines. The initial setting is SCOPE DISPLAY.

The format of the SET SCOPE subcommand is:

```
[SET] SCOPE Display|All
```

- **SET SHADOW**
specifies whether or not a shadow line is displayed to indicate how many lines have been excluded from the display. The shadow line appears in the physical position in the file of the excluded line(s). The initial setting is ON.

The format of the SET SHADOW subcommand is:

```
[SET] SHADow ON|OFF
```

Screen Control Enhancements

Several new and enhanced subcommands provide greater flexibility in tailoring the XEDIT screen.

Vertical Screen Split

In addition to dividing the screen horizontally, you can now split the screen vertically or in a combination of horizontal and vertical segments. Each logical screen becomes, in effect, an independent terminal with its own file identification line, command line, and message line.

The format of the extended SET SCREEN subcommand is:

```
[SET] SCREEN n      [Horizontal | Vertical]
                Size  s1[s2[s3...[sn]]]
                Width w1[w2[w3...[wn]]]
                Define sl1 sw1 sh1 sv1 [sl2 sw2 sh2 sv2]...
```

where:

n is the number of screens created as a result of the command, either horizontal (by default) or vertical.

Size indicates that the screens are created horizontally and s1 s2 s3 ...sn are the number of lines of each screen.

Width indicates that the screens are created vertically and w1 w2 w3...wn are the number of columns of each screen.

Define indicates that the screens are created by giving the layout of each screen. Any number of screens can be created, provided "sln" is not less than 5 and "swn" is not less than 20.

The entire screen width must be accounted for when SET SCREEN WIDTH or SET SCREEN DEFINE is used.

Examples:

```
SET SCREEN 2 V
    splits the screen vertically into two logical screens.
```

```
SET SCREEN SIZE 14 10
    splits the screen horizontally into two logical screens, the first of which is 14
    lines long and the second of which is 10 lines long.
```

```
SET SCREEN WIDTH 25 25 30
    creates three logical vertical screens, 25, 25, and 30 columns wide.
```

On a 24 X 80 screen, suppose you want to define three logical screens with the following layout:

```
(1,1)                (1,41)
*****
*                    *                    *
*                    *                    *
*                    *                    *
*          16 X 40    *          16 X 40    *
*                    *                    *
*                    *                    *
*                    *                    *
(17,1)*****
*                    *                    *
*                    *                    *
*                    *                    *
*                    *                    *
*                    *                    *
*****
```


The required subcommand is:

```
SET SCR DEF 16 40 1 1 16 40 1 41 8 80 17 1
```

Message Line

The new **MSGLINE** option on the **SET** subcommand defines the location of the message line on the screen and the maximum number of lines that a message may occupy. It may also be used to determine whether or not a blank line is normally displayed. With this option, you can prevent multi-line messages from being displayed on a cleared screen.

The format of the **SET MSGLINE** subcommand is:

```
[SET] MSGLine ON M[+n|-n] | [+|-]n [p|1] [Overlay]
                OFF
```

“M” stands for the middle of the screen (rounded up for odd-sized screens). “M” can be combined with a constant positive or negative integer to mean “n” lines below (+) or above (-) the middle line.

Middle of Screen

The following subcommands are also enhanced to allow you to specify the location of the line in question relative to the middle of the screen as well as relative to the top or bottom (as shown in the **SET MSGLINE** subcommand format, above).

- **SET CURLINE**
- **SET RESERVED**
- **SET SCALE**
- **SET TABLINE**

Command Line

A new operand **OFF** added to **SET CMDLINE** removes the command line from the screen.

CMDLINE OFF must be issued before a **SET RESERVED** subcommand can overlay the command line. This prevents the user from accidentally overlaying the command line, thereby making it impossible to issue **XEDIT** subcommands.

PF Key Changes

For consistency with other products, the initial settings of various PF keys have been changed for both **XEDIT** and **CMS** commands that use **XEDIT** to interface with the user. New functions, discussed below, are assigned to some of the PF keys. On terminals with 24 PF keys, PF keys 13-24 have the same settings as PF keys 1-12 as discussed here.

XEDIT PF Key Changes

Two new macros and one enhanced subcommand are assigned to PF keys 10, 11, and 12. These new functions are designed to be used from a PF key, but they may also be entered from the command line.

- **RGTLLEFT [n] (PF10)**

The RGTLEFT macro moves the view of the screen to the right or to the left (based on on the current value of the VERSHIFT setting). Pressing the PF key assigned to RGTLEFT moves the view to the right; pressing the PF key again returns the screen to the original view of the file.

- SPLTJOIN (PF11)

The SPLTJOIN macro enables you either to split a line or join two lines, depending on the cursor position. If the cursor is positioned *before* (or at) the last character in a line, the line is *split*. If the cursor is positioned *after* the data, the next line is *joined* to it.

SPLTJOIN incorporates the functions of both SPLIT and JOIN on one key; and it assures that you will not lose data by accidentally joining a line and overlaying data on that line.

- CURSOR HOME subcommand (PF12)

HOME is a new operand on the CURSOR subcommand that moves the cursor from the command line or to its previous position on the screen, or vice versa.

However, if you have tailored a PROFILE XEDIT macro, when you XEDIT a file your PROFILE XEDIT macro will continue to override the XEDIT defaults, including the PF keys.

CMS Commands

The following CMS commands that use XEDIT to interface with the end-user have been updated to use the new functions described above or to use PF key settings in a consistent way. The PF key display on the bottom of the panels created by these commands is updated to reflect the changes. In addition, the PF key display is now strung out sequentially across the screen instead of the “block” format.

The EXECUTE function that was assigned to PF10 for FILELIST, RDRLIST, and SENDFILE is now assigned to the ENTER key.

FILELIST command	PF10 no longer performs the EXECUTE function; this function is assigned to the ENTER key. PF10 is left undefined. PF12 is set to CURSOR HOME.
HELP command	PF4 is set to RETURN (former CANCEL function assigned to PF12). The two functions previously assigned to PF4 (TAB for menus and PRINT for HELP files other than menus) are assigned to the the ---> key and the PA2 key, respectively. PF12 is set to CURSOR HOME. In addition, on a menu, pressing the ENTER key as well as the PF1 key displays the HELP file name under which the cursor is positioned.
NAMES command	PF12 is set to CURSOR HOME.
NOTE command	PF10 is set to RGTLEFT. PF11 is set to SPLTJOIN.
PEEK command	PF10, PF11, and PF12 are set to RGTLEFT, SPLTJOIN, and CURSOR HOME, respectively.

RDRLIST command PF10 no longer performs the EXECUTE function; this function is assigned to the ENTER key. PF10 is left undefined. PF12 is set to CURSOR HOME.

SENDFILE command PF10 no longer performs the EXECUTE function; this function is assigned to the ENTER key. PF10 is left undefined. PF12 is set to CURSOR HOME. The description of the PF5 key has changed from "EXECUTE and QUIT" to "SEND" (although the function performed is unchanged). Pressing the ENTER key sends the file but does not quit. Also, the SENDFILE FILELIST screen has been changed in the same fashion as the FILELIST screen.

Summary of PF Key Changes

This figure shows the PF key changes for Release 3 for XEDIT and for the full screen panels that run in XEDIT environment.

KEYS	PF4	PF10	PF11	PF12
XEDIT		Rgtright	Spltjoin	Cursor
FILELIST		undefined		Cursor
HELP	Return			Cursor
NAMES				Cursor
NOTE		Rgtright	Spltjoin	
PEEK		Rgtright	Spltjoin	Cursor
RDRLIST		undefined		Cursor
SENDFILE		undefined		Cursor

Figure 4. PF Key Changes

Note: On the full screen panels, PF10 no longer performs the EXECUTE function. Press ENTER to EXECUTE.

Additional New Functions

- EXTRACT subcommand**
 Use the EXTRACT subcommand in a macro to get information about internal XEDIT variables or about file data. EXTRACT now includes the function of the TRANSFER subcommand. The information is returned as one or more data variables in the form "name.n," where "name" is the same as the variable requested and "n" is a subscript that distinguishes the different values returned for each option requested.

For example, if a macro wants information about the current line it can issue:

```
EXtract /CURLINE/
```

This returns information about, and the contents of, the current line in the variables (CURLINE.0 - CURLINE.4).

- MERGE**
 The MERGE subcommand combines two sets of lines. It deletes the first set and modifies the second in place.

The format of the MERGE subcommand is:

Merge target1 target2 [col]

- SET options
 - SET ALT
Changes the number of alterations that have been made to the file since the last AUTOSAVE or SAVE. Displayed in the file identification line as ALT=n.
 - SET ENTER
Defines a meaning for the ENTER key.
 - SET FULLREAD
Allows recognition by XEDIT of 3270 display terminal's null characters in the middle of screen lines. FULLREAD ON, with NULLS ON, allows XEDIT to "see" the nulls in the middle of lines, thereby allowing you to write tabular or pictorial data on the screen without worrying about its being "crushed" to the left side of the screen.
 - SET LASTLORC
Specifies the contents of the LASTLORC (last locate or change) buffer. (The buffer is updated automatically with the last user subcommand *as is* when a LOCATE, CLOCATE, CHANGE or any subcommand in the FIND family is executed.)
 - SET PAn
Defines a meaning for a PA key.
 - SET REMOTE
Speeds the transmission of data, for example, when a terminal is remotely attached.
 - SET SIDCODE
Inserts a character string in every line of an update file.
 - SET SPILL
Allows you to specify if data should be truncated or "spilled" onto new lines following various subcommands (for example, CHANGE and GET).
 - SET TRANSLAT
Controls uppercase translation of specified characters. This option is designed to be used on terminals whose keyboards support characters other than English. By default, only the English alphabetic characters are translated to uppercase.

Modifications to Existing Functions

These will improve flexibility and/or consistency among related subcommands.

- Improved Cursor Handling
Two new operands are added to the CURSOR subcommand.

The **PRIORITY** operand allows you to associate a priority with a specified cursor position. If no priority is specified, all priorities are ignored, and the last CURSOR subcommand issued positions the cursor.

The **HOME** operand is discussed under "PF Key Changes".

- **Targets**
 - **Logical AND Operator**

The ampersand symbol “&” is defined to be the logical AND operator on multi-string targets. This is similar to the “|” symbol defined as the logical OR operator. For example:

```
/if/ & /of/
```

will find a line containing both the strings “if” and “of”.
 - **Number of Strings in Target**

Removes the limitation of four strings in targets when logical expressions (AND and/or OR) are specified.
- **SCHANGE n**

The SCHANGE macro is enhanced to do an automatic repeat of the subcommand saved in the LASTLORC buffer (last locate or change) whenever it is called without a CLOCATE or CHANGE subcommand available in the stack.
- **SPLIT and JOIN Aligned**

The ALIGNED operand on the SPLIT and JOIN macros allows proper treatment of indented data. The new SPLTJOIN macro handles indented data in the same way.

SPLIT gives the created line(s) the same number of leading blanks as the original line. Conversely, JOIN removes the same number of leading blanks as there are on the line being joined to.
- **XEDIT Command**

The following options are added to the XEDIT command (and XEDIT subcommand and the LOAD subcommand).

 - **NOMsg** option enters a file with a default MSGMODE OFF.
 - **UNTil** option, used only in conjunction with CTL option, specifies the last update filetype to be applied.
- **Filetype Defaults**

Defaults for various filetypes have been extended.
- **Protected FILE and SAVE**

PFILE and PSAVE are protected equivalents to FILE and SAVE, and are the defaults.
- **Multiple Symbolic Names**

Multiple symbolic names can be assigned for a line (using separate SET POINT subcommands or .xxxx prefix subcommands).
- **Nulls in Prefix Area**

You may set nulls in the prefix area with this format:

```
SET PREFIX Nulls
```

With NULLS ON, the leading zeroes are translated to nulls. This allows the use of insert mode when typing in the prefix area.

