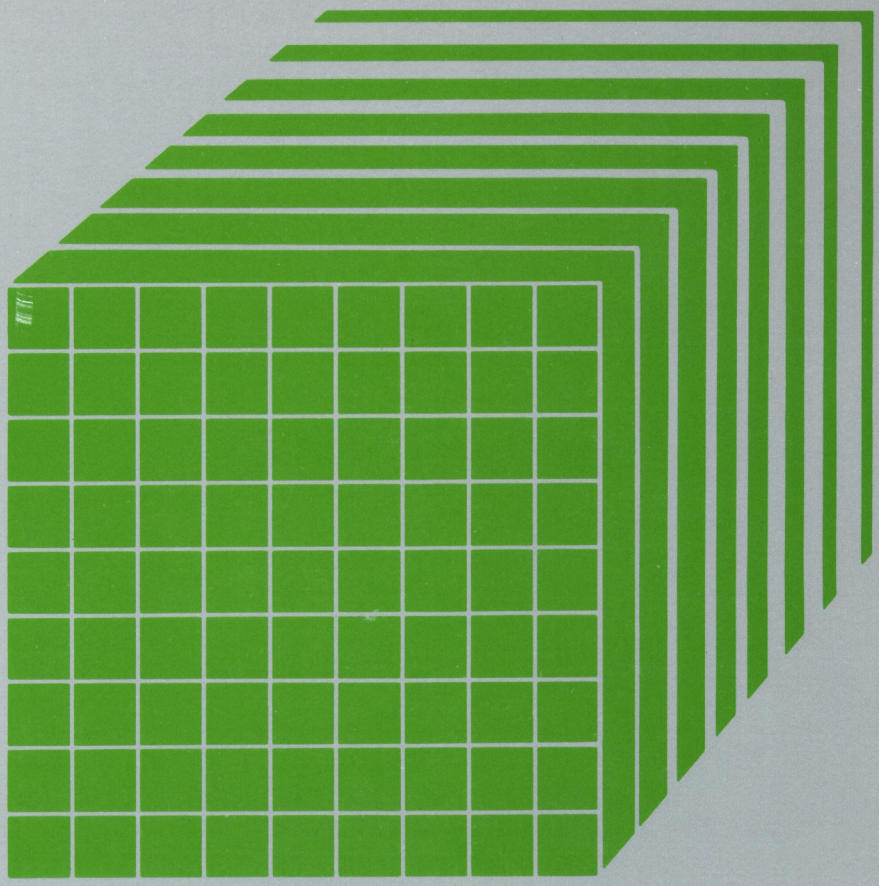
IBM

Virtual Machine/ System Product High Performance Option

Planning Guide and Reference

Release 4.2

SC19-6223-6



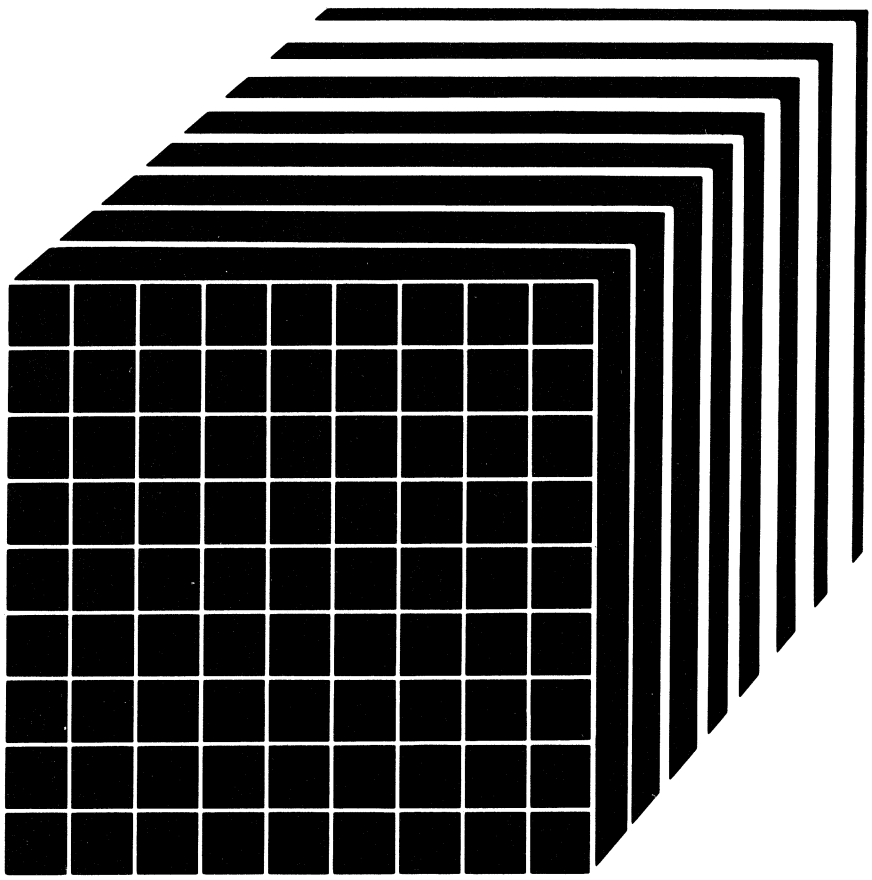
IBM

Virtual Machine/ System Product High Performance Option

Planning Guide and Reference

Release 4.2

SC19-6223-6



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

Seventh Edition (January 1986)

This is a major revision of SC19-6223-4 and SC19-6223-5. See "Summary of Changes" on page iii for the changes made to this manual. Technical changes or additions made to the text and illustrations for this release are indicated by a vertical line to the left of the change.

The seventh (current) edition applies to Release 4.2 of IBM Virtual Machine/System Product High Performance Option (Program Number 5664-173), and to later releases and modifications until otherwise indicated in new editions or Technical Newsletters. The seventh edition is a revision of the fifth edition (Release 3.6) and the sixth edition (Release 4.0).

To order previous editions that still apply to Releases 3.2, 3.4, 3.6, and 4.0 use the following temporary order numbers:

Release 3.2	Third Edition	ST00-1438
Release 3.4	Fourth Edition	ST00-1555
Release 3.6	Fifth Edition	ST00-1615
Release 4.0	Sixth Edition	ST00-1739

Changes are made periodically to the information herein; before using this publication to operate 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 at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, Department 52Q, Neighborhood Rd., Kingston, N.Y. 12401. IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 1983, 1986

Summary of Changes

Summary of Changes
for SC19-6223-6
As Updated January 1986
for VM/SP HPO Release 4.2

AUTODEACTIVATION OF RESTRICTED PASSWORDS AND DIRECTORY ENHANCEMENTS

New: Programming Support

Adds support to enhance system integrity by minimizing the exposure of unauthorized system access through the use of restricted passwords.

Directory enhancements include two new control statements, PROFILE and INCLUDE, which permit a group of control statements common to more than one user directory entry to be coded only once, and the removal of the directory size limit.

ACCESS VERIFICATION ROUTINES

New: Programming Support

Adds support, including a new directory control statement, ACIGROUP, for access verification routines. When used with the RACF/VM Support PRPQ, access verification routines help increase security for VM/HPO installations. The access verification routines support does not itself provide security functions; but it allows the installation to install software that does. To increase security of minidisk accesses, logon passwords, and movement of spool files, an installation should install access verification routines with the Resource Access Control Facility (RACF) (Program Number 5740-XXH) and RACF/VM Support PRPQ (Program Number 5767-002).

SECURITY ENHANCEMENTS

New: Programming Support

This support allows an installation to specify the number of password attempts that a user is allowed before being locked out of the system and the amount of time that the user will be locked out after the unsuccessful attempts. In addition, if an installation has installed RACF, this support now calls RACF to authorize the use of STCP and LINK commands.

VECTOR FACILITY

New: Hardware Support

Support is provided for the Vector Facility in System/370 mode configured to a 3090 Processor. The Vector Facility is a synchronous vector/scalar instruction processor that can manipulate values (usually floating-point) at high speed. Compiled engineering and scientific FORTRAN applications can use the array processing capability of the Vector Facility. VM/SP HPO supports the use of this facility by multiple virtual machines.

PAGE MIGRATION

Changed: Programming Support

Page migration is changed to select pages (rather than segments) for migration on a reference basis instead of by time-stamp (age basis). Also, pages are migrated down the demand page hierarchy, instead of being migrated directly to the pre-allocated migration area. This reduces the time required to retrieve those pages that become active in the near future.

Because migration of swap tables is sometimes necessary even when page migration is not actively moving pages, swap table migration is now invoked independently of page migration (rather than after page migration). Swap table migration is further improved by migrating swap tables regardless of whether all the pages in the segment have been migrated.

SYSPAG macro, commands, and monitor are enhanced to support the changed migration algorithms. Installations should retune free storage.

3380 DIRECT ACCESS STORAGE DEVICE MODELS AE4/BE4

New: Hardware Support

VM/SP HPO now supports the 3380 DASD Models AE4/BE4. The 3380 Models AE4/BE4 are count-key-data (CKD) devices that attach to high-speed channels only, via the 3880 Control Unit. The 3880 can attach up to 16 physical spindles (32 logical devices) of 3380 Models AE4 and BE4 directly to data streaming channels. The AE4 models attach to the system and may be the first device on a string. Strings of different 3380 device models may be intermixed at the control unit level.

DOCUMENTATION CHANGES

Minor technical and editorial changes have been made throughout this publication.

Summary of Changes
for SC19-6223-5
As Updated November 1985
for VM/SP HPO Release 4.0

Note: Release 4 does not *support 3090 processors. 3090 processors are supported by Release 3 Modification 6. For information on Release 3 Modification 6, order the* Planning Guide and Reference *using the pseudo-number ST00-1615.*

3880 MODELS 13 AND 23 SUPPORT

New: Hardware Support

VM/SP HPO now supports the 3880 Storage Subsystem Models 13 and 23. This support is available after HPO Release 4.0 general availability via program update tape. The 3880 Models 13 and 23 are high performance cached DASD subsystems designed especially for nonpaging applications (application data that resides on DASD that is not defined as a paging, swapping, spooling, or dump area). This support is designed to improve performance for MVS or CMS guest virtual machines running with 3380 DASDs. The performance improvement is obtained principally by maximizing the number of read accesses that can be resolved by accessing the cache copy rather than by accessing the DASD itself.

The support includes:

- Operator commands to enable/disable the caching function so that the cache may be taken off-line while support personnel work on it
- Recognition of 3380 Model 13 and 23 hardware errors
- Monitoring of records of cache use from both storage directors
- Control of the cache by a dedicated guest virtual machine.

PREFERRED MACHINE ASSIST EXTENSIONS: CONTROL SWITCH ASSIST

New: Programming Support

Control switch assist consists of extensions to preferred machine assist. This support is available at HPO Release 4.0 general availability via program update tape. This support provides the performance benefits of preferred machine assist and selected CP DIAGNOSE support available to virtual machines. It also helps users who are experiencing line timeout problems on preferred machine assist guest-owned devices attached to CP-owned channels.

GROUP CONTROL SYSTEM (VM/SP HPO GCS)

New: Programming Support

This new component of VM/SP HPO is a virtual machine supervisor that provides simulated MVS services and supports a multitasking environment. For more information on the Group Control System (GCS), refer to the *VM/SP Group Control System Guide.*

SIGNAL SYSTEM SERVICE

New: Programming Support

This new CP system service allows virtual machines in a virtual machine group to signal each other. The signal system service can only be used by virtual machines in a virtual machine group.

SAVED SYSTEM 8M LIMIT REMOVAL

Changed: Programming Support

With the addition of this support, the SAVESYS, VMSAVE, and IPL functions have been enhanced to allow a page image copy of up to a 16M virtual machine to be saved and restored.

CP FRET TRAP

New: Programming Support

The CP FRET trap can be used as an aid in solving problems caused by improper use of CP storage and to solve many storage overlay problems.

VMDUMP ENHANCEMENTS

Changed: Programming Support

DIAGNOSE Code X'94' is available to allow a virtual machine to request dumping of its virtual storage. Also, the three address range restriction has been removed from the VMDUMP command.

DIAGNOSE CODE X'98'

New: Programming Support

Using DIAGNOSE Code X'98', a virtual machine can lock and unlock virtual pages, and execute its own real channel programs.

THE PROGRAMMABLE OPERATOR FACILITY

Changed: Programming Support

The programmable operator facility has been enhanced to support distributed operations in an SNA network through an interface, the Programmable Operator/NCCF Message Exchange (PMX) with the Network Communications Control Facility (NCCF). The VM/SP Release 4 programmable operator:

- Allows an NCCF operator to be identified to the programmable operator so that any messages intended for the logical operator may be routed to that NCCF operator
- Allows an NCCF operator to issue programmable operator commands and receive responses
- Provides the LGLOPR command for assigning, releasing and replacing the logical operator during operation.

CPTRAP ENHANCEMENTS

Changed: Programming Support

CPTRAP is a service aid used in problem determination. Enhancements to the CPTRAP command provide two additional functions, GROUPID and WRAP, and one additional entry type, X'3D'.

Enhancements to TRAPRED makes reviewing the trap data easier by providing more selectivity for X'3D', X'3E', and X'3F' entries and by providing a way to display formatted output of the trapped data.

INTERACTIVE PROBLEM CONTROL SYSTEM (VM/SP IPCS)

New: Programming Support

VM/SP HPO Release 4 has been enhanced to include IPCS as a component of VM/SP. VM/SP IPCS is equivalent to the VM/Interactive Problem Control System Extension (VM/IPCS/E) Licensed Program Product (5748-SA1).

EXPANSION OF USER CLASSES

Changed: Programming Support

The DIRECT command has been enhanced and the OVERRIDE command has been added to provide the user with more than the seven IBM defined user classes. You can now choose from 32 user classes, A - Z, and 1 - 6.

REMOTE SPOOLING COMMUNICATIONS SUBSYSTEM NETWORKING VERSION 2

Changed: Programming Support

With the release of the Remote Spooling Communications Subsystem Networking Version 2 program product (5664-188), any reference to RSCS in this manual applies to RSCS Version 2. Information about RSCS can be found in the *VM/SP Remote Spooling Communications Subsystem Version 2 General Information.*

VM/SP HPO 3800 MODEL 3 COMPATIBILITY SUPPORT

New: Hardware Support

Compatibility support allows VM/SP users to access the 3800 Model 3 Printing Subsystem. Existing programs designed to produce 3800 Model 1 printer output may produce output for the 3800 Model 3 printer with little or no program change. Use of this support provides improved print quality (240 x 240 pel resolution) and the addition of a 10 lines-per-inch (LPI) vertical space option.

DIAGNOSE CODE X'8C'

Changed: Programming Support

DIAGNOSE code X'8C' has been enhanced to allow a user to access all the data returned by CP's WRITE STRUCTURED FIELD QUERY.

3880 STORAGE SUBSYSTEM MODEL 21 SUPPORT

New: Hardware Support

VM/SP HPO now supports the 3880 Storage Subsystem Model 21. The 3880 Model 21 is designed as a high-performance paging and swapping subsystem. It is an enhancement over the 3880 Model 11. In addition, the 3880 Storage Subsystem Model 11 is supported for swapping.

4381 MODEL 3

New: Hardware Support

VM/SP HPO now supports the 4381 Processor Model 3 (dyadic). ECPS:VM/370 level 22 specifically supports the 4381 Model 3. It includes:

- Assists for VM/SP HPO free storage and dispatching algorithms and a CP assist for DIAGNOSE code X'18' (Disk I/O).
- Enhancements to existing assists to support the extended real addressing facility of VM/SP HPO.
- Enhancements to existing assists to support a multiprocessor system.

VIRTUAL DEVICE LIMIT EXTENSION

Changed: Programming Support

The number of virtual devices that can be attached to a virtual machine has been increased.

DOCUMENTATION CHANGES

Minor technical and editorial changes have been made throughout this publication.

Summary of Changes
for SC19-6223-4
As Updated November 6, 1984
for VM/SP HPO Release 3.6

VM/SP HIGH PERFORMANCE OPTION RELEASE 3.6

New: Expanded Programming Support

VM/SP Release 3, or an equivalent program product, is the prerequisite program product for VM/SP High Performance Option Release 3.6. VM/SP High Performance Option Release 3.6 operates with VM/SP Release 3 and incorporates the new and expanded programming facilities, features and support provided by VM/SP Release 3.

3090 PROCESSOR COMPLEX SUPPORT

New: Hardware Support

This release supports the 3090 Processor Complex Model 200 as a dyadic processor.

4381 PROCESSOR SUPPORT

New: Hardware Support

This release supports the 4381 Processor Model 3 as a dyadic processor.

EXTENDED CHANNEL SUPPORT

New: Programming Support

Extended channel support lets an installation configure its system resources over 48 channels for a 3090 Processor Complex. As a result, many CP commands, messages, and macros now accept or return a four-digit real device address.

EXPANDED STORAGE (PAGING STORAGE) SUPPORT

New: Programming Support

Expanded Storage, also called Paging Storage, is optional storage that can be installed on the 3090 Processor Complex as a paging assist. Paging Storage support lets an installation use this storage as a high-speed, system-owned paging area. Monitor records, the DASD Dump Restore program, the page migration routine, several commands, and the SYSPAG macro have been enhanced for this support.

CONTROL SWITCH ASSIST EXTENSIONS TO PREFERRED MACHINE ASSIST

New: Programming Support

This support, available on 308X, 4381, and 3090 processors, allows an MVS/SP V=R virtual machine guest (Release 1 Enhancement or later) to use IUCV, many DIAGNOSE instructions, and some Service Call instructions. It also reduces line timeout problems for such guests by letting CP reflect virtual I/O interruptions to the guest.

IBM 3880 MODEL 21 STORAGE SUBSYSTEM

New: Programming Support

The IBM 3880 Model 21 Storage Subsystem is similar to the Model 11, but has a larger cache, two storage directors, and improved data transfer. The Model 21 has a different addressing scheme than the Model 11. Support for this device modifies block paging by providing sequential access. Now both the Model 11 and the Model 21 can use block paging.

DOCUMENTATION CHANGES

Various technical and editorial changes have been made throughout the publication.

Preface

This publication is intended for system programmers and those responsible for the planning and installation of a VM/SP High Performance Option (VM/SP HPO) system. It provides information to assist you in the planning of your VM/SP HPO system.

You should have a general understanding of System/370 data processing techniques, teleprocessing techniques, and virtual machine concepts.

Summary of Contents

This publication has two parts, plus appendixes.

"Part 1. Planning for System Generation" describes the components, features, and options of VM/SP HPO, and tells you what you must do during system generation to support them.

"Part 2. Defining Your VM/SP HPO System" tells you how to create the files that define your system.

The appendixes include information about:

- Program products, language processors and emulators
- Configuring VM/SP HPO
- Compatible devices
- VM/SP HPO restrictions
- Worksheet to aid in coding the NAMESYS macro
- Sample SNTMAP output.

Corequisite Publications

IBM Virtual Machine/System Product High Performance Option (VM/SP HPO)

Commands (General User), SX22-0003

Commands (Other than General User), SX22-0004

CP Command Reference for General Users, SC19-6227

Data Areas and Control Block Logic—CP, LY20-0896

General Information Manual, GC19-6221

Release 4.2 Guide, SC23-0189

Installation Guide, SC38-0107

Library Guide, Glossary, and Master Index, GC23-0187

OLTSEP and Error Recording Guide, SC19-6230

Operator's Guide, SC19-6225

Quick Reference, SX22-0005

Service Routines and Program Logic, LY20-0898

System Definition Files, SC23-0186

System Logic and Problem Determination Guide—CP, LY20-0897

System Messages and Codes, SC19-6226

System Messages Cross-Reference, SC23-0190

System Programmer's Guide, SC19-6224

IBM Virtual Machine/System Product

Data Areas and Control Block Logic—CMS, LT00-1606

Group Control System Guide, ST24-5249

Interactive Problem Control System Guide, ST24-5260

System Logic and Problem Determination Guide—CMS, LT00-1604

Terminal Reference, GT00-1581

Virtual Machine Running Guest Operating Systems, GT00-1586

IBM 3270 Information Display System

IBM 3270 Information Display System Library User's Guide, GA23-0058

3081 Processor Complex

Input/Output Configuration Program User's Guide and Reference, GC28-1027

3090 Processor Complex

Input/Output Configuration Program User's Guide and Reference, SC38-0038

3850 Mass Storage System

IBM 3850 Mass Storage System (MSS) Installation Planning and Table Create, GC35-0028

3704/3705/3725 Communications Controllers

IBM 3704 and 3705 Control Program Generation and Utilities Guide and Reference Manual (OS/VS TCAM Levels 5 and 6 in VS1; VS2 rel 1.6, 1.7, 2) SCP 5744-BA1, GC30-3008

IBM 3704 and 3705 Control Program Generation and Utilities Guide and Reference Manual (TCAM 10 SVS - 5742-017) SCP 5742, 5744-AN1/BA2, 5747-AG1-AG2, GC30-3008

Guide to Using the IBM 3704 Communications Controller Control Panel, GA27-3086

Guide to Using the IBM 3705 Control Panel, GA27-3087

DASD Support

Device Support Facilities User's Guide and Reference, GC35-0033

Notes:

1. *See also the* Functional Characteristics *document for your processor.*
2. *References in the text to titles of VM/SP HPO publications are in abbreviated form.*