- **REFRESH**
Issued from a macro, REFRESH presents the screen at that moment in processing, without waiting for input.
- **LINEND in Synonyms**
This accepts LINEND separation in synonym definitions.
- **Horizontal TOL and EOL**
The column pointer can now be moved to zone-1 (TOL) or to zone+1 (EOL). CLOCATE -* now moves the column pointer to TOL; CLOCATE +* now moves the column pointer to EOL. By first moving the column pointer to TOL or EOL, you can use CLOCATE to find a string that starts in the left or right zone.
- **SET VERIFY**
The H operand can be spelled out as "HEX".
- **QUERY PF [n | *]**
This allows "Query PF *" with the same function as the current "Query PF."
- **Locate and CLocate //**
CLOCATE // advances the column pointer one column. LOCATE // advances the line pointer one line.
- **WRAP Warning**
This provides a warning when WRAP occurs during LOCATE, CLOCATE, or any of the FIND family, whether wrap around EOF or TOF.
- **Screen Changes**
 - An additional equal sign "=" has been added to the arrow of the command line, aligning vertically the first input position of the command line with the first column of the file data area.
 - ALT=nn is displayed in the file identification line.
- **ARBCHAR alone in CHANGE**
CHANGE /\$/xxx/ where \$ is the ARBCHAR character, performs the change up to zone2, not up to the truncation column.
- **STATUS/MODIFY Changes**
STATUS and MODIFY have been updated for all applicable SET subcommands.
- **SET STAY**
The STAY option has been expanded to include additional XEDIT subcommands.
- **QUIT/PURGE from a macro**
If only one file is being edited and "QUIT" is issued from a macro, it gives RC=1 and executes the QUIT when the macro completes execution.

For more information about XEDIT, see the *VM/SP System Product Editor Command and Macro Reference* and *VM/SP System Product Editor User's Guide*.

Enhanced Serviceability

New functions which will increase **reliability, availability and serviceability** include:

- Support of System/370 Program Event Recording (PER)
 - Symptom Record (VM/IPCS/E)
 - DMSDOS split
 - DMSQRY split
-

Support of System/370 Program Event Recording (PER)

This enhancement provides CP command support to facilitate debugging in a virtual machine by the end-user. The new command, CP PER, uses the System/370 hardware program event recording (PER) feature to selectively trace the execution of instructions that cause specific events. The specific events that can be traced are:

- successful branches
- the fetching and execution of instructions
- the execution of instructions that alter storage in the virtual machine
- the execution of instructions that alter general purpose registers.

In addition, the support allows you to trace within multiple ranges, to compare altered storage and registers to specific values, and to execute CP commands on specific events.

The trace output produced by the CP PER command can be recorded on the terminal, the virtual printer, or on both the terminal and the printer.

Using the CP PER Command

The CP PER command has many options that allow you to choose which events are to be monitored. Trace output for the CP PER command is always produced *after* the instruction executes.

The RANGE option allows CP PER to monitor events that occur as a result of the execution of instructions within a specified range or ranges. For example, if you wish to monitor all instructions within your program, (assuming that your program is 500 bytes in length and is loaded at X'20000'), you can issue

```
per instruct range 20000.500
```

There is no need to use the ADSTOP command first, as is the case with CP TRACE. In this example, only instructions in the range 20000-204FF are monitored.

When your assembly program has been loaded and started, you will receive information at your terminal that looks like this:

```
020000 STM 90ECD00C 00DFAC CC=0
```

This line indicates that a STM instruction (located at address 20000) stored the registers 12 through 14 at location 00DFAC and that the condition code is now 0.

The CP QUERY command with the PER option can be used to determine what events are currently being traced. For example:

```
query per
1 INSTRUCT RANGE 020000-0204FF TERMINAL NORUN
```

In addition, if you wish to trace instructions from your program that alter registers, enter:

```
per g range 20000.500
```

The events that you are now monitoring (your current trace set) are:

```
query per
1 INSTRUCT RANGE 020000-0204FF TERMINAL NORUN
2 G RANGE 020000-0204FF TERMINAL NORUN
```

If you continue program execution by entering BEGIN you will receive information at your terminal that might look like this:

```
020004 BALR 05C0 000000 CC=0 G12=40020006
```

This line indicates a BALR instruction at address 020004 changed register 12 to 40020006. As with CP TRACE, it is possible to specify that output is sent to the virtual printer and/or the terminal for any event.

The NORUN option causes the program execution to stop after the trace output has been sent to the terminal. NORUN is the default if either TERMINAL or BOTH is specified.

CP PER has additional options that can be used with all events. For example, the RANGE or FROM option can be used to set up multiple instruction address ranges. This can increase the selectivity with which instruction execution is monitored.

The PASS option allows you to suppress a specific number of events between displays. The CMD option can be used if you want to execute CP command(s) whenever a given event occurs. Also, the STEP option lets a specified number of events be displayed before the CP command environment is entered.

You can find more information on the CP PER command in the *VM/SP CP Command Reference for General Users*.

Symptom Record (VM/IPCS/E)

Symptom record support aids in program problem determination and problem source identification. Symptom record information can be useful in communicating problems to the IBM support center.

The symptom record is a variable length, 2K record added to the front of any dump resulting from a CP abend or use of the VMDUMP command. IPCS/E creates the symptom record and includes it as the first physical record in the IPCS/E dump file. You can view the record on the console screen or have it printed with, or independent of, the dump. The record is made up of three sections:

1. Environmental information - component/release/feature level of the system, processor model and serial number, date and time, and type of dump
2. Offsets to, and length of, third section of the dump
3. Symptom string - component-related symptoms consisting of the error code, component ID, module ID, and register/PSW difference.

The VM Interactive Problem Control Program System Extension Release 2 is required to implement this enhancement.

You can find more information on symptom record in the *IPCS/E User's Guide and Reference*.

DMSDOS Split

The DMSDOS module receives initial control on all VSE SVC invocations. Several SVCs have been split from the DMSDOS module and eleven new modules were created, all residing in the CMSDOS discontinuous shared segment (DCSS). This was done to improve serviceability and maintainability. DMSDOS invokes the appropriate processing routines, and control returns to DMSDOS upon completion.

VM/SP Release 3 supports a new operand for SVC107, PCEXIT, which returns the pcexit routine address and save area in register 0 and register 1, respectively.

Also, CMS/DOS now simulates the VSE IDUMP macro. When a program product issues the VSE IDUMP macro, a dump is generated and sent to a virtual printer.

The new modules are:

DMSLDF	DMSRPG	DMSSUB	DMSVIS	DMSGTM	DMSLIC
MSGVE	DMSMCM	DMSSTX	DMSSVL	DMSGMF	

You can find information about the DMSDOS module, new modules, and SVC Support Routines in the *VM/SP System Logic and Problem Determination Guide Volume 2 (CMS)*.

DMSQRY Split

The DMSQRY module has been split into eight modules. The CMS QUERY command is now a nucleus-resident command, and no longer executes in the transient area.

The eight new modules are:

DMSQRS	DMSQRT	DMSQRU	DMSQRV
DMSQRW	DMSQRX	DMSQRY	DMSQRZ

These eight modules are described in the *VM/SP System Logic and Problem Determination Guide Volume 2 (CMS)*.

For information on the CMS command, QUERY, see the *VM/SP CMS Command and Macro Reference*.

CP Enhancements

CP has several enhancements that include:

- **Enhancements for application programs**
 - Enhancements to the **Inter-User Communication Vehicle (IUCV)**
 - Addition of a **DASD Block I/O System Service**
 - **Enhancements for improved CP Performance**
 - Nucleus reduction
 - Enhancement to **SET QDROP**
 - Support for larger-than-4K buffer terminals
 - **Other CP enhancements**
 - DIAGNOSE code X'00'
 - DIAGNOSE code X'8C'
 - DIAGNOSE code X'14'
 - SET LOGmsg
-

Enhancements for Application Programs

Enhancements to the Inter-User Communication Vehicle (IUCV)

IUCV has several additions and changes that may be used to advantage by application programs. The changes in several functional areas include:

1. **Data specification in the parameter list and related protocol extensions**

An IUCV communication enhancement reduces the path length within IUCV by providing a new option on the **SEND** and **REPLY** functions. This new option, called **DATA=**, allows you to specify two fullwords of data in the parameter list instead of specifying a buffer address and a message length. Since IUCV will reference only the user's parameter list, no other reference to the virtual machine's address space will be necessary for data movement.

DATA=PRMMSG specifies that your message or reply data is contained in the parameter list. **DATA=BUFFER** specifies that your message or reply will be contained in a buffer.

A new option, called **PRMDATA=**, on the **ACCEPT** and **CONNECT** functions is used to specify acceptance of the message data in the parameter list. Specify **PRMDATA=YES**, if you are willing to receive messages via the **DATA=PRMMSG** option in your parameter list. Specify **PRMDATA=NO**, if you are not willing to receive messages sent into your parameter list but will only accept messages sent using a buffer.

2. **Extended mask capability** to allow the masking of individual IUCV control interrupts

To allow a virtual machine to better control the individual IUCV control interrupts currently being handled, a new IUCV **SETCMASK** function is defined. The virtual machine may specify a new mask to separately enable or disable external interrupts for the five types of IUCV control interrupts, these include:

- Connection Pending
- Connection Complete
- Path Severed
- Path Quiesced
- Path Resumed

3. Expanded trace capability to record all IUCV operations in the CP trace table

The IUCV trace table entries for the CP trace table have been expanded to include entries for **QUERY**, **TEST MESSAGE**, **RETRIEVE BUFFER**, and **SETCMASK**. With these functions included, all IUCV functions will be recorded in the CP trace table.

The trace entry processing accumulates trace information and creates a trace table entry only at the completion of the function. Thus, invalid IUCV trace table entries can no longer occur.

4. An IUCV macro option lets you initialize the parameter list while inhibiting the generation of the IUCV instruction

To enhance the usability of the new CMS IUCV macros, **HNDIUCV** and **CMSIUCV**, a new IUCV macro form option is provided. This option, called **MF=**, allows you to initialize an IUCV parameter list for the specified IUCV function without generating the instruction that actually invokes IUCV. The **MF=L** option is allowed as a keyword parameter on any IUCV function which uses a parameter list. Using this option, you can initialize your IUCV parameter list and then invoke the **CMSIUCV** macro with the parameter list as the required input.

5. Support DASD Block I/O System Service

The new System Service for DASD Block I/O utilizes the IUCV system service interface.

The DASD Block I/O System Service can coexist with the existing IUCV system services: Console Communications Services and the Message System Service.

The IUCV macro will do the same extensive checking of the new function as it does for existing functions to verify that the parameters are consistent and meaningful.

All existing use of the IUCV macro and IUCV system services will be unchanged to provide you with a consistent interface from release to release. The operation of all existing IUCV functions remains unchanged with the addition of this support. Programs currently using IUCV will run correctly under the new support without modification.

You can find more information about IUCV in the *VM/SP System Programmer's Guide* and *VM/SP System Logic and Problem Determination Guide Volume 1 (CP)*.

Addition of a DASD Block I/O System Service

The DASD Block I/O System Service is a CP system service. It provides a virtual machine with a device-independent method of reading or writing fixed-size blocks of data on a 512-byte, 1K-, 2K-, or 4K- CMS minidisk.

Device types supported are:

- the Count Key Data (CKD) devices:

2314	2319	3330	3333	3340
3344	3350	3375	3380	

- the Fixed Block Architecture (FBA) devices:

3310 3370

- Device 2319 is formatted as a 2314
- Device 3333 is formatted as 3330
- Device 3344 is formatted as a 3340

This service supports CMS minidisks formatted in block sizes of 512, 1024, 2048, and 4096 bytes.

The CMS RESERVE command and the CMS DISKID function must be issued before using the DASD Block I/O System Service. These two facilities enable you to format a DASD and obtain information to use the DASD Block I/O System Service. For further information, see *VM/SP CMS Command and Macro Reference*.

DASD Block I/O uses IUCV to set up communication between itself and a virtual machine. The IUCV macro checks the validity of all the IUCV parameters. Any IUCV errors are handled according to IUCV specifications. The DASD Block I/O System Service checks the validity of all the parameters required by the DASD Block I/O System Service. No special authorization is required for a virtual machine to use DASD Block I/O.

You can find more information about the DASD Block I/O System Service in the *VM/SP System Programmer's Guide* and *VM/SP System Logic and Problem Determination Guide Volume 1 (CP)*.

Enhancements for Improved CP Performance

Nucleus Reduction

The CP resident nucleus has been modified by:

- making all or part of selected resident modules pageable
- constructing new modules from portions of selected resident modules so that you can delete current modules from the loadlist
- eliminating duplicate code in selected resident modules

- conditionally assembling AP/MP-only code

You can find more information about the nucleus reduction in the *VM/SP System Programmer's Guide* and *VM/SP System Logic and Problem Determination Guide Volume 1 (CP)*.

Enhancement to the SET QDROP Command

The SET QDROP command prevents the specified virtual machine from migrating to Q3.

This new support modifies the handling of the dispatching queues for those virtual machines for which the operator has issued SET QDROP OFF. Those virtual machines will never be considered as residing on Q3. Thus, a noninteractive virtual machine, that was formerly migrated from Q2 to Q3, will remain on Q2 and will continue to be dispatched from Q2. This support does not affect the migration of a virtual machine between Q1 and Q2. If the command is issued with the USERS option, those users are not affected by the new support because the new queue migration checking will not be done for their virtual machines. The dispatching of virtual machines which have not issued SET QDROP OFF is not directly affected by this support.

You can find more information about SET QDROP in the *VM/SP System Programmer's Guide*.

Support for Larger-than-4K Buffer Terminals

This support enables terminals that require a larger-than-4K buffer to do full screen operations in a more efficient manner. This support also makes more efficient use of your storage space.

You can find more information about support for larger-than-4K buffer terminals in the *VM/SP System Logic and Problem Determination Guide Volume 1 (CP)*.

Other CP Enhancements

DIAGNOSE Code X'00'

DIAGNOSE X'00' provides general system information such as system name and program products installed, to the virtual machine. It supports the Time Zone differential value between local time and Greenwich Mean Time. The Time Zone value will be a signed hexadecimal fullword value in seconds.

DIAGNOSE Code X'8C'

DIAGNOSE X'8C' retrieves certain device-dependent information from the RDEVBLK eliminating the need for a virtual machine to issue a WRITE STRUCTURE FIELD QUERY REPLY.

When DIAGNOSE X'8C' retrieves certain device-dependent information from the RDEVBLK, it creates a diagnose interface to enable the virtual machine to access it. Because the information is obtained by the control program only when the terminal is turned on, if the characteristics of the terminal are altered dynamically, those changes will not be reflected in the data returned by DIAGNOSE X'8C'. If a "NETWORK ENABLE" is issued to a device with

advanced features and a "NETWORK ATTACH" issued before turning on the device, the advanced features will not operate. The data returned by DIAGNOSE X'8C' will reflect this status.

DIAGNOSE Code X'14'

DIAGNOSE X'14' can determine whether a spool file is in use by other system functions.

DIAGNOSE X'14' has been modified so that subcodes X'0FFE' and X'0FFF' respect the SFBINUSE flag when the file is not in use by the reader. These subcodes wait for a file being used by a system function. If, however, the file is not available within the 250-millisecond time limit, a condition code of 3 and a return code of 24, is returned. This condition shows system problems because of performance or errors in the spooling area.

You can find more information in the *VM/SP System Programmer's Guide*.

SET LOGMSG Command

The CP SET LOGmsg command has been enhanced so that a HOST operator can SET LOGmsg on a distributed system that is using a programmable operator. A HOST operator can specify a LOGmsg text following the LOGmsg sequence number. This lets the operator create or change a LOGmsg without causing a CP READ that would stop the operation of a programmable operator and then have it logged off after 15 minutes.

You can find more information in the *VM/SP Operator's Guide*.

CMS Enhancements

CMS now has several new enhancements that include:

- **Enhancements for Application Programs**
 - CMS support of IUCV
 - Wait on ECB
 - End-of-command nucleus extensions
 - ABEND exit
 - Reading directly from the virtual machine console
 - Creating your own immediate commands
 - QSAM tape end-of-volume exit
 - Support for CP DASD Block I/O System Service
 - **Enhancements to the CMS File System**
 - 512-byte blocksize minidisks
 - RESERVE command
 - Update-in-place
 - **Enhancements for Improved CMS Performance**
 - LISTFILE and RENAME migration
 - Reduction of the SCBLOCK look-aside buffer
 - Removal of the CMSSEG segment
 - Enhancement to GLOBALV
 - **Other CMS Enhancements**
 - Enhancements to TELL and DEFAULTS commands
 - EXECOS command
 - SET EXECTRAC command
-

Enhancements for Application Programs

There are several CMS enhancements included to support CMS application programs. All new Assembler macros have standard, list, and execute formats to let you code reentrant programs.

CMS Support of IUCV

CMS now supports Inter-User Communication Vehicle (IUCV) communications. IUCV allows a program running in a virtual machine to communicate with another virtual machine or with a CP system service. In addition, multiple programs running in one virtual machine can also use IUCV. CMS makes it easier for a program to use IUCV functions.

You can invoke IUCV functions via the new CMS macros, HNDIUCV and CMSIUCV. These macros enable you to:

- initialize and end a program's IUCV environment
- begin or end communications with another virtual machine or with CP

- specify particular exits for IUCV external interrupts.

HNDIUCV Macro

Use the HNDIUCV macro to tell CMS that your program wishes to use IUCV. A general exit is set up, and it will be given control whenever an IUCV CONNECT PENDING external interrupt is reflected to the virtual machine for your program. No CMS IUCV function is permitted by a particular program unless the program has first issued the HNDIUCV macro and identified itself to CMS.

CMSIUCV Macro

Use the CMSIUCV macro to begin or end IUCV communications with another IUCV program or with CP. By issuing CMSIUCV CONNECT or CMSIUCV ACCEPT, CMS invokes the IUCV CONNECT function or IUCV ACCEPT function, respectively. This establishes a path between the program and another program or CP, and establishes an exit to be given control whenever an external interrupt occurs on this path. By issuing CMSIUCV SEVER, CMS will invoke the IUCV SEVER function and the exit associated with the path will be cleared. All other IUCV functions, such as SEND and RECEIVE, are issued directly by the program.

Some IUCV functions should not be used with CMS IUCV support. These functions are listed in the *VM/SP System Programmer's Guide*, along with more information on CMS IUCV support.

Wait on ECB

An Event Control Block (ECB) is a standard mechanism used to synchronize multiple events in CMS. ECBs are fullwords that have OS or VSE formats. The new CMS macro, WAITECB, waits on an Event Control Block (ECB) or a list of ECBs.

The CMS nucleus, NUCON, contains an ECB that makes it easier to wait for a console I/O in a series of multiple events. This ECB is posted whenever there is a line (or lines) in the terminal input buffer. This ECB applies only to the terminal input buffer. No ECB is associated with the program stack.

You can find more information about the wait support of ECB in the *VM/SP CMS Command and Macro Reference*.

End-of-Command Nucleus Extensions

The NUCEXT function has been extended to allow NUCLEUS extensions to receive control at normal end-of-command processing. After a command is processed, CMS invokes all nucleus extensions with the ENDCMD attribute. ENDCMD nucleus extensions only receive control after a command is entered from the virtual console. They do not receive control if the command was issued from an EXEC, a user program, or CMS subset mode.

ENDCMD is a new option to the NUCXLOAD command. If you issue the NUCXLOAD command with the ENDCMD option, an ENDCMD nucleus extension is established. The NUCXMAP command may be issued to determine whether or not an ENDCMD nucleus extension exists.

For more information, see *VM/SP CMS Command and Macro Reference*.

ABEND Exit

The ABNEXIT macro instruction allows a program to specify the address of a routine which will receive control before CMS ABEND recovery begins. Once driven, the exit routine may choose to avoid ABEND or return to CMS for ABEND recovery. The ABNEXIT RESET function should be issued if a program wishes to avoid ABEND recovery and continue. The RESET function clears the condition that indicates control was given to an exit routine. More than one exit routine can be set, and CMS maintains a list of these exits. The last exit set is considered to be the current exit. When an exit is cleared, the previous exit on the list, if one exists, becomes the current exit. An exit can be cleared regardless of its position in the list.

For more information, see *VM/SP CMS Command and Macro Reference* and *VM/SP System Logic and Problem Determination Guide Volume 2 (CMS)*.

Reading Directly from the Virtual Machine Console

An option, TYPE=DIRECT, has been added to the RDTERM macro. This option allows a program to read an input line directly from the virtual machine console.

When a program issues RDTERM TYPE=DIRECT, a VM READ is presented at the terminal. The program stack and terminal input buffer are bypassed and not changed. When the response is entered, the first "logical line" is read and transferred to the user's buffer. If multiple "logical lines" are entered, the remaining lines are added to the terminal input buffer in a FIFO manner.

For more information see *VM/SP CMS User's Guide* and *VM/SP CMS Command and Macro Reference*.

Creating Your Own Immediate Commands

CMS now allows you to define your own immediate commands. Immediate commands can be established from Assembler language programs via the IMMCMD macro, from within EXECs via the IMMCMD command, and as a NUCLEUS extension via the NUCXLOAD command and the NUCEXT function.

The IMMCMD macro is issued from an Assembler language program. It declares, clears, and queries immediate commands. The program supplies an exit address that is given control whenever a command is entered from the terminal.

The IMMCMD command sets and clears immediate commands. It also determines the status of immediate commands. The IMMCMD command should be issued only from EXECs (System Product Interpreter EXECs, EXEC 2 EXECs, and CMS EXECs.)

The NUCEXT facility now allows a nucleus extension to be invoked as an immediate command. The IMMCMD attribute signals this new type of nucleus extension. IMMCMD is a new option to the NUCXLOAD command. If you issue the NUCXLOAD command with the IMMCMD option specified, an IMMCMD nucleus extension is established. You can issue the NUCXMAP command to verify whether or not an IMMCMD nucleus extension exists.

User-defined immediate commands can override built-in CMS immediate commands. However, built-in CMS immediate commands cannot be cleared.

You may have a regular command and an immediate command with the same name. For CMS to distinguish between the two, you can set an escape character using the SET IMESCAPE command.

The format of the SET IMESCAPE command is:

```
SET IMESCAPE ON|OFF|char
```

If you issue SET IMESCAPE ON, the escape character is a semi-colon (the default value), or you can specify your own escape character. If you wish CMS to handle your command as an immediate command, the immediate command name must be prefixed with either a semi-colon or the character you set.

For more information on defining your own immediate commands, see *VM/SP CMS Command and Macro Reference*.

QSAM Tape End-of-Volume Exit

A program working with CMS simulation of OS QSAM can set up an exit to be entered on the end-of-volume condition on IBM standard-label tapes. This exit receives control after the trailer labels have been processed and the tape has been rewound and unloaded. Without this exit, CMS stops the application program after the end-of-volume condition is reached.

You can find more information about the QSAM tape end-of-volume exit in the *VM/SP System Logic and Problem Determination Guide Volume 2 (CMS)*.

Support for CP DASD Block I/O System Service

CP DASD Block I/O System Service provides a fast, device-independent method of reading or writing fixed size blocks of data on a 512-byte, 1K-, 2K-, 4K- CMS minidisk. In order to use this support with CMS formatted minidisks, two new functions and utilities are provided:

- an extension to the FILEDEF command
- the DISKID function

FILEDEF Command Extension

The FILEDEF command can now associate a “ddname” to a virtual minidisk address. This information is needed whenever you use the DASD Block I/O System Service.

The format of the new attribute of the FILEDEF command is:

```
FILEDEF ddname DISK vaddr
```

The “ddname” is the name of the virtual minidisk referred to in your program. The “vaddr” is the virtual address of the device referred to in your program by “ddname.”

This form cannot be used as a regular FILEDEF for OS simulation. If an OPEN is issued for such a “ddname,” it will fail.

You can find more information about the FILEDEF command in the *VM/SP CMS Command and Macro Reference*.

DISKID Function

The DISKID function returns information on the physical organization of a minidisk that was RESERVED. DISKID obtains the virtual address, the blocksize, and the offset of the minidisk. The offset is the number of sequential blocks used on the minidisk by the CMS file system to implement its structure.