The VM/SP HPO Library

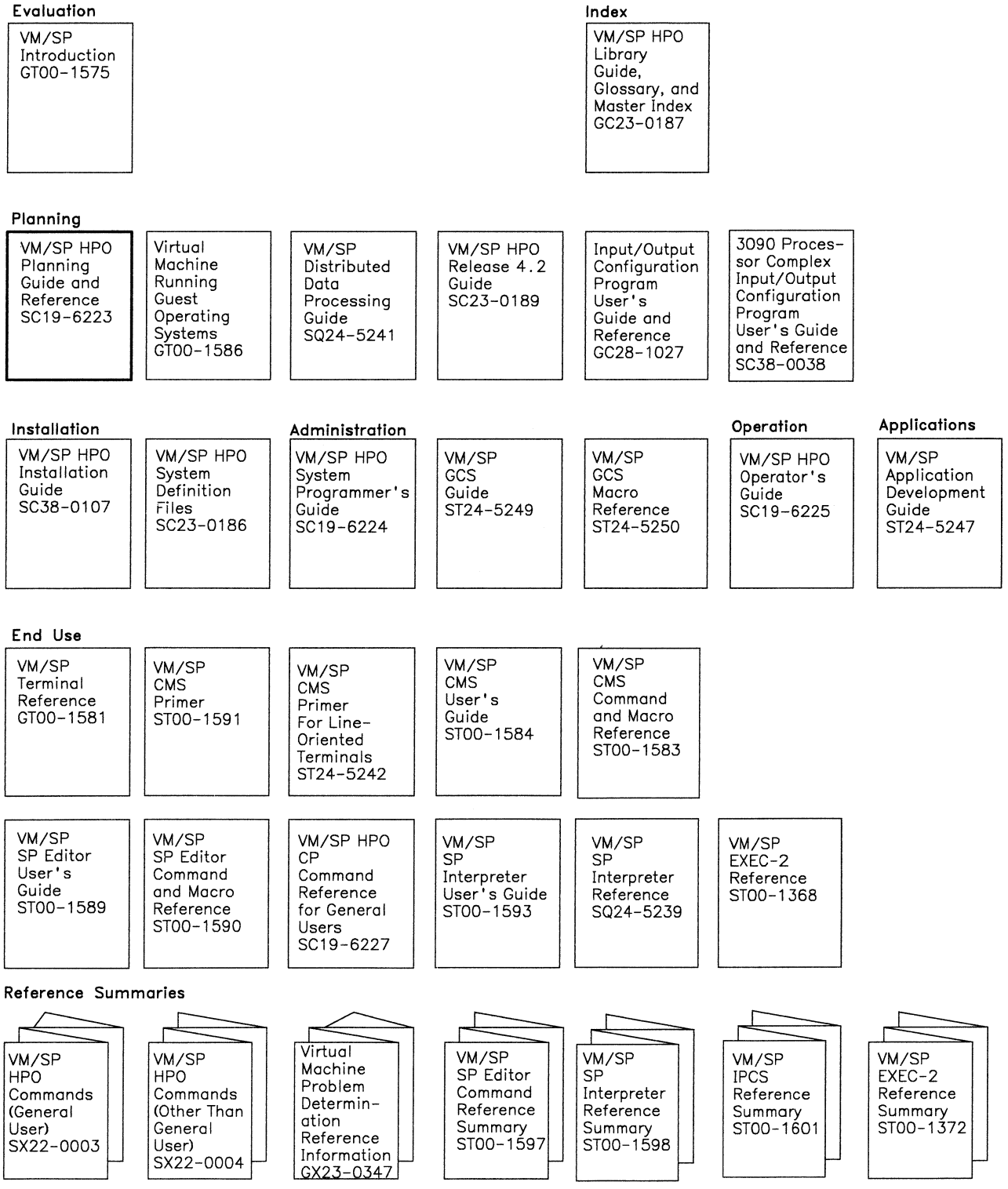


Figure 1 (Part 1 of 2). Virtual Machine/System Product HPO Library

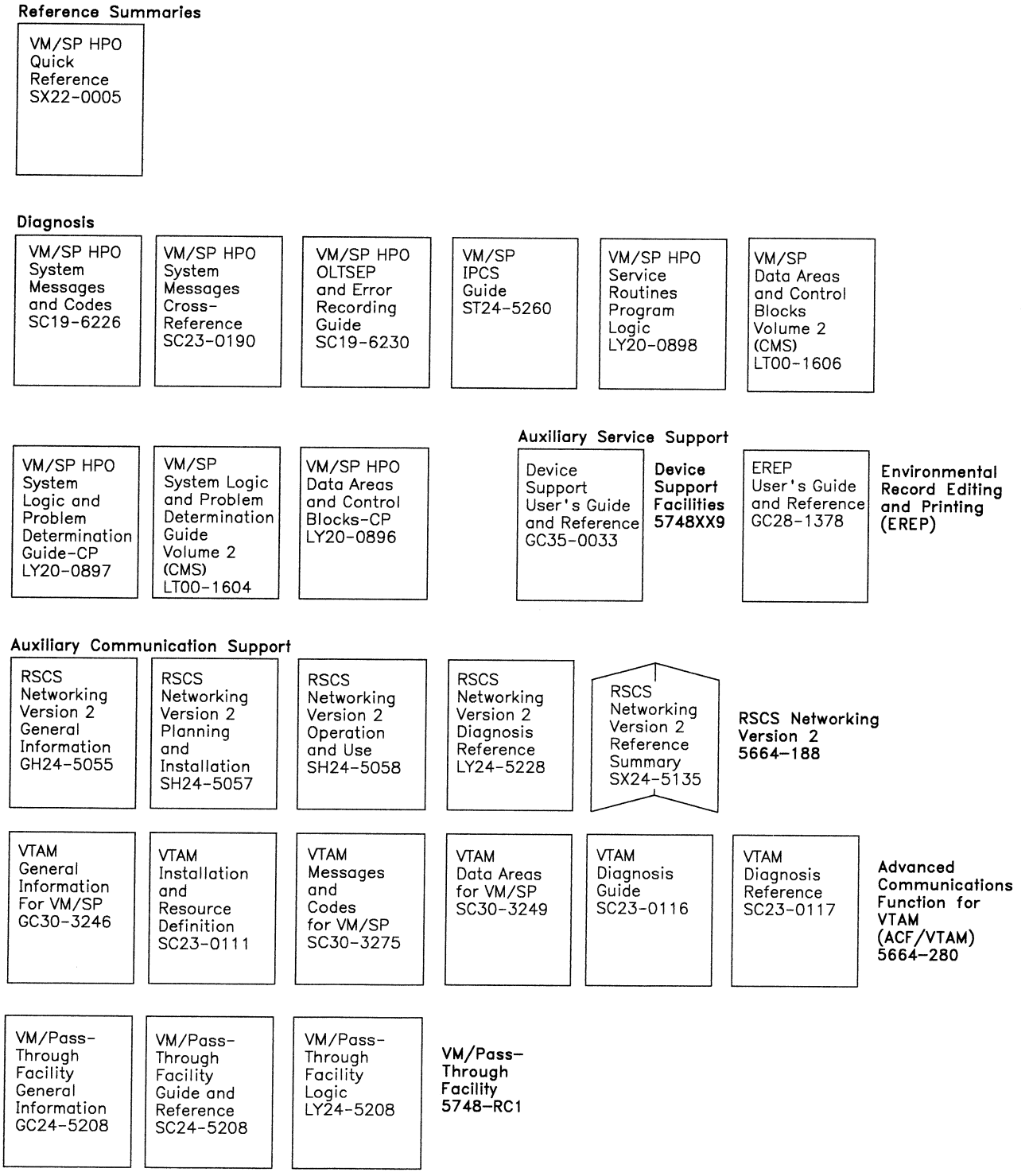


Figure 1 (Part 2 of 2). Virtual Machine/System Product HPO Library

Contents

Part 1. Planning for System Generation 1

Chapter 1. Introduction 3
Virtual Machine Operating Systems 3
Introduction to System Generation 4

Chapter 2. Configurations 7
VM/SP HPO Minimum Configuration 7
Configurations Supported by CMS 8
Configurations Supported by a Remote Spooling Communications Subsystem (RSCS) 9
Devices Supported by VM/SP HPO 9
- Processors 10
- Direct Access Storage Devices 11
- Magnetic Tapes 13
- Unit Record Devices 14
- Terminals and Display Stations 14
- Transmission Control Units 21
- Remote Spooling Communications Subsystem (RSCS) 24

Other Considerations for Planning Your Configuration 26
- Two-Channel Switch 26
- Multisystem Communication Unit 26
- Unsupported Devices 26
- Service Record File 27
- Processor Controller 28
- Vector Facility 28

Chapter 3. Estimating CP Storage Requirements 29
Real Storage Requirements for CP 29
Reducing the CP Nucleus Size 31
Direct Access Storage Requirements for CP 33
- DASD Characteristics 33
- CP Nucleus DASD Requirements 36
- Error Recording DASD Requirements 37
- Warm Start Data DASD Requirements 37
- Checkpoint Start Data DASD Requirements 37
- Dump Space DASD Requirements 37
- VM/SP HPO Directory DASD Requirements 37
- Saved System DASD Requirements 38
- Paging and Spooling DASD Requirements 39

VSAM and Access Method Services Requirements 39
Vector Facility Requirements 40
IPCS Considerations 40
- External Storage 40
- Real Storage 41

Estimating DASD Storage Requirements for CMS Minidisks 41

Chapter 4. Planning for CMS 43
CMS Storage Requirements 43
Virtual Storage 43
Auxiliary Storage 44
Device Support 44
CMS Libraries 45
System Macro Libraries 46
System Text Libraries 47
Other Libraries 47
CMS Command Language 47
CMS Program Language Facilities 48
Limited Support of OS and VSE in CMS 49
DL/I in the CMS/DOS Environment 50
CMS Disk and File Management 51
Disk Access 51
File Sharing 52
CMS Disk File Format 52
Identifying Disk Files 54
CMS Tape Support 54
CMS Unit Record Support 55
Saving CMS 55

Chapter 5. Minidisks 57
Defining Minidisks 57
Minidisk Space Allocation 58
OS Minidisks 60
VSE Minidisks 60
MSS Minidisks 60
Minidisk Initialization 61
Alternate Tracks/Blocks 61
3330/3350 DASD 61
3340/3344 DASD 62
2314/2319 DASD 63
3375/3380 Disks 64
FB-512 DASD 64
Labels 65
Sharing Minidisks 65

Chapter 6. Planning and Requirements for IPCS 67
Machine Requirements 67
Programming Requirements 67
Conversion Considerations 67
Storage Requirements 68
External Storage 68
Calculating DASD Space 69

Chapter 7. Planning for CMS VSAM and Access Method Services 71
Hardware Devices Supported by VSE 71
Programming Languages Supported 72
Data Set Compatibility Considerations 72
ISAM Interface Program (IIP) 73
Planning Considerations for Installing VSAM Under CMS 73

Chapter 8. Planning for CMS/DOS 75
VSE System Generation Considerations 75
When the VSE System Must Be Online 76
CMS/DOS Tape Handling 77
Label Information Area 77

Chapter 9. Planning for Virtual Machine Operating Systems (Other Than CMS) 79
Operating Systems that Execute in Virtual Machines 79
2K Key Operating Systems 79
4K Key Operating Systems 80
The VM/VS Handshaking Feature 80
Multiple Virtual Machines Using the Same Operating System 82
Remote Spooling Communications Subsystem 82
The RSCS Virtual Machine 82
RSCS Planning Considerations 83
The IX/370 Handshaking Feature 84