You can find more information about support for CP DASD Block I/O System Service in the *VM/SP CMS Command and Macro Reference*. Information about how to use DASD Block I/O System Service is found in the *VM/SP System Programmer's Guide*.

Enhancements to the CMS File System

512-byte Blocksize Minidisk

The physical DASD blocksize of a CMS minidisk can now be 512 bytes. Use the FORMAT command to format a 512-byte block minidisk.

The new option on the FORMAT command is BLKSIZE 512.

RESERVE Command

Use the new RESERVE command to allocate all available blocks of a 512-byte, 1K-, 2K-, or 4K- block formatted minidisk to a unique CMS file.

The format of the RESERVE command is:

```
RESERVE filename filetype filemode
```

After the completion of the command, the file defined has the following characteristics:

- filename, filetype, and filemode as defined by you
- filemode number 6 (indicating "update-in-place") if not specified in the command
- logical record length equal to the CMS disk block size
- fixed (F) record length
- the number of records is the total number of blocks available on the disk minus the number of blocks used by CMS. This CMS overhead varies with the size of the minidisk.

Issue the RESERVE command only if a minidisk is already accessed. The RESERVE command does not rewrite the data blocks of the file being created. The data blocks contain whatever was left in the location they occupied on disk. To clear these blocks with binary zeros, use the FORMAT command before you issue the RESERVE command.

For more information see *VM/SP CMS Command and Macro Reference* and *VM/SP CMS User's Guide*.

Update-in-Place

Update-in-place allows you to write blocks back to their previous location on disk rather than in a new location. The filemode number 6 differentiates CMS files with the update-in-place attribute from regular CMS files. This applies only to 512-byte, 1K-, 2K-, or 4K- block formatted minidisks.

To take advantage of update-in-place, the FSWRITE macro must be used, whether explicitly by you or implicitly by the system.

For more information see *VM/SP CMS Command and Macro Reference* and *VM/SP CMS User's Guide*.

Enhancements for Improved CMS Performance

LISTFILE and RENAME Migration

The LISTFILE and RENAME functions, formerly executed in the transient area, are now reentrant and have been moved to the CMS shared nucleus.

The migration of these two functions, LISTFILE and RENAME, affects CMS in two major ways:

- **Installation**
Simplifying installation and system maintenance
- **Performance**
Improving performance in these commands and in programs making heavy use of these functions, such as XEDIT and FILELIST.

There is no difference in the invocation of LISTFILE and RENAME.

Reduction of the SCBLOCK Look-Aside Buffer

The number of SCBLOCKs that are kept in look-aside buffers is reduced from 128 to 64. Sixty-four SCBLOCKs was chosen because the amount of storage taken up by them is an even integral of half a page.

Removal of the CMSSEG Segment

The discontinuous saved segment, CMSSEG, has been removed by merging modules from the CMSSEG segment into the CMS shared nucleus.

The removal of the CMSSEG segment should benefit you in two major ways:

- **Installation**
Installing and maintaining a CMS system are easier because the maintenance personnel do not have to generate the CMSSEG.
- **Performance**
Merging the CMSSEG and the shared nucleus eliminates duplicate modules.

For more information see *VM/SP System Logic and Problem Determination Guide Volume 2 (CMS)* and *VM/SP Installation Guide*.

Enhancement to GLOBALV

The GLOBALV command has been enhanced to use the EXECCOMM interface available in EXEC 2 and System Product Interpreter. This makes it possible to set and retrieve global variables by just specifying the name of the variable on the GLOBALV command call.

This is done with the PUT and GET subfunctions that have been added to the SELECT phrase. The PUT and GET subfunctions can be used only from an EXEC 2 or System Product Interpreter EXEC.

For details of the GLOBALV command see *VM/SP CMS Command and Macro Reference*.

Other CMS Enhancements

Enhancements to TELL and DEFAULTS Commands

The TELL command has been enhanced to allow you to select the CP command of your choice, MSG, MSGNOH, WNG, or SMSG, to send local messages. The TELL command now reads the GLOBALV file set up for each user to find any TELL defaults which might have been set by the user.

The DEFAULTS command has been changed to support the enhancement made to the TELL command. When you set your default, it remains that way until you reset it.

Note: To use the MSGNOH command, you must be a Class B user. To use the WNG command, you must be either a Class A or a Class B user.

For more information see *VM/SP CMS Command and Macro Reference*.

EXECOS Command

EXECOS is a new CMS command that resets the OS environment under CMS without returning to the interactive environment. Specify EXECOS with no parameters, or *precede* any CMS command with EXECOS.

If EXECOS precedes a CMS command, first the CMS command is processed, and then the EXECOS command performs the OS reset function. The return code is that of the CMS command that was processed. If EXECOS is specified with no parameters, the OS environment is reset.

The EXECOS command is primarily intended for use in an EXEC 2 or System Product Interpreter EXEC that either invokes several OS programs sequentially or invokes the same OS program repetitively. The EXECOS command clears the following:

- STAE exits
- STIMER exits
- SPIE exits
- STAX exits
- TXTLIBs
- MACLIBs
- SSTAT extension
- LINKLIST (LINKSTRT and LINKLAST)

- OS environment flags (OSSFLAGS)

If VSAM is running, VSAM cleanup is also done.

For more information on EXECOS see *VM/SP CMS Command and Macro Reference*.

SET EXECTRAC Command

You can set a new function with the CMS SET command. The setting of EXECTRAC determines whether you have tracing turned on or off for your System Product Interpreter EXEC or your EXEC 2 EXEC. The format of the SET EXECTRAC command is:

```
SET EXECTRAC ON|OFF
```

For more information on CMS SET see *VM/SP CMS Command and Macro Reference*.

Enhanced Support for VSE/VSAM

CMS support for VSAM includes the following additional functions:

- Support for **VSE/VSAM Release 3**
 - **CATCHECK** command
 - Support for **VSAM Assembler language macros**
 - **VSE/VSAM macros**
 - **OS/VSAM macros**
 - **VSEVSAM command**
 - **CMS/VSAM support of the 3380 Direct Access Storage**
-

Support for VSE/VSAM Release 3

VM/SP supports VSE/VSAM Release 3 for use with CMS. VSE/VSAM Release 2 is not supported.

The CMS support is based on VSE/VSAM Release 3. If you are an OS VSAM user and plan to use CMS to manipulate VSAM files, you are allowed to use those functions of Access Method Services that are available under VSE/VSAM.

VSE/VSAM Release 3 supports the functions that were previously supported as well as the following functions:

- Multiple catalogs may own space in the same DASD volume
- Checking and updating of catalog records where necessary
- Resetting and processing of a list of available catalogs by VSE/VSAM Release 3
- Verifying the syntax of the Access Method Services commands without actually executing them with the SYNCHK parameter of the Access Method Services PARM command
- Deleting incomplete catalog information that might have resulted from a system failure during DEFINE or DELETE processing by using the IGNOREERROR parameter of the Access Method Services DELETE command
- Verifying a complete catalog structure via the CMS CATCHECK command that invokes the VSE/VSAM Catalog Check Service Aid
- Resetting broken free chains (list of available catalog records) and then processing them by VSE/VSAM.

CATCHECK Command

CATCHECK is a new CMS command. The CATCHECK command allows the CMS VSAM user (with or without DOS set ON) to invoke the VSE/VSAM Catalog Check Service Aid to verify a complete catalog structure.

The format of the CATCHECK command is:

```
CATCHECK      [catname  
               catname/password ]
```

CATCHECK produces a print file containing the catalog analysis. For example, issuing:

```
dlbl ijsysuc f dsn private.catl (vsam
```

and

```
catcheck
```

results in a print file containing the VSE/VSAM Catalog Check output for private.catl.

Additional information about the CMS CATCHECK command is in the *VM/SP CMS Command and Macro Reference* and the *VM/SP CMS User's Guide*.

Support for VSAM Assembler Language Macros

CMS supports VSAM macros for use in CMS programs. All of the VSE/VSAM macros and their options and a subset of the OS/VSAM macros are supported by CMS.

Support for VSE/VSAM Macros

The VSE/VSAM macros and their options are supported for use in Assembler language programs under CMS/DOS. The VSE/VSAM macros are:

ACB	EXLST	SHOWCAT
BLDVRP	GENCB	SHOWCB
DLVRP	MODCB	TCLOSE
ENDREQ	POINT	TESTCB
ERASE	RPL	WRTBFR

All options are supported with the exception of "AM=VTAM," which is not supported on any of the macros.

The EXLST EXCPAD exit may be specified but will never be taken in the CMS environment. The reason is that VSE/VSAM takes this exit when it is waiting for I/O to complete, but in the CMS environment I/O is always complete when control is returned to VSE/VSAM.

Support for OS/VSAM Macros

OS users can use a subset of OS/VSAM Assembler language macros in their Assembler language programs. The macros supported are contained in the OSVSAM MACLIB distributed with VM/SP.

You can find more information about the OS/VSAM macros in the *VM/SP CMS Command and Macro Reference* and in the *VM/SP CMS User's Guide*.

VSEVSAM Command

Using the new CMS VSEVSAM command, VSAM users can obtain the VSE/VSAM Assembler language macros from the Licensed Optional Machine Readable Materials tape. The VSEVSAM command creates the VSEVSAM MACLIB for you, containing all of the VSE/VSAM assembler language macros and the following VSE macros:

CDLOAD
CLOSE
CLOSER
GET
OPEN
OPENR
PUT

You can find more information about the CMS VSEVSAM command in the *VM/SP Installation Guide*.

CMS/VSAM Support of the 3380 Direct Access Storage

CMS/VSAM supports the use of VSAM formatted 3380 Direct Access Storage for the OS environment of CMS only. CMS/DOS does not support VSE files or libraries on the 3380, or the use of VSE macros to access data on the 3380.

Previously Announced Functions

VM/SP Release 3 includes these functions, previously announced as late announcements for VM/SP Release 2:

- Support of the **missing interrupt handler**
 - Support of the **speed matching buffer** for the IBM 3375
-

Support of the Missing Interrupt Handler

When an I/O device fails to return an expected interrupt to the control program, a missing interrupt condition exists. An incomplete I/O operation of this type can degrade system performance for virtual machine users.

This support monitors system I/O activity for interrupts not completing within a certain time period and attempts to correct the condition.

When a missing interrupt is detected, the control program will try to correct the condition. When the try at corrective action is complete, a record will be written to LOGREC and a message will be sent to the system operator stating that a missing interrupt condition was detected and either cleared or not cleared. Corrective action is in the form of simulating an error condition to either CP's I/O supervisor or to the virtual machine, depending on the originator of the I/O operation.

Missing interrupt handler support in VM/SP monitors system I/O activity. If an I/O interrupt does not complete within a previously defined interval of time, the control program attempts to clear the condition by simulating an operation that restarts the I/O. If the restart operation is successful and the missing interrupt is cleared, the device remains operational. If the attempt at clearing the missing interrupt is unsuccessful, the operator may be required to perform manual action to free the device. In any event, the control program sends an informational message to the system operator and records the event in the system error recording area.

The missing interrupt handler monitors I/O operations for several classes of devices. Your installation can either specify the time interval it desires or it can use the IBM-supplied default intervals to activate missing interrupt monitoring for the following device types:

- DASD (count-key-data and fixed block architecture)
- Graphic (local)
- Tape
- Unit record (card punch, card reader, and printer)
- Miscellaneous Devices (Mass Storage System, 3287 printer devices, 3800 printer)

The missing interrupt handler does not support System Network Architecture (SNA), pass-through virtual machine, terminal, or special devices.

The MITIME operand of the SET command is used to change the time intervals specified in the system control file. You can also stop missing interrupt monitoring with the SET MITIME OFF command.

The MITIME operand of the QUERY command displays the current time interval settings for all device classes.

The SYSMIT macro instruction is optional. You can use it to define the time intervals desired for monitoring missing interrupts.

Support of the Speed Matching Buffer for the IBM 3375

This supports the 3375 Direct Access Storage when connected to a 1.5 megabyte channel. This includes paging, spooling, and/or minidisks.