Chapter 10. Planning for I/O 85
Channel Switching 85
Alternate Path Support 88
Restrictions 89
3880 Storage Subsystem 90
Configuration 90
The Cache 91
Addressing 93
Error Conditions 95
Channel-Set Switching Facility 95
Monitoring and Service Support Facility 96
Input/Output Configuration Program 97
Stand-Alone Input/Output Configuration Program 98
VM Input/Output Configuration Program 98
MVS Input/Output Configuration Program 99
Using IOCP in Single Processor Mode 99
References 99
Operating Systems Using Reserve/Release 100
Hardware-Supported Reserve/Release 101
Virtual Reserve/Release 103
Control Program Handling of a Reserve CCW 103
Restrictions 103
Missing Interruption Handler 105
Logical Device Support Facility 106
Inter-User Communication Vehicle 107
Virtual Machine Communication Facility 108
Summary of VMCF Functions 109
Extended Channel Support 110

Chapter 11. Improving System Performance 111
Performance Measurement and Analysis 112
Using Performance Options 113

Specifying a Virtual=Real Machine 113
Real Storage Validation 115
V=R Virtual Machine Storage Requirements 117
Specifying the Virtual=Real Storage Size 117
Virtual Machine Assist 121
Extended Storage Support 122
VM/370 Extended Control-Program Support 122
Shadow Table Bypass Assist 123
MVS/System Extensions Support 124
MVS/System Product as a Virtual Machine 124
MVS/System Product Support 125
Improving the Performance of an MVS/SP Virtual Machine 125
Shared Segment Protection 132
Enhanced Paging Subsystem 132
Paging Storage 133
Enhanced Performance for Dyadic and Dual Processors 133
Other Desirable Features 133

Chapter 12. Attached Processors and Multiprocessor Systems 135
Attached Processors 135
Modules Containing AP Support 135
Multiprocessors 136
Modules Containing MP Support 136

Chapter 13. Planning for 3270 Information Display System Terminals 137
Remote Terminal Attachments 137
3270 Support on Binary Synchronous Lines 138
Remote Configurations Supported 139
System Generation Requirements for Remotely-Attached Terminals 140
The CLUSTER Macro 141
The TERMINAL Macro 141
The RDEVICE Macro 142
The Resource Identification Codes 142
An Example of a Remote 3270 Configuration 143
Local Attachments 145
Printers 146
System Generation Requirements for Locally-Supported Display Systems 146

Chapter 14. Planning for the 3704, 3705, and 3725 Communications Controllers 147
CP Support of the 3704, 3705, and 3725 Communications Controllers 147
3704 and 3705 Support Package 148
Emulation Program (EP) with VM/SP HPO 148
Planning Considerations 149
Creating an Entry in the System Name Table 150
Reserving DASD space for the 3704/3705 Control Program Image 150
Related Publications 150

Chapter 15. Planning for SNA Console Communications Services 153
Structure of the SNA Environment 153
NCP and PEP Sharing 153

Tracing for SNA Console Communications Services 154
Normal Trace 154
Error Trace 154
Excluding SNA CCS Modules 155

Chapter 16. Planning for the 3800 Image Library 157
Creating and Updating a 3800 Named System 157
Hardware Features Supported 158
Related Publications 158

Chapter 17. Planning for the 3850 Mass Storage System 159
Hardware Supported 159
Defining the MSS Communication Device 162
The Mass Storage Control Tables 163
Example 164
Creating MSS Volumes 164
Copying 3330-1 Volumes to 3330V Volumes 164
Using 3330V Volumes for VS System Residence 165

Part 2. Defining Your VM/SP HPO System 167

Chapter 18. Introduction to System Definition 169
System Definition Files 169
Required Files 169
Optional Files 169
System Integrity 170
MVS Guest Machine Environment 171
Customer Responsibilities 171

Chapter 19. Creating Your Directory 173
Considerations for Preparing the Directory Control Statements 174
General Considerations 174
Using Directory Profiles 174
Security Considerations 175
Virtual Machines You Must Define 177
Sample Directory Entries 178
The System Operator's Virtual Machine (OPERATOR) 179
A Virtual Machine to Receive System Dumps (OPERATNS) 179
Other System Virtual Machines 180
The Directory Supplied with the Starter System 182
Allocating DASD Space for the Directory 183
The Directory Program 184
Invoking the Directory Program (DMKDIR) 186
Directory Control Statements 188
Directory Entries for CMS/DOS 230

Chapter 20. Defining Your Own User Classes 231
Planning Your New Class Structure 232
Changing the Class Structure 234
The Override Program 235
OVERRIDE Command 235
The Class-Override File 236
Directory Control Statements for Defining User Classes 239

Chapter 21. Preparing the Real I/O Configuration File (DMKRIO) 241
Coding the Real I/O Configuration Macros for Remote 3270s 243
CLUSTER Macro 245
Examples 246
TERMINAL Macro 247
Examples 251
RDEVICE Macro 253
Examples 264
Unit Record Error Messages 265
Control Unit Error Messages 265
Coding the RDEVICE Macro for Remote 3270 Information Display Devices 266
Coding the RDEVICE Macro for the 3704/3705/3725 Communications Controller 268
Coding the RDEVICE Macro for the 3800 Printing Subsystem 273
Coding the RDEVICE Macro for the 3880 Storage Subsystem 274
RCTLUNIT Macro 277
Examples 281
Channel Error Message 281
RCHANNEL Macro 282
Examples 283
RIOGEN Macro 284
Examples 285
Example of Coding the Real I/O Configuration File (DMKRIO) 286
Considerations for Coding the Input/Output Configuration Source File 287
Example of Coding an SNA Controller 293

Chapter 22. Preparing the System Name Table File (DMKSNT) 295
Coding the NAMESYS Macro 296
Available Space for Each DASD Type 299
Relationship of Page Numbers, Segment Numbers, and Hexadecimal Addresses 300
More Information 300
Coding the NAMENCP Macro 301
Coding the NAME3800 Macro 303

Chapter 23. Preparing the CP System Control File (DMKSYS) 305
Performance Considerations for Coding the DMKSYS File Macros 311
SYSOWN Macro 313
Example 313
SYSPAG Macro 315
General Coding Considerations 318
Format/Allocate (DMKFMT) Considerations 319
Page Migration 320
Establishing the Access Structure 322
Error Messages Generated by the SYSPAG Macro 326
SYSRES Macro 329
Example 335
SYSMON Macro 337
Example 339
SYSJRL Macro 340

SYSCOR Macro 343
Example 346
SYSOPR Macro 347
Example 347
SYSACNT Macro 348
SYSTIME Macro 350
Examples 351
SYSFORM Macro 352
Examples 353
SYSPCLAS Macro 354
Examples 355
SYSID Macro 356
Examples 356
SYSMIH Macro 358
Example 360
SYSFCN Macro 361
SYSLOCS Macro 363

Chapter 24. Additional System Definition Files 365
Altering the Forms Control Buffer Load (DMKFCB) 365
Altering the Universal Character Set or Font Offset Buffer 366

Chapter 25. Defining Saved Systems and Discontiguous Saved Segments 367
Saved Systems 367
Discontiguous Saved Segments 367
Installing the Newly Defined DCSS or Saved System 368
Considerations 369
Shared Segments 369
SNTMAP EXEC 370
Introduction 370
Input to SNTMAP 370
Output from SNTMAP 371
How to Use SNTMAP 371
Security Considerations 373

Appendixes 375

Appendix A. Program Products, Installed User Programs, Field-Developed Programs, and Emulators 377
VM/370 Assembler 377
Program Products 377
Integrated Emulators 381

Appendix B. Configuration Aid 383
Matching Hardware to Control Units 383
Matching Hardware to Shared or Nonshared Virtual Control Units 388
Temporary Disk Space (TDISK) 389
Unit Record Devices 390
Terminals and Displays 391
Miscellaneous Virtual Devices 392

Appendix C. Compatible Devices 393
IBM 6640 Document Printer 393
IBM Office System 6 Information Processors (6/650, 6/440, 6/430) 393
IBM Mag Card II Typewriter and IBM 6240 Mag Card Typewriter 393

Appendix D. VM/SP HPO Restrictions 395
Channel Program Restrictions 395
Dynamically Modified Channel Program Restrictions 395
Minidisk Restrictions 396
Timing Dependencies 400
Processor Model-Dependent Functions 401
Channel Model-Dependent Functions 402
Virtual Machine Characteristics 403
3340/3344 Direct Access Storage Device Restrictions 408
Mass Storage System Restrictions 409
3880 Storage Subsystem Restrictions 410
Conversational Monitor System Restrictions 411
Virtual Device Restrictions 414
Extended Channel Support Restrictions 415
Preferred Machine Assist Restrictions 416
Miscellaneous Restrictions 418
  Restrictions on VM in General 418
  Restrictions Relating to I/O 420
  Restrictions on Specific Devices 421
  Restrictions on Commands or Features 423

Appendix E. Worksheet to Aid in Coding the NAMESYS Macro 427

Appendix F. Sample SNTMAP Output 431
DASD SNTMAP 431
MEMORY SNTMAP 432

Index 435

Figures

1. Virtual Machine/System Product HPO Library xii
2. Where to Find Information 6
3. Real Storage Requirements for CP 29
4. Load Lists and System Configurations 31
5. Reducing the Size of the CP Nucleus 32
6. Sample Layout of a 3310 34
7. FBA Blocks and Pages per Access Position 35
8. Converting Block and Page Numbers to Pseudo-Cylinder Numbers 35
9. DASD Space Requirements by DASD Type 36
10. Information Pages Required for Saved Systems 38
11. External Storage Requirements for IPCS 40
12. Devices Supported by a CMS Virtual Machine 45
13. Volume Label Contents for CMS Formatted Disks 52
14. CMS Disk File Statistics for 800-Byte CMS Blocks 53
15. Use and Definition of Minidisks 59
16. Approximate Space Usage for Fixed Area and One Dump by Device Type 69
17. Cylinders Required to Contain a Dump Depending on Virtual Storage Size 70
18. A Remote Spooling Communications Subsystem 83
19. Channel Switching between Two Processors 86
20. Channel Switching on One Processor 86
21. Real I/O Control Block Structure for Alternate Control Unit Specification 88
22. Real I/O Control Block Structure for Alternate Channel Specification 89
23. 3880 Storage Subsystem Dual-Frame Configuration 91
24. Examples of the 3880 Model 11 and Model 21 Storage Subsystem 94
25. Summary of Reserve/Release Support 104
26. Logical Device Support Facility Subfunctions 107
27. Virtual Machine Communication Facility (VMCF) Functions 109
28. Attached Processor Systems 135
29. Multiprocessor Systems 136
30. Example of Determining Line Code for Remote 3270 Resource Identification Codes 143
31. Sample List of Resource Identification Codes for Operations 145
32. Sample Worksheet for Planning a New Class Structure 233
33. Remote 3270 Control Unit and Device Addressing 249
34. Examples of Remote 3270 Addressing 252
35. IBM 3704/3705 Models 271
36. Example of a Real Configuration 288
37. Real I/O Configuration File 289
38. Input/Output Configuration Program Source File 291
39. How Pages and Swap Sets Are Migrated Down the Paging Hierarchy 320
40. I/O Distribution across Good Paging Hierarchies 321
41. I/O Distribution across a Bad Paging Hierarchy 321
42. SYSPLIST/ALOCBLOK Generation and Relationship 323
43. Sample SYSPLIST/ALOCBLOK Chaining Structure 324

44. Dynamic Checkpoint Start Area Calculations 334
45. Warm Start Area Calculations 336
46. Integrated Emulators that Execute under VM/SP HPO 382

Part 1. Planning for System Generation

Part 1 contains planning information. It describes the various components, options, and features of VM/SP HPO and tells you what you must do to install them. Part 1 contains the following sections:

- Introduction to Planning
- Configurations
- Estimating Storage Requirements
- Planning for CMS
- Minidisks
- Planning and Requirements for IPCS
- Planning for CMS VSAM and Access Method Services
- Planning for CMS/DOS
- Planning for Virtual Machine Operating Systems (Other than CMS)
- Planning for Input/Output
- Improving System Performance
- Multiprocessing Systems
- Planning for 3270s
- Planning for the 3704, 3705, and 3725 Communications Controllers
- Planning for SNA Console Communications Services
- Planning for a System that Supports the 3800 Image Library
- Planning for a 3850 Mass Storage System

Chapter 1. Introduction

The IBM VM/System Product High Performance Option (VM/SP HPO) is a program product that manages a real computing system so that all its resources—processor(s), storage, and input/output devices—are available to many users at the same time. Each user has at his disposal the functional equivalent of a real computing system. Because this functional equivalent is simulated by VM/SP HPO, it is called a "virtual" machine.