The 3880 speed matching buffer feature for the IBM 3375 uses a 16K-byte storage buffer to change the direct access data transfer path between the 3375 and the multiplexer channel. With this feature, the 3375 Direct Access Storage, with its 1.859 megabytes per second data rate, can be attached to block multiplexer channels with data rates as low as 1.5 megabytes per second, and to high speed multiplexer channels. Therefore, the 3375 can be shared by processors with different block multiplexer channel speeds.

Data is buffered in the storage director of the 3880 to accomplish the speed matching function. If channels with speeds different than the 3375's 1.859 megabytes per second data rate are switched to a storage director with the buffer feature, the 1.5 megabyte block multiplexer channels operate at the channel's data rate of 1.5 megabytes per second; the 3.0 and 2.0 megabyte block multiplexer channels operate at the 3375's data rate.

Hardware Support

VM/SP Release 3 includes support for new devices, such as:

- Printers
 - Tapes
 - Communication units
-

Printer Support

3262 Printer, Model 5

The 3262 printer is a channel-attached, medium-speed-impact line printer in the 3262 family. It will appear to an operating system to be identical to the 3262 Model 1 in areas of commands supported, sense data, and recovery actions required of the operating system. It is supported as a virtual, system, or dedicated device. The support is identical to the 3262 Model 1 support.

3800 Printing Subsystem Models 3 and 8

The 3800 Printing Subsystem Models 3 and 8 are supported only as local dedicated printers. Changes have been made to CP to provide dedicated support of these printers.

4245 Line Printer

The 4245 Line Printer is a channel-attached, high-speed line printer (2000 LPM) using print band technology. It combines 3203 and 3262 technologies with newly developed hardware. The programming support for this printer is similar to the 3262 programming support. The 4245 printer is supported as a virtual, dedicated, and system printer. In addition, FCB lengths for 4245 will be allowed the maximum length that hardware supports (256 byte length). It is not limited to a 180-byte maximum length as are previous printers.

4250 Printer

The 4250 printer is a graphics printer with the ability to produce camera-ready, high-quality output. It uses electromatrix technology and aluminum coated paper for printing. The 4250 printer is supported as a local dedicated printer only. The driving of this printer, Start Input/Output (SIO), and error handling are handled by the Compose Fonts Program Product.

The 4250 printer does not support the full range of 3270 commands and does not support the copy function. The VM/SP support of this printer allows you to take advantage of all the functions of the 4250 printer when dedicated to a virtual machine using the Compose Fonts Program Product.

Tape Support

3430 Magnetic Tape Subsystem

The 3430 Magnetic Tape Subsystem is a manual loading reel-to-reel device used to read and record information on standard IBM one-half inch magnetic tape. System attachment is through the S/370 block multiplexer channel using a standard S/370 I/O interface. The 3430 tape drive operates in two different modes, 1600 BPI and 6250 BPI, with respective data rates of 80 Kb/sec and 312 Kb/sec. The default mode is 6250 BPI.

VM/SP CP supports the 3430 tape drive in the following areas:

- System generation
- System macro definition
- I/O management
The 3430 can be dedicated to a virtual machine for guest use (CMS or non-CMS).
- System functions
Include support for MONITOR, SPTAPE and system dump.
- Utilities
DASD-dump-restore (DDR) support 3430.
- Error recovery
Support for 3430 error recovery and recording.

Channel Communication Unit

3088 Multisystem Channel Communication Unit

The 3088 Multisystem Channel Communication Unit is a new I/O device used to interconnect multiple systems via a Multisystem Communications Unit attached to a block multiplexor channel. The 3088 operates in high speed transfer mode, provides an unshared subchannel for each unique unit address, and is fully compatible with existing channel-to-channel adapter (CTCA) protocols.

VM/SP supports all models of the 3088:

- Model 1 which attaches four systems
- Model 2 which attaches eight systems

Both models can support either 32 or 64 unit addresses per I/O interface.

This support is consistent with existing CTCA support in the following areas:

- The existing SYSGEN macros (REDEVICE and RCTLUNIT) have been extended to support the 3088 device type and addressing requirements.

- All functions of the real 3088 are available to a virtual machine as a dedicated channel-to-channel (CTC) device.
- The unit address reserved for Online-Testing is available to a virtual machine as a dedicated 3088 unit address for diagnostic purposes.
- All functions of the real 3088 (except for the OLT functions) are simulated to provide virtual 3088 support.
- The DEFINE command has been extended to support the 3088 as a valid virtual device.
- The existing CTCA addressing and scheduling limitations have been extended to allow scheduling on the device block instead of the control unit block for both the real and virtual CTC devices.

The 3088 Multisystem Communications Unit extends the existing CTCA addressing and scheduling limitations by assigning more than one unit address per channel and by allowing non-zero control unit addresses. With these restrictions removed, you can specify multiple unit addresses per control unit. Additional CP improvements implement block multiplexor channel scheduling for real and virtual CTCAs and 3088s.

This part of the book describes general requirements for migration considerations from VM/SP Release 2 to VM/SP Release 3. It describes:

- **Migration Aids**
- **Migration Compatibilities/Incompatibilities**
- **Performance**

Migration Aids

IBM provides **migration aids** to migrate from VM/SP Release 2 to VM/SP Release 3. These aids include:

- **SPOOL file migration aid**
 - Utility provided to convert **Programmable Operator routing tables**
 - **AP/MP Processor Unit**
-

SPOOL File

SPOOL file information has been changed in Release 3. A migration aid, available via the SPTAPE function, lets current VM/SP Release 1 or Release 2 users migrate their spool files to VM/SP Release 3 without performing a cold start.

Programmable Operator Routing Tables

VM/SP Release 2 Programmable Operator Routing Tables must be converted to the VM/SP Release 3 routing table format. A utility, **PROPRTCV EXEC**, is provided to aid in this conversion.

See "Enhancements to the Programmable Operator Facility" on page 27 for additional information on routing tables.

AP/MP Processor Unit

VM/SP Release 3 support of AP/MP and 3081 Processor Unit Model D16 will be available in the first quarter of 1984.

Migration Compatibilities/Incompatibilities

Compatibilities and incompatibilities in migrating from VM/SP Release 2 to VM/SP Release 3 are described here.

VM/SP Release 3 is compatible with VM/SP Release 2, VM/SP Release 1, VM/370 Release 6 and its extensions, VM/370 Basic System Extensions program product (Program Number 5748-XX8), and VM/370 System Extensions program product (Program Number 5748-XE1).

Migration Compatibilities

Other than the optional compatibility macros announced in VM/SP Release 1, and the previously described conversion aids, no other conversion aids are required to move from VM/Basic System Extensions Release 2, VM/System Extensions Release 2, or VM/SP Release 1 or 2, to Release 3 of VM/SP.

Migration Incompatibilities

Removal of the EDGAR Support

The EDGAR compatibility support is removed from VM/SP Release 3. All those using the migration support *must* complete the migration to the System Product Editor when VM/SP Release 3 is installed.

See "System Product Editor" on page 9 for additional information.

IPCS Incompatibilities with VM/SP

Do not use the IPCS component of VM/370 in a VM/SP environment for:

- Analyzing, formatting, and printing CP dumps taken when the system runs in MP mode
- Analyzing CP dumps with an abend code added since VM/370 Release 6

In addition, be aware that control blocks are formatted by the IPCS component in the Release 6 format. If a block is extended for VM/SP, IPCS does not format the control block extensions. Since the pointer path to RECBLOK has changed for VM/SP, the information that the IPCS component formats to represent the RECBLOK is unpredictable.

The following control blocks are formatted by the IPCS component of VM/370. Those marked with an asterisk show that they are extended to accommodate VM/SP code. The extensions to these control blocks are not formatted by the IPCS component:

CONTASK	*RSPLCTL
*ECBLOK	*SFBLOK
*IOBLOK	*VCHBLOK
IOERBLOK	*VCONCTL
RCHBLOK	VCUBLOK
*RCUBLOK	*VDEVBLOK
*RDEVBLOK	*VMBLOK
*RECBLOK	VSPLCTL

To implement the full capabilities of the VM/SP product, installations should consider a migration to the Release 2 level of the VM/Interactive Problem Control System (VM/IPCS) Extension program product (Program Number 5748-SA1).

RSCS Incompatibilities with VM/SP

The RSCS component of VM/370 has not been modified to support the new function contained in VM/SP that has been added since VM/370 Release 6. The unmodified RSCS component of VM/370 does not:

- Handle virtual 3800 spool files.
- Use a system disk formatted in 512-byte, 1K-, 2K-, or 4K- block sizes.
- Use a system disk residing on a 3380 or FBA (3310/3370) device.
- Support remote 3270 Information Display printers, dedicated to virtual machines.

Installations wishing to implement the full capabilities of VM/SP should consider a migration to the Release 3 level of the RSCS Networking program product (5748-XP1) that supports new VM/SP function.

System Product Editor Migration Considerations

- **PF key changes**
For consistency among CMS and other products in the use of PF keys, certain new or enhanced commands have been implemented in XEDIT. These new functions are assigned to PF keys 10, 11, and 12 by default.

The initial settings for the PF keys 10 through 12 are:

PF10	Rgtright	View data to the right of the screen; press again to return to the original display.
PF11	Spltjoin	Split a line or join two lines at the cursor position.
PF12	Cursor Home	Moves the cursor to the command line or back to its previous location on the screen.

However, if you have tailored a PROFILE XEDIT macro, when you XEDIT a file your PROFILE XEDIT macro will continue to override the XEDIT defaults, including the PF keys.

The following figure shows the PF key changes for Release 3 for XEDIT and for the full screen panels that run in XEDIT environment.

KEYS	PF4	PF10	PF11	PF12
XEDIT		Rgtright	Spltjoin	Cursor
FILELIST		undefined		Cursor
HELP	Return			Cursor
NAMES				Cursor
NOTE		Rgtright	Spltjoin	
PEEK		Rgtright	Spltjoin	Cursor
RDRLIST		undefined		Cursor
SENDFILE		undefined		Cursor

Figure 5. PF Key Changes

Note: On the full screen panels, PF10 no longer performs the EXECUTE function. Press ENTER to EXECUTE.

Also, on the full screen panels the PF keys are strung out across the bottom of the display, instead of the earlier "block" format.

- **SET CMDLINE**
SET CMDLINE OFF must be issued before a SET RESERVED subcommand can overlay the command line. This prevents you from accidentally overlaying the command line, thereby making it impossible to issue XEDIT subcommands.

With SET CMDLINE TOP (command line on line 2) and the default SET MSGLINE setting (line 2), a message overlays the command line, including the arrow. You must press the ENTER or CLEAR key to recover the command line. To avoid this situation, assign the message line to line 1 or line 3 when using CMDLINE TOP.

- **"Protected" FILE and SAVE**
If you change the file identifier while editing a file so that it is the same as that of an existing file, a FILE or SAVE is suspended and the following message is displayed:

```
FILE 'fn ft fm' ALREADY EXISTS. USE FFILE/SSAVE.
```

You then have a choice of either writing over the existing file (by issuing FFILE or SSAVE), or changing the file identifier so that it is unique.

- **Column pointer movement**
The column pointer can now go one position beyond the zones (in either direction). Top of Line (TOL) and End of Line (EOL) are the horizontal equivalents of Top of File and End of File. CLOCATE -* or CLOCATE +* now moves the column pointer one column to the left of the left zone (TOL) or one column to the right of the right zone (EOL). By moving the column pointer to TOL or EOL, you can use CLOCATE to find a string that starts in the left or right zone.
- **Pending notice**
When a prefix subcommand or macro is executed and additional input is required (for example, a "C" has been entered and an "F" or "P" has not yet been entered), the status area displays the following pending notice:

```
'value' pending...
```

where "value" is the prefix subcommand or macro name.

- Null string target in CLOCATE
CLOCATE // is now similar to LOCATE //. CLOCATE // advances the column pointer one column; LOCATE // advances the line pointer one line. Neither returns the message "NO TARGET FOUND" when a null string target is specified.
- New tags on READ TAG
READ TAG now stacks tags for the ENTER key (ETK) and PA keys (PAK).
- Filetype defaults
To be compatible with System Product Interpreter EXECs, the initial CASE setting for filetypes XEDIT and EXEC has been changed from UPPER to MIXED. This will cause problems in EXEC 2 macros if the initial setting is not changed.

The initial NONDISP setting for all file types is " instead of blank. The initial IMAGE setting for file types TEXT, MODULE, and MACLIB is OFF instead of ON. The initial SERIAL setting for file type MACLIB is OFF instead of ON.

- Screen changes
 - The command line arrow has an additional equal sign (====>) so that it aligns with the first column that the user can enter data on.
 - The alteration count is displayed in the file identification line as ALT=.
 - On the panels that operate under the control of XEDIT (FILELIST, NAMES, NOTE, PEEK, RDRLIST, SENDFILE) the message line is moved from the second line of the screen to the line preceding the command line. These panels also have PF key changes, discussed above.
- FILE and QUIT can be used in a macro to exit from the last file in the XEDIT ring. Doing a trial QUIT and stacking a QUIT command on return code 1 causes an extra QUIT to be issued.
- The translate tables for displaying characters have been updated.
- The prefix area is decoded the same way regardless of whether SET NUMBER ON or OFF is in effect. The location of the cursor in the prefix area has no effect on the decoding.
- Cursor handling has changed when there are multiple screens. The cursor remains in the screen it was in, even if a CURSOR subcommand is issued in another screen.
- MODIFY POINT is no longer supported.
- Cursor placement, after the execution of subcommands, is different because of the priorities assigned to the different subcommands.
- The following subcommands are no longer diagnosed and are ignored on typewriter terminals:

SET APL	SET NULLS
SET CTLCHAR	SET NUMBER
SET MASK	SET PREFIX
SET NONDISP	SET TEXT

- SOS options UP, DOWN, LEFT, RIGHT, NFIELD, BFIELD, and NLINE, previously part of the EDGAR compatibility support, are no longer supported.
- To display a reserved line, using SET RESERVED, over the position occupied by the message line, you must use the “Overlay option” when setting the message line (using SET MSGLINE).
- The filename for the AUTOSAVE file has the format of “rrrnnnnn”

where

rrr is the number of times XEDIT has been called recursively

nnnnn is the current autosave number

- When defining a prefix synonym, up to eight characters may be specified for the OLDNAME and up to five characters may be specified for the NEWNAME.

Also, to set a synonym for a built-in block prefix subcommand, you must issue an explicit SET SYNONYM subcommand. For example, SET PREFIX SYNONYM Q " does not imply that QQ is a synonym for "" ; this must be explicitly stated i.e. SET PREFIX SYNONYM QQ "".

Performance

This section describes the Performance of VM/SP Release 3.

Virtual Machine/System Product Release 3 performance should be slightly better than that of Release 2 in most CMS environments. Guest operating systems should experience similar performance when running on VM/SP Release 3 as on Release 2.

Both CP and CMS have addressed performance in this release, as described in the “Enhancements for Improved CP and CMS Performance” sections. Reductions have been made in pathlength, storage, and I/O overhead for certain CMS commands.

Part Four: Internal Design Changes

This part of the book lists **changes** to the **internal design** of VM/SP that have performance, usability, or maintenance implications. It includes new and changed:

- **CMS Modules**
- **CP Modules**
- **Control blocks and macros**
- **Miscellaneous**

Module, Control Block, and Macro Changes

This section, listing changes to the **internal modules, control blocks, and macros**, may help you with planning. The functions are listed alphabetically. Each function is defined with:

- **New and Changed CMS modules**
- **New and Changed CP modules**
- **New and Changed control blocks and macros**
- **Miscellaneous**

General changes follow for:

- **CMS**
 - Deleted assemble files
 - Deleted EXECs
 - Deleted modules
 - Deleted XEDIT files
 - Files now in mode 1
 - Module splits
 - New modules
 - **CP**
 - Module splits
 - New copy files
 - New macro
 - New modules
-

Abend Exit

New and Changed CMS Modules

DMSABN DMSABX DMSDBG DMSFNC

New and Changed Control Blocks and Macros

ABNEXIT ABNXTCB DMSABW NUCON

CATCHECK Command

New and Changed CMS Modules

DMSCK DMSLAB DMSLDF

New and Changed Control Blocks and Macros

NONE

CMS File System Enhancement

New and Changed CMS Modules

DMSACC	DMSERD	DMSINI	DMSMOD	DMSSLN
DMSACM	DMSERS	DMSINT	DMSQRY	DMSSST
DMSBWR	DMSFNC	DMSLAD	DMSRSV	
DMSDIO	DMSFOR	DMSLST	DMSPRT	

New and Changed Control Blocks and Macros

ADT	FVS
-----	-----

CMSSEG Segment, Removal of

Changed CMS Modules

DMSABN	DMSFNC	DMSLOS	DMSSET	DMSTLB
DMSAMS	DMSGAM	DMSLSB	DMSSFF	DMSVIB
DMSBOP	DMSHLL	DMSINS	DMSSLN	
DMSBTP	DMSLDF	DMSITS	DMSSVN	
DMSDOS	DMSLDR	DMSNUC	DMSSVT	
DMSEDX	DMSLIB	DMSSAB	DMSSVU	

New and Changed Control Blocks and Macros

NUCON	SYSNAMES
-------	----------

Deleted CMS Assemble Files

DMSEXC Assemble	DMSSEG Assemble	DMSXSG Assemble
DMSOLD Assemble	DMSTLA Assemble	

Deleted CMS Modules

DMSEXT	DMSXFT	XEDIT	XEDMAIN
--------	--------	-------	---------

Deleted EXEC

CMSXGEN

Miscellaneous

MSGEND	CMSLOAD	CMSLOADL
--------	---------	----------

Files that are now Mode 1 instead of Mode 2

DMSGAM	DMSLSY	DMSSEB	DMSSOP
DMSLGT	DMSOLD	DMSSCT	DMSSQS
DMSLIB	DMSSAB	DMSSFF	DMSSVN
DMSLOS	DMSSBS	DMSSLN	DMSSVT
DMSLSB	DMSSBD	DMSSMN	DMSSVU
DMSLSY	DMSSCT	DMSSQS	DMSTLB

Communication Unit, 3088 Multisystem Channel Support

New and Changed CP Modules

DMKDEF	DMKDSP	DMKVCA	DMKVDE	DMKVIO
DMKDEG	DMKEMA	DMKVDA	DMKVDS	
DMKDIB	DMKQVM	DMKVDC	DMKVSI	
DMKDIR	DMKSCN	DMKVDD	DMKVSJ	

New and Changed Control Blocks and Macros

DEVTYPE\$ RCTLUNIT RDEVICE VBLOKS VCTA

CP DASD BLOCK I/O System Service

New and Changed CP Modules

DMSBIO DMKCFQ

New and Changed Control Blocks and Macros

BIOBLOK VMBLK VMBLOK

CP, Other Enhancements to

New and Changed CP Modules

DMKDRD DMKHVD DMKIUA DMKVIO
DMKFPS DMKHVE DMKVER DMKVMC

New and Changed Control Blocks and Macros

NONE

DASD Block I/O System Service, CMS Support for

New and Changed CMS Modules

DMSALU DMSDID DMSFLE DMSMVE DMSSOP
DMSBWR DMSFLD DMSLBD DMSSAB DMSSVT

New and Changed Control Blocks and Macros

CMSCB DISKID

DEFAULTS Command

New and Changed CMS Modules

NONE

New and Changed Control Blocks and Macros

NONE

Miscellaneous

DEFAULTS EXEC

Diagnose Code X'00'

New and Changed CP Modules

DMKHVD

New and Changed Control Blocks and Macros

NONE

Diagnose Code X'8C'

New and Changed CP Modules

DMKGRC	DMKHVD	DMKRGA	DMKVCT
DMKHVC	DMKHVE	DMKVCP	

New and Changed Control Blocks and Macros

RBLOKS RDEVICE NETWORK

Diagnose Code X'14'

New and Changed CP Modules

DMKCQH	DMKCSU	DMKMIA	DMKSPT	DMKWRM
DMKCST	DMKCSV	DMKSPS	DMKVIO	

New and Changed Control Blocks and Macros

None

DMSDOS Split

New and Changed CMS Modules

DMSDOS	DMMSGVE	DMSMCM	DMSSTX	DMSSVL
DMSGMF	DMSLDF	DMSRPG	DMSSUB	DMSVIS
DMSGTM	DMSLIC			

New and Changed Control Blocks and Macros

SVCWORK

MISCELLANEOUS

DOSGEN EXEC

DMSQRY Split

New and Changed CMS Modules

DMSFNC	DMSQRT	DMSQRV	DMSQRX	DMSQRZ
DMSQRS	DMSQRU	DMSQRW	DMSQRY	

Deleted CMS Module

QUERY

New and Changed Control Blocks and Macros

QRYWORK

End-of-Command Nucleus Extensions

New and Changed CMS Modules

DMSINT	DMSITS	DMSNXL	DMSNXM
--------	--------	--------	--------

New and Changed Control Blocks and Macros

NUCON SCBLOCK

EXECOS command

New and Changed CMS Modules

DMSCPF DMSFNC DMSSTG

New and Changed Control Blocks and Macros

NONE

GLOVALV Enhancement

New and Changed CMS Modules

DMSGLO

New and Changed Control Blocks and Macros

NONE

Immediate Commands, Creating Your Own

New and Changed CMS Modules

DMSABN DMSFNC DMSINT DMSNXL DMSSET
DMSCIT DMSIMM DMSITS DMSNXM DMSQRT
DMSEXE

New and Changed Control Blocks and Macros

IMMBLOK IMMCMD IMWKSECT NUCON SCBLOCK

IUCV, CMS Support of

New and Changed CMS Modules

DMSABN DMSFNC DMSITE DMSIUC DMSNUC

New and Changed Control Blocks and Macros

CMSIUCV HNDIUCV IUCVIDBK IUCVPTBK IUCVTAB
NUCON

IUCV, CP Enhancements to

New and Changed CP Modules

DMKIUA DMKIUC DMKIUE DMKIUG

New and Changed Control Blocks and Macros

IUCV IUCVBLOK IPARML

Larger-than-4K-Buffer Terminals, Support for

New and Changed CP Modules

DMKCFQ DMKQCN DMKRGB DMKVCN DMKVCV
DMKGRF DMKRGF DMKRGD DMKVCR DMKVDR

New and Changed Control Blocks and Macros

IOBLOKS

LISTFILE and RENAME Migration

New and Changed CMS Modules

DMSLST DMSRNM

New and Changed Control Blocks and Macros

NONE

Deleted CMS Modules

LISTFILE RENAME

Nucleus Reduction

New and Changed CP Modules

DMKBSC	DMKDID	DMKSCN	DMKTTY	DMKVSJ
DMKCPX	DMKIOS	DMKSCO	DMKTTZ	DMKVSP
DMKCVT	DMKIOT	DMKSSS	DMKVCN	DMKVSU
DMKCVU	DMKLOC	DMKSST	DMKVCX	DMKVST
DMKDGD	DMKQCN	DMKSSU	DMKVDA	DMKVSU
DMKDGF	DMKQCO	DMKSYM	DMKVDS	DMKVSU
DMKDGP	DMKQCP	DMKTBN	DMKVMA	

New and Changed Control Blocks and Macros

CALL

Previously Announced Functions

These code changes have been shipped as part of Release 2.