The Control Program (CP) manages the real resources of a computing system to provide multiple virtual machines. VM/SP HPO adds new functions to CP. VM/SP HPO adds no new functions to the Conversational Monitor System (CMS), a component of VM/System Product. CMS provides a wide range of conversational and time-sharing facilities. Using CMS, you can create and manage files, and compile, test, and execute problem programs.

You can appreciably expand the capabilities of a VM/SP HPO system by installing the RSCS Networking program product (5748-XP1) and the VM/IPCS Extension program product (5748-SA1). These program products provide extended features to the following VM/370 components:

- The Remote Spooling Communications Subsystem (RSCS), which transfers spool files between VM/SP HPO users and remote locations over telecommunication lines.

- The Interactive Problem Control System (IPCS), which provides VM/SP HPO problem analysis and management facilities, including problem report creation, problem tracking, and CP abend dump analysis.

The processors that VM/SP HPO supports are listed under the heading "Processors" on page 10. The real System/370 must have the Dynamic Address Translation feature, a hardware facility that translates virtual storage addresses to real storage addresses, and the System Timing facility. Also, it must operate in extended control mode, a mode in which all the features of a System/370, including dynamic address translation, are operational.

For an overview of the functions performed by VM/SP HPO, see the *VM/SP HPO Introduction.*

Virtual Machine Operating Systems

While the control program of VM/SP HPO manages the concurrent execution of the virtual machines, it is also necessary to have an operating system managing the work flow within each virtual machine. Because each virtual machine executes independently of other virtual machines, each one may use a different operating system, or different releases of the same operating system.

The operating systems that can run in virtual machines are listed in Chapter 9, "Planning for Virtual Machine Operating Systems (Other Than CMS)," later in this book.

The operating systems themselves execute as though they were controlling real devices and real storage, but they must not violate any of the restrictions listed in Appendix D, "VM/SP HPO Restrictions."

Introduction to System Generation

The purpose of the system generation is to create a system that meets your installation's particular needs.

If you do not have an operational system the first step in the system generation procedure is to restore the starter system. The starter system is a small working copy of a basic system. Using the starter system, you tailor a VM/SP HPO system to your own hardware configuration. You also describe your DASD volumes and define how they are to be used.

The following versions of the starter system can be ordered:

- 3330 starter system
- 3350 starter system
- 3375 starter system
- 3380 starter system

All starter systems must be restored to a compatible disk (for example, a 3330 starter system to a 3330 disk), but all starter systems can then be used to build any supported system residence volume type.

If you have an operational system, the starter system is not required. The *VM/SP HPO Installation Guide*, SC38-0107, describes the system generation procedures for generating a VM/SP HPO system from the starter system or a current program product system.

Before you begin the system generation procedure, you should:

- Know which devices to include in your system.
- Create the real I/O configuration (DMKRIO) file describing your I/O configuration. If an IBM Mass Storage System is to be attached to the system, you must coordinate the real I/O configuration with the Mass Storage Control's tables.
- Decide how many virtual machines to define.
- Create the directory control statement file describing the virtual machines.

- Decide which volumes are to be owned and used by CP (for system residence, paging, spooling, and so on), the amount of real storage available to VM/SP HPO, and the user identification of the real system operator.

- Create the CP system control (DMKSYS) file describing CP-owned volumes, the real storage size, and so on.

- Create the system name table (DMKSNT) describing the name and location of saved systems.

- If you wish, you can create your own forms control buffer (module DMKFCB). This module is supplied with the starter system.

Once you have defined your system with these files, you can begin the system generation procedure. You should read the rest of Part 1 to be sure you have all the information you need to generate your system. Figure 2 shows you where to find planning information in Part 1. Part 2 has the information you need to code the macro statements that define your system. Before you start the system generation procedure be sure you have the following manuals available:

VM/SP HPO Release 4.2 Guide
VM/SP CMS Command and Macro Reference
VM/SP CMS User's Guide
VM/SP HPO CP Command Reference for General Users
VM/SP HPO Operator's Guide
VM/SP HPO System Messages and Codes
VM/SP HPO System Programmer's Guide

For terms and definitions used in this publications see:

VM/SP HPO Library Guide, Glossary, and Master Index

If you are using the MSS support, you will also need the following:

OS/VS Message Library: Mass Storage System Messages

OS/VS Mass Storage System (MSS) Installation Planning and Table Create

OS/VS1 System Generation Reference

or

OS/VS2 System Programming Library: System Generation Reference

You will also need the *Functional Characteristics* document for your processor.

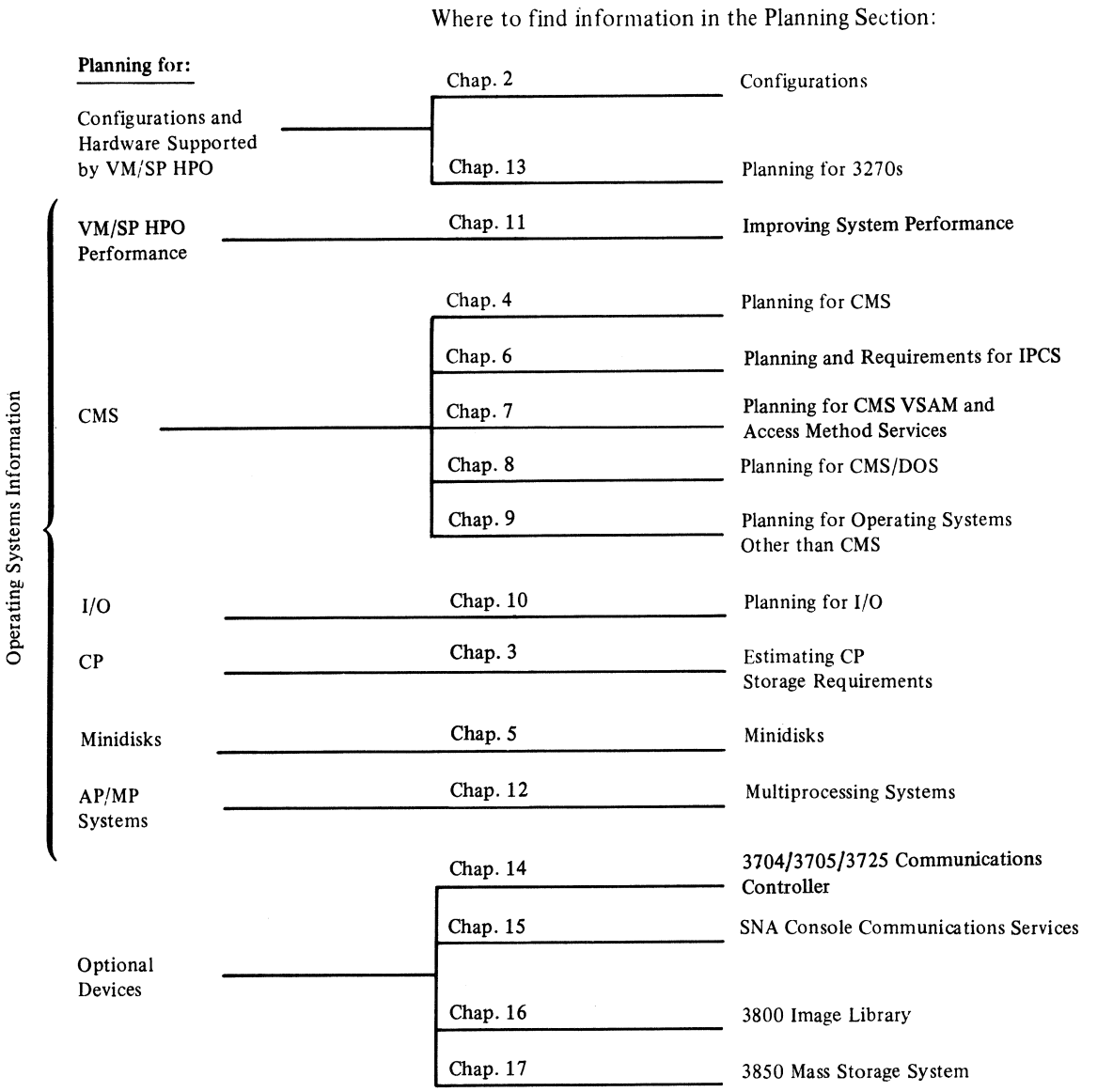


Figure 2. Where to Find Information

During the system generation procedure, you apply the Program Update Tape (PUT) supplied with the starter system, then the VM/SP HPO Product Tape and any PUT that applies to VM/SP HPO. This updates your system to the current level. Then use the Installation Verification Procedure (IVP) to verify that the VM/SP HPO system is functioning properly.

Note: If you are installing a system Program Update tape, use this publication in conjunction with the Memo to Users and PUT Document contained on the PUT. After you generate CP and CMS, you can generate the 3704/3705 Control Program.

Chapter 2. Configurations

Before you begin the system generation procedure, make sure your installation has the minimum configuration supported by VM/SP HPO and the features and facilities required by VM/SP HPO.

VM/SP HPO Minimum Configuration

The minimum configuration supported by VM/SP HPO is:

One	Processor (1 megabyte of storage)
One	System console device
One	Printer
One	Card reader[1]
One	Card punch[1]
Two	Spindles of direct access storage
One	Nine-track magnetic tape unit
One	Multiplexer channel
One	Selector or Block multiplexer channel

To determine the amount of real storage and direct access storage necessary for a configuration, see Chapter 3, "Estimating CP Storage Requirements."