Printer Support

New and Changed CMS Modules

DMSASN DMSPRN

New and Changed CP Modules

DMKCFE	DMKCSO	DMKIOE	DMKRSP	DMKVSQ
DMKCFG	DMKDEF	DMKIOF	DMKSEP	DMKVST
DMKCKP	DMKDIR	DMKIOJ	DMKSPS	DMKVSU
DMKCKV	DMKEMA	DMKIOS	DMKSSP	DMKWWM
DMKCPB	DMKEMB	DMKIOT	DMKURS	
DMKCSB	DMKFCB	DMKMSW	DMKVDR	
DMKCSC	DMKIOC	DMKRSE	DMKVSP	

New and Changed Control Blocks and Macros

DEVTYPE	LDBLOK	RDEVICE	VBLOKS
IOER	RBLOKS	SPOOL	

Program Event Recording (PER)

New and Changed CP Modules

DMKCDM	DMKEMB	DMKPEQ	DMKPRW	DMKTRC
DMKCFE	DMKEMA	DMKPER	DMKQCO	DMKTRD
DMKCFJ	DMKPEI	DMKPET	DMKSVC	DMKUSO
DMKCFP	DMKPEL	DMKPRG	DMKTMR	DMKVAT
DMKDSP	DMKPEN	DMKPRV	DMKTRA	DMKVAU

New and Changed Control Blocks and Macros

PERBLOK PESBLOK PEXBLOK VMBLOK

Programmable Operator

New and Changed CMS Modules

DMSPOL DMSPOP DMSPOQ DMSPOR DMSPOS

New and Changed Control Blocks and Macros

PROP

Miscellaneous

CMSEND EXEC PROPLGER EXEC PROPRTCV EXEC
PROP RTABLE PROPPCHK EXEC PROPST EXEC
PROPHCHK EXEC

QSAM Tape End-of-Volume Exit

New and Changed CMS Modules

DMSFNC DMSST DMSSEB DMSSOP

New and Changed Control Blocks and Macros

CMSCB TEOVEXIT

Reading Directly from Your Virtual Machine Console

New and Changed CMS Modules

DMSCIT DMSCRD

New and Changed Control Blocks and Macros

NUCON RDTERM

SCBLOCK Look-Aside Buffer, Reduction of

New and Changed CMS Modules

DMSITS

New and Changed Control Blocks and Macros

NONE

SET EXECRAC Command

New and Changed CMS Modules

DMSSET DMSQRT

New and Changed Control Blocks and Macros

NUCON

SET LOGMSG Command

New and Changed CP Modules

DMKCFU DMKCQY DMKLOH

New and Changed Control Blocks and Macros

LOGMBLOK

SET QDROP Command

New and Changed CP Modules

DMKDSP DMKSCH

New and Changed Control Blocks and Macros

NONE

SQL/DS

These modules are listed separately under each function.

Symptom Record

New and Changed CP Modules

DMKDMP DMKVMD

New and Changed Control Blocks and Macros

NONE

Miscellaneous

DMPEQU

System Product Editor

New and Changed CMS Modules

DMSXBG	DMSXER	DMSXMA	DMSXQR	DMSXSU
DSSXCG	DMSXFC	DMSXMC	DMSXRE	DMSXTB
DMSXCM	DMSXFD	DMSXMD	DMSXSC	DMSXTE
DMSXCN	DMSXFL	DMSXML	DMSXSD	DMSXTF
DMSXCT	DMSXGT	DMSXMS	DMSXSE	DMSXTR
DMSXDC	DMSXHL	DMSXPO	DMSXSF	DMSXUP
DMSXDS	DMSXIN	DMSXPT	DMSXSS	DMSXWS
DMSXED	DMSXIO	DMSXPX	DMSXST	

New and Changed Control Blocks and Macros

ALL	HEXTYPE	PREFIXX	RGLEFT	STATUS
CANCEL	JOIN	PRFSHIFT	SCHANGE	ZBLOCKS
CAPPEND	LSCREEN	PRFSHOW	SORT	ZDESTYP
FLTYP	MODIFY	PRFSHOW	SPLIT	ZMACST
HELPEXED	PARSE	PRSCB	SPLTJOIN	

Deleted EXECs

EDGAR EXEC CNV\$PROF EXEC

Deleted CMS XEDIT Files

\$EDGAR XEDIT CNV\$PROF XEDIT EDGAR XEDIT

Deleted Module

ECOMMAND

XEDIT EXECs Changed for CMS in general

CMSGEND EXEC	CMSLOAD EXEC	CMSLOADL EXEC*
DEFAULTS EXEC	DISCARD EXEC	DMSSP3 EXEC
DOSGEN EXEC	EXECUPDT EXEC	FILELIST EXEC
NAMES EXEC	NOTE EXEC	PEEK EXEC
PROPHCHK EXEC	PROPLGER EXEC	PROPPCHK EXEC
PROPRTCV EXEC	PROPST EXEC	RDRLIST EXEC
RECEIVE EXEC	SENDFILE EXEC	TELL EXEC
VSAMGEN EXEC	VSEVSAM EXEC	PROFFLST XEDIT
PROFNOTE XEDIT	PROFPEEK XEDIT	PROFRLST XEDIT
PROFSEND XEDIT	EXECUTE XEDIT	HELPEXED XEDIT
X\$FLST\$X XEDIT	X\$NAME\$X XEDIT	X\$PEEK\$X XEDIT
X\$SEND\$X XEDIT		

System Product Interpreter

New and Changed CMS Modules

DMSRCN	DMSREX	DMSRIN	DMSRTC	DMSRVA
DMSREV	DMSRFN	DMSRSF	DMSRXE	

New and Changed Control Blocks and Macros

NUCON

Tape Support

New and Changed CMS Modules

DMSASN DMSTIO

New and Changed CP Modules

DMKCCW	DMKIOC	DMKIOJ	DMKSSP	DMKTAP
DMKDDR	DMKIOE	DMKMCC	DMKSST	DMKVER
DMKDMP	DMKIOF	DMKMSW		

New and Changed Control Blocks and Macros

DEVTPES	IOER	RCTLUNIT	RDEVICE	SDRBLOK
DMSTAPE	OBRRECN			

TELL Command

New and Changed CMS Modules

NONE

New and Changed Control Blocks and Macros

NONE

Miscellaneous

TELL EXEC
DEFAULTS EXEC

VSAM Support of the 3380 Direct Access Storage

New and Changed CMS Modules

DMSASN DMSBOP DMSGVE DMSVVL

New and Changed Control Blocks and Macros

NONE

VSE/VSAM Release 3 Support

New and Changed CMS Modules

DMSVVN

New and Changed Control Blocks and Macros

CMSVSAM DOSLNK

VSE/VSAM, Support for Assembler Language Macros

New and Changed CMS Modules

DMSDMP DMSDOS DMSVIP DMSVVN DMSVVS

New and Changed Control Blocks and Macros

IKQACB IKQEXLST IKQRPL

New VSE Macros

CDLOAD CLOSER OPEN PUT
CLOSE GET OPENR

Miscellaneous

CMSVSAS DOSLNK VSEVSAM EXEC
OSVSAM MACLIB VSEVSAM SCAN

Wait on ECB

New and Changed CMS Modules

DMSCIT DMSCRD DMSFNC DMSSVN DMSWTE

New and Changed Control Blocks and Macros

DEVTAB NUCON WAITECB

General Changes Follow For:

CMS Module Splits

DMSDOS has been split into:

DMSDOS	DMSVIS	DMSGTM	DMSMCM
DMSLDF	DMSGVE	DMSSVL	DMSLIC
DMSSTX	DMSGMF	DMSRPG	DMSSUB

DMSQRY had been split into:

DMSQRY	DMSQRT	DMSQRV	DMSQRX
DMSQRS	DMSQRU	DMSQRW	DMSQRZ

DMSXSE has been split into:

DMSXSE	DMSXQR	DMSXSF
--------	--------	--------

DMSXBG has been split into:

DMSXBG	DMSXWS
--------	--------

DMSPOP has been split into:

DMSPOP	DMSPQQ
--------	--------

DMSPOR has been split into:

DMSPOR	DMSPQS
--------	--------

CMS Deleted Assemble Files

DMSEXC Assemble	DMSSEG Assemble	DMSXSG Assemble
DMSOLD Assemble	DMSTLA Assemble	

CMS Deleted EXECs

CMSXGEN EXEC	EDGAR EXEC
CNV\$PROF EXEC	

CMS Deleted Modules

DMSEXT	ECOMMAND	RENAME	XEDIT
DMSXFT	LISTFILE	QUERY	XEDMAIN

CMS Deleted XEDIT Files

\$EDGAR XEDIT	CNV\$PROF XEDIT	EDGAR XEDIT
---------------	-----------------	-------------

Files that are now Mode 1 instead of Mode 2

DMSGAM	DMSLSY	DMSSEB	DMSSOP	DMSTLB
DMSLGT	DMSOLD	DMSSCT	DMSSQS	
DMSLIB	DMSAB	DMSSFF	DMSSVN	
DMSLOS	DMSBS	DMSSLN	DMSSVT	
DMSLSB	DMSBD	DMSSMN	DMSSVU	
DMSLSY	DMSSCT	DMSSQS		

New CMS Modules

DMSABX	DMSIUC	DMSRFN	DMSRVA	DMSWTE
DMSCCK	DMSRCN	DMSRIN	DMSRXE	DMSXTE
DMSDID	DMSREV	DMSRSF	DMSRSV	DMSXTR
DMSIMM	DMSREX	DMSRTC		

CP Module Splits

New CP modules have been split from existing CP modules:

New Module	Existing Module
DMKCSC	DMKCSB
DMKCVU	DMKCVT
DMKIUJ	DMKIUC
DMKIUL	DMKIUE
DMKQCP	DMKQCO
DMKRST	DMKRSP
DMKSCO	DMKSCN
DMKSPK	DMKSPL
DMKTTZ	DMKTTY
DMKVSU	DMKVSP
DMKVSU	DMKVSP
DMKVSV	DMKVSR
DMKWRN	DMKWRM

Figure 6. CP Module Splits

New CP Copy Files

BIOBLOK	LOGMBLOK
LDBLOK	PERBLOKS

New CP Macro

TTYDATA

New CP Modules

DMKBIO	DMKPEL	DMKPEQ
DMKPEI	DMKPEN	DMKPET

Index

A

ABEND exit 53
ALL macro 33
AP/MP processor unit 70
ARBCHAR alone in CHANGE 41
availability enhancements, programmable operator 28

B

Block I/O system service support 55

C

CATCHECK command 61
channel communication unit 66
CLOCATE 41
CMS enhancements
 application programs 52-56
 ABEND exit 53
 CMSIUCV macro 53
 CP Block I/O system service support 55
 creating your own immediate commands 54
 DISKID function 56
 FILEDEF command extension 55
 HNDIUCV macro 53
 IUCV support 52
 nucleus extension to end-of-command 53
 QSAM tape end-of-volume exit 55
 reading directly from the virtual machine console 54
 wait on ECB 53
 CMS file system enhancement 56
 RESERVE command 56
 update-in-place 56
 512-byte blocksize minidisk 56
 improved CMS performance 57-58
 CMSSEG segment removal 57
 GLOBALV enhancement 57
 LISTFILE migration 57
 RENAME migration 57
 SCBLOCK look-aside buffer reduction 57
 other CMS enhancements 58-59
 DEFAULTS command 58
 EXECOS command 58
 SET EXECTRAC command 59
 TELL command 58
 CMS IUCV support 52
 CMS module splits 88
 CMS/VSAM support of the 3380 direct access storage 62
 CMSIUCV macro 53
 CMSSEG segment removal 57
 color and extended highlighting 30
 command line 36
 command line arrow 41
 communication unit 66
 control block changes 78-89
 CP Block I/O system service support 55
 CP enhancements
 application programs 47-49

DASD Block I/O system service 48-49
 Inter-User Communication Vehicle (IUCV) enhancements 47-48
 improved CP performance 49
 larger-than-4K buffer terminal support 50
 nucleus reduction 49
 SET QDROP command 50
 other CP enhancements 50
 DIAGNOSE code X'00' 50
 DIAGNOSE code X'14' 51
 DIAGNOSE code X'8C' 50
 SET LOGMSG command 51
 CP module splits 89
 creating your own immediate commands 54
 CURSOR HOME subcommand 37

D

DASD Block I/O system service 48
DEFAULTS command 58
Deleted CMS Assemble Files 88
Deleted CMS EXECs 88
Deleted CMS Modules 88
Deleted CMS XEDIT Files 88
DIAGNOSE code X'00' 50
DIAGNOSE code X'14' 51
DIAGNOSE code X'8C' 50
DISKID function 56
DMSDOS split 45
DMSQRY split 45

E

EDGAR support removal 71
end-of-command nucleus extension 53
enhanced text comparison 28
enhancements in Release 3
 list of 5-6
EREP Release 2.3 8
error handling enhancements 28
EXEC action routines 28
EXECOS command 58
EXTRACT subcommand 38