A representative configuration is:

IBM 3081 Processor Complex Model D24:

- 3081 Processor Unit
- 3082 Processor Controller
- 3087 Coolant Distribution Unit (CDU)
- 3278-2A System Console
- 3089 Power Unit

IBM 3279 Color Display Station (the operator console)

IBM 3287 Printer

IBM 3350 Direct Access Storage Model A2 or 3370 Direct Access Storage—Four drives attached to a 3880 Storage Control Model 1

IBM 3350 Direct Access Storage Model A2F—Fixed-head feature

IBM 3420 Magnetic Tape Units—Two

IBM 3800 Printing Subsystem

IBM 3705 Communications Controller

---

1 This device is not needed for a cardless system.

IBM 3278 Display Stations (as needed) with the 3274 Control Unit (local attachment or remote attachment).

IBM 3279 Color Display Stations (as needed) with the 3274 Control Unit (local or remote attachment).

Configurations Supported by CMS

CMS supports:

- Virtual storage size: minimum of 320K bytes, up to 16 million bytes in multiples of 4K.
- Virtual console: any terminal supported by VM/SP HPO as a virtual machine operator's console.
- The same unit record devices (card readers, punches, and printers) supported by VM/SP HPO as spooling devices, except the 2520 Punch. See "Unit Record Devices" on page 14.
- Up to twenty-six logical 2314, 2319, 3340, 3330 Model 1, 2, or 11, 3333 Model 1 or 11, 3350, 3375, 3380, or FB-512 direct access storage devices. The maximum size of a CMS minidisk is:

Device Type	Model(s)	CMS/VSAM	CMS 800-byte Format	CMS 1K, 2K, or 4K Format
2314/2319	-	200 cyls.	203 cyls.	203 cyls.
3310	-	126,016 blocks	not supported	126,016 blocks
3330	1 or 2	404 cyls.	246 cyls.	404 cyls.
3330	11	808 cyls.	246 cyls.	808 cyls.
3333	1	404 cyls.	246 cyls.	404 cyls.
3333	11	808 cyls.	246 cyls.	808 cyls.
3340	35	348 cyls.	348 cyls.	348 cyls.
3340	70	696 cyls.	682 cyls.	696 cyls.
3350	native mode	555 cyls.	115 cyls.	555 cyls.
3370	A1 or B1	558,000 blocks	not supported	558,000 blocks
3370	A2 or B2	712,752 blocks	not supported	712,752 blocks
3375	-	959 cyls.	182 cyls.	959 cyls.
3380	-	855 cyls.	121 cyls.	885 cyls.
3380	AE4 or BE4	1770 cyls.	121 cyls.	1770 cyls.

- Up to four 2400, 2415, 2420, 3410 (9 track only), 3420, or 3480 Magnetic Tape Units.

Configurations Supported by a Remote Spooling Communications Subsystem (RSCS)

RSCS supports:

- Virtual storage size: minimum of 512K, up to 16 million bytes in multiples of 4K.
- Virtual console: any terminal supported by VM/SP HPO as a virtual machine operator's console.
- Any virtual card readers, punches, and printers supported by VM/SP HPO as spooling devices.
- One logical 2314, 2319, 3330 Model 1, 2, or 11, 3333 Model 1 or 11, 3340, or 3350, direct access storage devices (800-byte format only).
- Transmission Control Units: 2701 Data Adapter Unit; 2703 Transmission Control Unit; or 3704, 3705, or 3725 Communications Controllers in EP mode only. Only binary synchronous communication transmission is supported.

The minimum configuration supported by RSCS is:

- 512K virtual storage
- One console
- One reader
- One transmission control unit
- One or more binary synchronous lines dedicated to the RSCS virtual machine.

Devices Supported by VM/SP HPO

The following devices are supported by VM/SP HPO except as otherwise noted. The devices are listed by device type:

- Processors
- Direct access storage devices
- Magnetic tapes
- Unit record devices (printers, readers, and punches)
- Terminals
- Transmission control units and communications controllers
- Remote spooling devices
- Other devices.

This section does not include SNA supported devices. For information concerning supported SNA devices, refer to the VM/VTAM Communications Network Application (VM/VCNA) publications listed in the Preface.

Processors

Processors Supported by VM/SP HPO

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

Processors	Configuration
System/370 Model 155-II	UP
System/370 Model 158	UP/AP/MP
System/370 Model 158-3	UP
System/370 Model 165-II	UP
System/370 Model 168	UP/AP/MP
System/370 Model 168-3	UP
3031	UP/AP
3032	UP
3033	UP/AP/MP
3042-2	AP
3081 Processor Complex	Dyadic **(see Note 1)**
3083 Processor Unit	UP
3084 Processor Complex	Partitioned Processing **(see Note 2)**
3090 Processor Complex Model 200	Dyadic **(see Note 1)**
4341	UP
4381 Model 1	UP
4381 Model 2	UP
4381 Model 3	Dual **(see Note 1)**

Notes:

1. *In a* **dyadic** *processor, like the 308x Processor Complex, the 3090 Processor Complex Model 200, or a* **dual** *processor, like the 4381 Model 3, two processing units share central storage. Each instruction processor has access to its own assigned channel set of up to 16 channels in a standard system, or up to 32 channels with the extended channel support option (available only for the 3090 Processor Complex). The entire complex may have up to 24 channels, or 48 channels with extended channel support. Channel switching is available for dyadic processors but not for dual processors.*

   A dyadic or dual processor cannot be configured into two independent uniprocessors using separate control programs.

2. *The 3084 is a four-way multiprocessor that can be reconfigured into two dyadic processors, each capable of executing a different operating system. In this configuration, the 3084 operates in* **partitioned processing mode**. *VM/SP High Performance Option does not support the 3084 operating as a single-image configuration four-way processor.*

Processor-Required Features and Facilities

Processor features and facilities required by VM/SP HPO are listed below. Only features and facilities that are not standard on a particular processor are described:

- The Channel Indirect Data Addressing feature on each of the 2860, 2870, and 2880 stand-alone I/O channels on the Model 165 II or 168:
  - 2860, features #1861, 1862, and 1863
  - 2870, feature #1861
  - 2880, features #1861 and 1862.

Note: The stand-alone channels that attach to the System/370 Models 165 II and 168 require that the Channel Indirect Data Addressing feature be ordered as a separate feature for proper operation of the input/output channels in a Dynamic Address Translation environment.

Direct Access Storage Devices

Direct Access Storage Devices Supported by VM/SP HPO

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

- IBM 2305 Fixed Head Storage, Model 2
- IBM 2314 Direct Access Storage Facility (not supported for CP system residence)
- IBM 2319 Disk Storage
- IBM 3330 Disk Storage, Models 1, 2, and 11
- IBM 3333 Disk Storage and Control, Models 1 and 11 for 3330 Models 1, 2, and 11
- IBM 3340 Direct Access Storage Facility, Models A2, B1, and B2; the 3348 Data Modules, Models 35, 70, and 70F; and the 3344 Direct Access Storage, Model B2
- IBM 3350 Direct Access Storage, Models A2 and B2
- IBM 3310 Direct Access Storage
- IBM 3370 Direct Access Storage, Models A1, A2, B1, and B2
- IBM 3375 Direct Access Storage
- IBM 3380 Direct Access Storage, Models A4, AA4, AD4, AE4, B4, and BE4
- IBM 3880 Model 11, 21, 13 and, 23 Storage Subsystem.

All of these direct access devices are supported as dedicated devices. All except the 2305 are supported by CMS. All except the 3880 Storage Subsystem are supported as paging, swapping, and spooling devices and as virtual devices for use by virtual machines. The 3330, 3350, 3375, 3380, and 3880 Model 13 or 23 Storage Subsystem are also supported as system residence devices. The 3880 Model 11 or 21 Storage Subsystem is supported as a paging or swapping device.

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

- IBM 2835 Storage Control
- IBM 2844 Auxiliary Storage Control
- IBM 3830 Storage Control Models 1, 2, 3
- IBM 3880 Storage Control Models 1, 2, 3
- IBM Integrated Storage Control on the System/370 Model 158 for:
  - 3330 Models 1 and 2
  - 3333 Models 1 and 11
  - 3340 Model A2 and 3344 Model B2
  - 3350 Model A2.
- IBM Integrated Storage Control on the System/370 Model 168:
  - 3330 Models 1 and 2
  - 3333 Models 1 and 11
  - 3340 Model A2 and 3344 Model B2
  - 3350 Model A2.

Desirable Features

The 3880 Storage Control Unit Buffer Features are:

- Feature #6550 for 3380 DASDs attached to 3880 Models 2 and 3

  The speed matching buffer feature (Feature #6550) for the 3380 supports the use of extended count-key-data channel programs.

  If the 3380 attached to the 3880 Controller Model 3 with Speed Matching Buffer (Feature #6550) is part of your installation, you can run extended count-key-data channel programs.

  Note: The speed matching buffer is not supported for 3380 Models AD4/BD4 or AE4/BE4.

- Feature #6560 for 3375 DASDs attached to 3880 Models 1 and 2

These features modify the direct access data transfer path by adding a buffer between the DASD device and the multiplexer channel. When the 3880 control unit is equipped with the respective buffer feature, 3380 or 3375 devices can be attached to channels that operate at data rates slower than that of the DASD device.

The respective Buffer Features allow attachment of 3380 or 3375 devices to the following types of channels:

- 1.5 megabyte block multiplexer channels on S/370 Models 155-II, 158, 165-II, 168, 168-3, 303x, and 3042 Model 2
- 2.0 megabyte block multiplexer channels on the 4341 Processor and high speed block multiplexer channels on the 4331 Model Group 2 Processor

The speed-matching operation for writing records to the 3375 DASD over a 1.5 megabyte block multiplexer channel requires that the data transfer across the channel and into the buffer begin in advance of the data transfer from the buffer to the 3375. To accomplish this with minimum loss of disk revolutions, special channel commands provide write-prenotification whenever possible.

The speed matching buffer is bypassed when attached to a 3 megabyte data streaming channel.

Note: The 3880 speed matching buffer is not available for the 3880 Storage Subsystem.

Magnetic Tapes

The magnetic tape devices supported by VM/SP HPO are:

- IBM 2401, 2402, and 2403 Magnetic Tape Units, Models 1, 2, 3, 4, and 5
- IBM 2415 Magnetic Tape Units, Models 1, 2, 3, 4, 5, and 6
- IBM 2420 Magnetic Tape Units, Models 5 and 7
- IBM 3410/3411 Magnetic Tape Unit, Models 1, 2, and 3
- IBM 3411 Magnetic Tape Unit and Control, Models 1, 2, and 3
- IBM 3420 Magnetic Tape Units, Models 3, 4, 5, 6, 7, and 8
- IBM 3430 Magnetic Tape Unit and Control
- IBM 3480 Magnetic Tape Unit

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

- IBM 2803 Tape Control
- IBM 2804 Tape Control
- IBM 3411 Magnetic Tape Unit and Control
- IBM 3430 Magnetic Tape Unit and Control
- IBM 3803 Tape Control

Unit Record Devices

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

VM/SP HPO supports the following printers:

- IBM 1403 Printer Models 2, 3, 7, and N1
- IBM 1443 Printer Model N1 (with 144 print positions)
- IBM 3203 Printer Model 5
- IBM 3211 Printer (Right Indexing only)
- IBM 3262 Printer Models 1, 5 (in Model 1 emulator mode), and 11
- IBM 3289 Model 4 Printer
- IBM 3800 Printer Models 1, 3 (in Model 1 compatibility mode), and 8 (dedicated)
- IBM 4245 Printer
- IBM 4248 Printer Model 1
- IBM 4250 Printer

VM/SP HPO supports the following readers/punches:

- IBM 2501 Card Reader Models B1 and B2
- IBM 2520 Card Punch Models B2 and B3
- IBM 2540 Card Read Punch Model 1
- IBM 3505 Card Reader Models B1 and B2
- IBM 3525 Card Punch Models P1, P2, and P3

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

- IBM 2821 Control Unit
- IBM 3811 Printer Control Unit

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

- IBM 3250 Models 1 and 2

Terminals and Display Stations

The following **system consoles** are supported by VM/SP HPO as virtual machine consoles (**3215 emulation mode only**):

- IBM 2150 Console with 1052 Printer-Keyboard Model 7
- IBM 3066 System Consoles Models 1 and 2 for the System/370 Models 165 II and 168
- IBM 3210 Console Printer-Keyboard Models 1 and 2
- IBM 3215 Console Printer-Keyboard Model 1
- IBM System Console for the System/370 Model 158 in printer-keyboard mode (with the 3213 Printer Model 1 required)
- IBM 7412 Console (via RPQ AA2846) with 3215 Console Printer-Keyboard Model 1

The following **system consoles** are supported by VM/SP HPO as virtual machine consoles **(in 3215 emulation mode or in 3270 mode)**:

- IBM System Console for the System/370 Model 158 in display mode
- IBM 3036 Console with the 303x processor
- IBM 3278 Model 2A Display Console with the 4300 and 308x processors (3287 printer optional)
- IBM 3279 Model 2C Color Display Console with the 4300 processors (3287 printer optional).

Note: During system initialization only, the primary system operator's console cannot be connected to the system via a teleprocessing line.

The following **terminals** are supported by VM/SP HPO as virtual machine consoles **(in 3215 emulation mode only)**:

- IBM 2741 Communication Terminal
- IBM 1050 Data Communication System
- Terminals on switched lines compatible with the line control used by the IBM Telegraph Control Type II Adapter (8-level ASCII code at 110 bps) such as the CPT-TWX (Model 33/35) terminals
- IBM 3101 Display Terminal, Models 10, 12, 13, 20, 22, and 23 (supported as teletype Model ASR 33/35 teletypewriter)
- IBM 3767 Communications Terminal Models 1 and 2 (operating as a 2741).

The following **display stations** are supported by VM/SP HPO as virtual machine consoles **(in 3215 emulation mode or in 3270 mode)**:

- IBM 3275 Display Station, Model 2 with integral control unit (remote attachment only)
- IBM 3276 Control Unit Display Station Models 2, 3, and 4 with integral control unit
- IBM 3277 Display Station, Model 2, via 3272 Control Unit, Model 2 (local attachment only)
- IBM 3277 Display Station, Model 2, via 3271 Control Unit, Model 2 (remote attachment only)
- IBM 3277 Display Station Model 2 via 3274 Control Unit Models 1B and 1D (local attachment)
- IBM 3278 Display Station Models 2, 3, and 4 via 3274 Control Unit Model 1B (local attachment)

- IBM 3278 Display Station Models 2, 3, 4, and 5 via 3274 Control Unit Model 1D (local attachment)
- IBM 3278 Display Station Models 2, 3, 4, and 5 via 3274 Control Unit Models 1C and 51C
- IBM 3278 Display Station Models 2, 3, and 4 via 3276 Control Unit Models 2, 3, and 4
- IBM 3279 Color Display Station Models 2A, 2B, 3A, and 3B via 3274 Control Unit Models 1B and 1D (local attachment)
- IBM 3279 Color Display Station Models 2A, 2B, 2X, 3A, 3B 3X, S2B, and S3G via 3274 Control Unit Models 1C and 51C
- IBM 3279 Color Display Station Models 2A, 2B, 2X, S2A, and S2B via 3276 Control Unit Models 2, 3, and 4
- IBM 3279 Color Display Station Models 3A, 3B, 2X, and S3G via 3276 Control Unit Models 3 and 4
- IBM 3290 Information Panel via 3274 Control Unit Models XC and XD.

Notes:

1. *Any local non-SNA system console or terminal that defaults to being defined to the system as a 3277/3278, will work with directory entry "CONSOLE cuu 3270".*
2. *3215 console simulation for graphics devices excludes processing multiple output channel programs which contain CCWs without carriage returns (X'01' CCW op code) on one line of the screen. These channel programs are treated separately and VM/SP HPO uses a new line for each one.*
3. *When VM/SP or VM/SP HPO is installed at PUT 8402 or higher, attached 3274 Control Units must be at or above EC level PTR3174. 3274 Model XD requires EC325070. At lower EC levels, 3278 and 3279 Display Station Models 3 and 4 will be supported as Model 2 Display Stations.*

Special Considerations and Required Features

Terminals that are equivalent to those explicitly supported may also function satisfactorily. You are responsible for establishing equivalency. IBM assumes no responsibility for the impact that any changes to the IBM-supplied programs or products may have on such terminals.

2741 Communications Terminal Required Features: The IBM 2741 Communication Terminal is supported on either duplexed switched or point-to-point nonswitched lines connected to a Western Electric 103A2 (or equivalent data set). The following features are required features:

- PTTC/EBCD (#9571, Part #1167963) or standard Correspondence (#9812, Part #1167043) print elements
- Transmit Interrupt (#7900) or Transmit Interrupt Control RPQ #E40681

- Receive Interrupt (#4708)

- Data Set Attachment (#9114) and Dialup feature (#3255) for switched lines

- For point-to-point nonswitched lines, one of the following features is required:

  - Data Set Attachment (#9115 duplexed for facility D1) or
  - Data Set Attachment (#9116 duplexed for facility B2) or
  - Data Set Attachment (#9120 duplexed for facility B1 or D1) or
  - IBM Line Adapter (#4635 for 4-wire limited distance line)
  - IBM Line Adapter (#4691-4694 for 4-wire shared nonswitched line)
  - IBM Line Adapter (#4647 for 4-wire nonswitched line)

3767 Communications Terminal Required Features: The IBM 3767 Communication Terminal, Models 1 and 2, is supported when it operates as an IBM 2741 Communication Terminal and is attached to a 3704, 3705, or 3725 Communications Controller. It requires the following features on either switched or nonswitched point-to-point lines:

- 2741 START/STOP (#7113)

- EBCDIC (#9391) or Correspondence (#9381) keyboard

- Duplexed, switched or nonswitched line (#9404) for connecting to a Western Electric 103A2 (or equivalent data set)

- One of the following:

  - EIA Interface with Clock (#3719) at 300 bps
  - 1200 bps Integrated Modem/Interrupt (#5505 or #5500 or #5506)

1050 Control Units Required Features: The IBM 1050 Data Communication System is supported on either switched or point-to-point nonswitched lines with these features:

- IBM 1051 Control Unit (Model 1 or 2) with these features:

  - Transmit Interrupt (#7900) or Transmit Interrupt Control RPQ #E26903

  - Receive Interrupt (#6100) or Receive Interrupt Control RPQ #E27428

  - Text Time-Out Suppression (#9698)

  - First Printer Attachment (#4408). This feature is required to attach a 1052 Printer-Keyboard to the 1051.

- IBM 1052 Printer-Keyboard (Model 1 or 2) with the PTTC/EBCD print element (#9571, Part #1167963)

- For switched lines, the Data Set Attachment (#9114) is required.

- For point-to-point nonswitched lines, *one* of the following is required:
  - Data Set Attachment (#9115 for facility D1)
    or
  - Data Set Attachment (#9116 for facility B2)
    or
  - Data Set Attachment (#9120 for facility B1 or D1)
    or
  - IBM Line Adapter (#4691-4694 for 4-wire shared nonswitched line)
    or
  - IBM Line Adapter (#4647 for 4-wire nonswitched line)

3270 Information Display Stations Required Features: The 3271/3272 and 3274 Control Units are terminal control units that can attach up to 32 displays, serial matrix printers and/or line printers. These terminals are grouped into two categories. The Category A terminals are displays and printers that were developed for attachment to the 3274 Control Unit, while the Category B terminals were designed for attachment to the 3271/3272 Control Units. The 3274 Control Unit was designed to attach the Category B terminals with certain limitations. A maximum of 16 of the 32 attachable terminals can be Category B terminals. See *An Introduction to the IBM 3270 Information Display System*, GA27-2739, for further information.

ATTACHABLE TERMINALS

Category A Terminals

3230 Printer Model 2
3262 Line Printer Models 3,13
3278 Display Station Models 2,3,4,5
3279 Color Display Station Models 2A,2B,2X,3A,3B,3X,S2A,S2B,S3G
3287 Printer Models 1,2,1C,2C
3289 Line Printer Models 1,2
3290 Information Panel

Note: The 3290 Information Panel cannot be attached to port 0.

Category B Terminals

3277 Display Station Models 1,2
3284 Printer Models 1,2
3286 Printer Models 1,2
3287 Printer Models 1,2
3288 Line Printer Model 2

The basic 3271/3272 Control Unit provides one device adapter which permits attachment of up to four devices. Up to 32 devices may be attached in increments of four devices by adding up to seven Device Adapters (#3250).

The basic 3274 Control Unit allows the attachment of up to eight Category A terminals. Two categories of terminal adapters can be featured in various combinations to provide the maximum terminal configuration of 32 terminals. A maximum of 16 of the 32 terminals can be Category B units and at least one Category A Display Station with keyboard is needed for diagnostic purposes.

TERMINAL ADAPTER TYPE A1 THROUGH A3 (#6901,#6902,#6903)

One of each of these adapters can be installed. Each adapter provides for the attachment of an additional eight Category A terminals. These terminal adapters must be installed in sequence:

Terminal Adapter Type A1 - terminals 9-16
Terminal Adapter Type A2 - terminals 17-24
Terminal Adapter Type A3 - terminals 25-32

TERMINAL ADAPTERS TYPE B1 THROUGH B4
(#7802,#7803,#7804,#7805)

Terminal Adapter Type B1 permits the attachment of four Category B terminals and provides for the installation of Terminal Adapters Type B2, B3 and B4 when additional Category B terminals are desired. Terminal Adapters type B2 through B4 permit the attachment of four additional Category B terminals. A maximum of one of each of these terminal adapters can be installed for a combined total of 12 additional or 16 Category B terminals attached to a control unit. These terminal adapters must be installed in sequence:

Terminal Adapter Type B2 - terminals 5-8
Terminal Adapter Type B3 - terminals 9-12
Terminal Adapter Type B4 - terminals 13-16

The TERMINAL macro in the real I/O files requires that terminals on a Type A adapter must precede terminals on a Type B adapter.