F

FILEDEF command extension 55
FILEIST command 37
Files that are now Mode 1 instead of Mode 2 88
filetype defaults 40

G

GLOBALV enhancement 57

H

hardware support
 channel communication unit support 66
 multisystem channel communication unit 66
 3088 66
 printer support 65
 3262 printer, model 5 65
 3800 printing subsystem models 3 and 8 65
 4245 line printer 65
 4250 printer 65
 tape support 65
 3430 magnetic tape subsystem 66
HELP command 37
HNDIUCV macro 53
horizontal EOL 41
horizontal TOL 41

I

improved cursor handling 39
Inter-User Communication Vehicle (IUCV)
 enhancements 47
interactive SQL (ISQL) 23
introduction 3
IPCS incompatibility 71
IPCS/E Release 2 8
IUCV enhancements 47

L

larger-than-4K buffer terminal support 50
LASTLORC 39
LINEND 41
LISTFILE migration 57
LOCATE 41
logging 28
LPREFIX subcommand 32

M

MERGE 38
message recording enhancements 28
message routing with nicknames 27
middle of screen 36
migration aids
 AP/MP processor unit 70
 programmable operator routing tables 70
 PROPRTCV EXEC 70
 SPOOL file 70
migration compatibility 71
migration incompatibility 71
 EDGAR support removal 71
 IPCS incompatibility 71
 RSCS incompatibility 72
 XEDIT migration consideration 72
 column pointer movement 73
 cursor handling 74
 filetype defaults 74
 new tags on READ TAG 74
 null string target in CLOCATE 74
 pending notice 73
 PF key changes 72
 Protected FILE and SAVE 73

QUIT from a macro 74
 screen changes 74
 SET CMDLINE 73
missing interrupt handler support 63
module changes 78-89
multisystem channel communication unit 66

N

NAMES command 37
new CMS modules 89
new CP copy files 89
new CP macro 89
new CP modules 89
new facilities in Release 3
 list of 5-6
NOMsg option 40
NOTE command 37
nucleus reduction 49

O

OS/VSAM macros 61
overview of Release 3 2

P

PEEK command 37
PENDING 31
PER command 43
performance 76
PF key changes 36
PFILE 40
prefix macro support 31
prefix macros 31
previously announced functions
 missing interrupt handler support 63-64
 speed matching buffer support for IBM 3375 64
printer support 65
 3262 printer, model 5 65
 3800 printing subsystem models 3 and 8 65
 4245 line printer 65
 4250 printer 65
program event recording (PER) support 43
programmable operator facility 4, 27-29
 availability enhancements 28
 enhanced text comparison 28
 error handling enhancements 28
 error recording enhancements 28
 EXEC action routines 28
 logging 28
 message recording enhancements 28
 message routing with nicknames 27
 PROPRTCV EXEC 29
 routing table format 29
programmable operator routing tables 70
programming requirements 8
 EREP Release 2.3 8
 IPCS/E Release 2 8
 SQL/DS Release 2 8
 VSAM Release 3 8
PROPRTCV EXEC 70
PSAVE 40

Q

QSAM tape end-of-volume exit 55
QUERY PF 41
QUERY PREFIX SYNONYM 32
QUIT/PURGE 41

R

RDRLIST command 38
reading directly from the virtual machine console 54
REFRESH 41
RENAME migration 57
RESERVE command 56
REXX 24
RGTLLEFT 36
RSCS incompatibility 72

S

S (Show) 31
SCBLOCK look-aside buffer reduction 57
SCHANG 40
screen control enhancements 34
selective line editing 33
SENDFILE command 38
serviceability enhancements
 DMSDOS split 45
 DMSQRY split 45
 program event recording (PER) support 43-44
 symptom record (VM/IPCS/E) 44-45
SET COLOR 30
SET CURLINE 36
SET DISPLAY 33, 34
SET EXETRAC command 59
SET FULLREAD 39
SET LOGMSG 51
SET MSGLINE 36
SET PREFIX SYNONYM 32
SET QDROP 50
SET REMOTE 39
SET RESERVED 36
SET SCALE 36
SET SCOPE 33, 34
SET SELECT 33, 34
SET SHADOW 33, 34
SET SIDCODE 39
SET SPILL 39
SET STAY 41
SET TABLINE 36
SET TRANSLAT 39
SET VERIFY 41
shift left 31
shift right 31
significant changes to Release 3 3
 programmable operator facility 4
 SQL/DS 3
 System Product Editor 4
 System Product Interpreter 3
special direction for products 8
 System Product Editor 9
 System Product Interpreter 9
speed matching buffer support for IBM 3375 64
SPLTJOIN 37
SPOOL file 70
SQL/DS 3
SQL/DS Release 2 8

statement of system integrity 10-11
STATUS/MODIFY changes 41
Structured Query Language/Data System (SQL/DS) 22-23
 Data Base Services (DBS) utility 22
 data base support 22
 SQL/DS Release 2 22
 VM/SP support for SQL 23
summary of the new facilities and enhancements 5
symptom record (VM/IPCS/E) 44
system integrity 10-11
 directory capabilities 10
 main storage protection 10
 password protected resource 10
 responsibilities of the installation 11
 secondary storage protection 10
System Product Editor 4
System Product Editor enhancements 30-41
 additional new functions 38-39
 EXTRACT subcommand 38
 SET ALT 39
 SET ENTER 39
 SET FULLREAD 39
 SET LASTLORC 39
 SET options 39
 SET PAn 39
 SET REMOTE 39
 SET SIDCODE 39
 SET SPILL 39
 SET TRANSLAT 39
color and extended highlighting 30
 SET COLOR 30
modifications to existing functions 39-41
 ARBCHAR alone in CHANGE 41
 CLOCATE 41
 filetype defaults 40
 horizontal TOL and EOL 41
 improved cursor handling 39
 LINEND 41
 LOCATE 41
 logical AND operator 40
 multiple symbolic names 40
 NOMsg option 40
 nulls in prefix area 40, 42
 number of strings in target 40
 PFILE 40
 protected FILE and SAVE 40
 PSAVE 40
 QUERY PF 41
 QUIT/PURGE 41
 REFRESH 41
 SCHANG 40
 SET STAY ON/OFF 41
 SET VERIFY 41
 SPLIT and JOIN 40
 STATUS/MODIFY changes 41
 targets 40
 UNtil option 40
 WRAP warning 41
 XEDIT command 40
PF key changes 36
 CURSOR HOME subcommand 37
 FILEIST command 37
 HELP command 37
 NAMES command 37
 NOTE command 37
 PEEK command 37
 RDRLIST command 38
 RGTLLEFT 36
 SENDFILE command 38
 SPLTJOIN 37

- prefix macro support 31-33
 - < (shift left) 31
 - > (shift right) 31
 - LPREFIX subcommand 32
 - PENDING 31
 - prefix macros 31
 - QUERY PREFIX SYNONYM 32
 - S (Show) 31
 - SET PREFIX SYNONYM 32
 - writing prefix macros 31
 - X (Exclude) 31
- screen control enhancements 34
 - CMS commands 37
 - command line 36
 - middle of screen 36
 - SET CURLINE 36
 - SET MSGLINE 36
 - SET RESERVED 36
 - SET SCALE 36
 - SET TABLINE 36
 - summary of PF key changes 38
 - vertical screen split 34, 36
 - XEDIT PF key changes 36
- selective line editing 33, 34
 - ALL macro 33
 - SET DISPLAY 33, 34
 - SET SCOPE 33, 34
 - SET SELECT 33
 - SET SHADOW 33, 34
- System Product Editor, special direction 9
- System Product Interpreter 3, 24-26
 - advantages 24
 - examples 25
 - REXX 24
- System Product Interpreter, special direction 9
- System/370 Program Event Recording (PER) support 43

T

- tape support 65
- TELL command 58

U

- UNtil option 40
- update-in-place 56

V

- vertical screen split 34
- VM/SP CMS User's Guide 19
- VM/SP Data Areas and Control Block Logic 19
- VM/SP Distributed Data Processing Guide 18
- VM/SP General Information 19
- VM/SP Installation Guide 18
- VM/SP Introduction 19
- VM/SP library
 - new books 18
 - VM/SP Distributed Data Processing Guide 18
 - VM/SP Installation Guide 18
 - VM/SP Release 3 Guide 18
 - VM/SP System Product Interpreter Reference 18
 - VM/SP System Product Interpreter User's Guide 18

- new look 15-17
 - bill of forms number 16
 - cover design 15
- restructured books 19-20
 - VM/SP CMS User's Guide 19
 - VM/SP Data Areas and Control Block Logic 19
 - VM/SP General Information 19
 - VM/SP Introduction 19
 - VM/SP Planning and System Reference 19
 - VM/SP Planning Guide and Reference 19
 - VM/SP Terminal Reference 19
 - VM/SP Planning and System Generation Guide 19
 - VM/SP Planning Guide and Reference 19
 - VM/SP Release 3 Guide 18
 - VM/SP System Product Interpreter Reference 18
 - VM/SP System Product Interpreter User's Guide 18
 - VM/SP Terminal Reference 19
 - VM/SP Terminal User's Guide 19
 - VSAM Assembler language macros 61
 - VSAM support of the 3380 direct access storage 62
 - VSE/VSAM Release 3 support 60
 - VSE/VSAM macros 61
 - VSE/VSAM Release 3 8
 - VSE/VSAM support enhancements 60-62
 - CATCHECK command 61
 - CMS/VSAM support of the 3380 direct access storage 62
 - VSAM Assembler language macros 61
 - OS/VSAM macros 61
 - VSE/VSAM macros 61
 - VSEVSAM command 61
 - VSE/VSAME Release 3 support 60
 - VSEVSAM command 61

W

- wait on ECB 53
- WRAP Warning 41
- writing prefix macros 31

X

- X (Exclude) 31

3

- 3088 multisystem channel communication unit 66
- 3262 printer, model 5 65
- 3375 hardware, speed matching buffer support of 64
- 3430 magnetic tape subsystem 66
- 3800 printing subsystem models 3 and 8 65

4

- 4245 line printer 65
- 4250 printer 65

5

- 512-byte blocksize minidisk 56



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. Comments may be written in your own language; English is not required.

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.*

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

	Yes	No
• Does the publication meet your needs?	<input type="checkbox"/>	<input type="checkbox"/>
• Did you find the material:		
Easy to read and understand?	<input type="checkbox"/>	<input type="checkbox"/>
Organized for convenient use?	<input type="checkbox"/>	<input type="checkbox"/>
Complete?	<input type="checkbox"/>	<input type="checkbox"/>
Well illustrated?	<input type="checkbox"/>	<input type="checkbox"/>
Written for your technical level?	<input type="checkbox"/>	<input type="checkbox"/>
• What is your occupation?	_____	
• How do you use this publication:		
As an introduction to the subject?	<input type="checkbox"/>	As an instructor in class? <input type="checkbox"/>
For advanced knowledge of the subject?	<input type="checkbox"/>	As a student in class? <input type="checkbox"/>
To learn about operating procedures?	<input type="checkbox"/>	As a reference manual? <input type="checkbox"/>

Your comments:

If you would like a reply, please supply your name and address on the reverse side of this form.

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.)

Reader's Comment Form

Cut or Fold Along Line

Fold and Tape

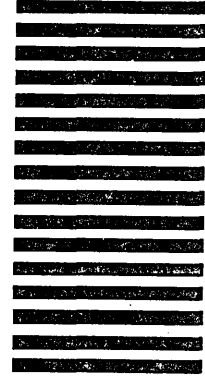
Please Do Not Staple

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 G60
P. O. Box 6
Endicott, New York 13760

Fold

Fold

If you would like a reply, *please print*:

Your Name _____
Company Name _____ Department _____
Street Address _____
City _____
State _____ Zip Code _____
IBM Branch Office serving you _____



VM/SP Release 3 Guide (File No. S370/4300-20) Printed in U.S.A. SC24-5240-0

SC24-5240-0

VM/SP Release 3 Guide (File No. S370/4300-20) Printed in U.S.A. SC24-5240-0

IBM[®]

SC24-5240-0