For further information on 3270 Information Display Station, see the following manuals:

An Introduction to the IBM Information Display System, GA27-2739
Color and Programmed Symbols, GA23-3065
3274 Control Unit Description and Programmer's Guide, GA23-0061
3274 Control Unit Planning, Setup, and Customization Guide, GA27-2827
3274 Customizing Guide: Configuration Support D, GA23-0065
3274 Customizing Guide: Configuration Support P, GA23-0176
3276 Control Unit Display Station Description and Programmer's Guide, GA18-2081
3276 Control Unit Display Station Planning and Setup Guide, GA18-2041

Local Configurations Supported

CONTROL UNITS

The following control units can be attached locally on a byte multiplexer, block multiplexer, or selector channel to support 3270 devices:

- IBM 3272 Control Unit Model 1 or 2 for attachment of up to 32 Category B terminals.
- IBM 3274 Control Units Model 1B, 1D, 21B, 21D, 31D for the attachment of up to 32 display stations and printers. The 3278 Display Station Model 5 is not supported with the 3274 Control Unit Model 1B.

Remote Configurations Supported

The following control units can be remotely attached to leased lines via a 2701 Data Adapter Unit, 2703 Transmission Control Unit, or 3704/3705/3725 Communications Controller in emulation mode to support 3270 devices:

- IBM 3271 Control Unit Model 2 for attachment of up to 32 Category B terminals.
- IBM 3274 Control Unit Model 1C, 21C, 31C, or 41C for the attachment of up to 32 display stations and printers.
- IBM 3274 Control Unit Model 51C for the attachment of up to 12 display stations and printers.
- IBM 3274 Control Unit Model 61C for the attachment of up to 16 display stations and printers.
- IBM 3276 Control Unit Display Station Models 2, 3, and 4 for attachment of up to 7 additional display stations or printers:
  - 3278 Display Stations Model 2, 3, or 4. (3278 Model 4 cannot be attached to a 3276 Model 2 or 3.)
  - 3279 Color Display Stations Model S2A, S2B, S3G, 2X, or 3X. (Models S3G and 3X cannot be attached to 3276 Model 2.)
  - 3287 Printers Model 1, 1C, 2, or 2C. (Models 1C and 2C do not support extended color printing for the 3276.)
  - 3289 Printers Model 1 or 2.

IBM 3275 Display Station Model 2, 3, or 12 stand-alone control unit and display station. A 3284 Printer Model 3 can be attached. To support this configuration, the following are required:

- For the 3275 to be used on a switched line, Dial feature (#3440) is required. VM/SP HPO does not support full screen read/write to a 3275 with Dial feature.
- Transmission Speed feature (#7820 or #7821)
- A Printer Adapter feature (#5550), to attach a 3284 Printer Model 3

Transmission Control Units

VM/SP HPO supports the following transmission control units:

- IBM 2701 Data Adapter Unit
- IBM 2702 Transmission Control Unit
- IBM 2703 Transmission Control Unit
- IBM 3704, 3705-I, 3705-II and 3725 Communications Controllers

2701 Data Adapter Unit: Required Features

- For line control of CPT-TWX (Model 33/35) terminals and the 3101 display terminals, the Telegraph Adapter Type II (#7885) is required.
- For 2770, 2780, 3270, 3770 (as a 2770; 3776 also as a 3780), and 3780 terminals, the following are required:
  - Synchronous Data Adapter Type II (#7698)
  - EBCDIC code (#9060)
  - EBCDIC transparency (#8029)
- For 1050 and 2741 terminals, the following are required:
  - IBM Terminal Adapter Type I, Model II (#4640)
  - Selective Speed, 134.5 bps (#9581)
  - 2741 Break Feature RPQ #M53193, and Break Command RPQ #858492
- The Expanded Capability feature (#3815) is required if there are:
  - More than two low speed adapters (either IBM Type I Model II, or Telegraph Type II), or
  - More than one high speed adapter (Synchronous Data Adapter Type II), or
  - One high speed and at least one low speed adapter attached to the same 2701.
- The Expansion Feature (#3855) is required for each line adapter after the first.

2702 Transmission Control Unit: Required Features

- For 1050 and 2741 terminals, the following are required:
  - Terminal Control Base for IBM Terminal Control (#9696)
  - IBM Terminal Control Type I (#4615)
  - Selective Speed, 134.5 bps (#9684)
  - Type I Terminal Interrupt (#8200)
  - Data Set Line Adapter (#3233) or IBM Line Adapter (#4635), 4-wire IBM Terminal Control Type I (#4615)
- For line control of CPT-TWX (Model 33/35) terminals and the 3101 display terminals, the following are required:
  - Terminal Control Base for Telegraph Terminal Control (#9697)
  - Telegraph Terminal Control Type II (#7912)
  - Pluggable End Characters (return key generates an interrupt) RPQ #E62920, optional
  - Data Set Line Adapter (#3233)
  - Terminal Control Expansion (#7935), required only if both of the terminal bases (#9697 and #7912) are attached to the same 2702.
- The 31 Line Expansion (#7955) is supported as needed.

2703 Transmission Control Unit: Required Features

- For 1050 and 2741 terminals, the following are required:
  - Start-Stop Base Type I (#7505) or Type II (#7506)
  - IBM Terminal Control Base (#4619)
  - IBM Terminal Control Type I (#4696)
  - Line Speed Option, 134.5 bps (#4878)
  - Type I Terminal Interrupt (#8200)
  - Data Line Set (#3205) and/or IBM Line Set 1B (#4687)
- For line control of CPT-TWX (Model 33/35) terminals and 3101 display terminals, the following are required:
  - Telegraph Terminal Control Base (#7905)
  - Telegraph Terminal Control Type II (#7912)
  - Line Speed Option, 110 bps (#4877)
  - Data Line Set (#3205), and Data Line Set Expander (#3206)

- For 2770, 2780, and 3780 Terminals, the following are required:
  - Synchronous Base (#7703, 7704, or 7706)
  - Synchronous Terminal Control for EBCDIC (#7715)
  - Transparency (#9100)
  - Synchronous Line Set (#7710)
- The Base Expansion feature (#1440) is required if more than one base type is to be attached to the same 2703.

3704/3705/3725 Communication Controllers: Required Features

IBM 3704, 3705, and 3725 Communications Controllers are supported in Network Control Program mode, Partitioned Emulation Program mode, and 2701, 2702, 2703 Emulation Program mode.

Note: VM/SP HPO supports the CPT-TWX (Model 33/35) terminals at 110 bps and the 3101 display terminals at 110, 150, 300, and 600 bps, when attached to a 3704, 3705, or 3725.

VM/SP HPO supports all models of 3704, 3705, and 3725 Communications Controllers. However, because the network control program and partitioned emulation program require 48K of storage, the following models are supported for the emulation program only.

- IBM 3704 Communications Controller, Models A1 and A2
- IBM 3705-I Communications Controller, Models A1, B1, C1, and D1
- IBM 3705-II Communications Controller, Models E1, F1, G1, and H1
- IBM 3725 Communications Controller

The features required on a communications controller do not depend on VM/SP HPO. VM/SP HPO, when supporting network control mode, requires a communications controller with at least 48K of storage. Other 3704/3705/3725 features depend on the intended use of the communications controller and the type of 3704/3705/3725 control program (emulation, network control, or partitioned emulation program) to be executed.

VM/SP HPO does not support the following 3704/3705/3725 features:

- Line Set Type 2A (#4721)
- Line Set Type 3A (#4731)
- Line Set Type 4B (#4742)

Remote Spooling Communications Subsystem (RSCS)

Remote spooling is supported by the Remote Spooling Communications Subsystem (RSCS) component of VM/370. The RSCS Networking program product (5748-XP1) expands the capabilities of VM/370 RSCS. It is recommended that RSCS Networking be used with VM/SP HPO.

The VM/370 Remote Spooling Communications Subsystem (RSCS) supports the following:

- IBM 2770 Data Communication System
- IBM 2780 Data Transmission Terminal, Models 1 and 2
- IBM 3770 Data Communication System (as a 2770)
- IBM 3780 Data Communication Terminal
- HASP supported programmable workstations

2770 Data Communication System: Required Features

The IBM 2770 Data Communication System with the 2772 Multipurpose Control Unit can be connected to the central System/370 via a switched or nonswitched point-to-point communication line.

The following devices and features are required for operating a 2770 as an RSCS nonprogrammable terminal:

- One IBM 2213 Printer, Model 2, or one IBM 2203 Printer, or one IBM 1053 Printer
- One IBM 2502 Card Reader, Model A1 or A2
- EBCDIC Transmission Code (#9761)

2780 Data Transmission Terminal: Required Features

The IBM 2780 Data Transmission Terminal, Models 1 and 2, can be connected to the central System/370 via a switched or nonswitched point-to-point line. EBCDIC Transmission code (#9762) is required.

3770 Data Communications System: Features

The IBM 3770 Data Communication System can be connected to the central System/370 via a switched or nonswitched point-to-point communications line. Required features are:

- EBCDIC Transmission Code (#9761)
- SDLC/BSC Switch Control (#1460), or BSC point-to-point (#1461)

3780 Data Communications Terminal: Features

The IBM 3780 Data Communications Terminal can be connected to the central System/370 via a switched or nonswitched point-to-point communications line. EBCDIC transmission code (#9761) is required.

Programmable Terminals

RSCS Spool MULTI-LEAVING (SML) supports, as a remote workstation, any processor that is supported as a HASP workstation and is programmed to operate as a HASP workstation.

Processors Supported

- IBM 1130 Computing System (except Models 4A and 4B) is supported if it has at least 8K words of storage and the Synchronous Communications Adapter (#7690).
- Any System/360 or System/370 is supported if it has at least 8K bytes of main storage and a 2701, 2703 or 3704/3705/3725 in EP mode, equipped for EBCDIC transmission and binary synchronous communications.
- Any submodel of the System/360 Model 20 or the IBM 2922 is supported if it has at least 8K bytes of main storage and the following features:
  - Binary Synchronous Communications Adapter (#2074)
  - EBCDIC Transmission code (#9060)
  - Full Transparency (#4100)
- IBM System/3 Models 6, 8, 10, 12, and 15 are supported with the following features:
  - Binary Synchronous Communications Adapter (#2074)
  - EBCDIC Transmission code (#9060)
  - Text transparency (#7850)
- IBM System/32 is supported with the following features:
  - 5320 System Unit (any model A12 through B33)
  - Binary Synchronous Communications Adapter (#2074)
  - System Control Program (5725-SC1)

Other Considerations for Planning Your Configuration

Two-Channel Switch

If any I/O devices controlled by VM/SP HPO for its exclusive use are attached to control units with two-channel switches, the processor or virtual machine controlling the other channel interface must vary the CP-owned devices offline.

See "Channel Switching" on page 91 in Chapter 10, "Planning for I/O" for more information about using the two-channel switch.

Multisystem Communication Unit

The IBM Multisystem Communication Unit is an input/output (I/O) device that interconnects multiple systems by means of channels attached to a block multiplexer channel. The 3088 is designed to be fully compatible with existing channel-to-channel adapters (CTCAs). The 3088 may be used to form a loosely coupled multiprocessing system connecting as many as eight processors.

In configurations using the 3088, the device must be defined using the ADDRESS and DEVTYPE operands of the RDEVICE macro, and the ADDRESS, CUTYPE, and FEATURE=xxx-DEVICE operands of the RCTLUNIT macro.

Unsupported Devices

Virtual machine users may be able to use I/O devices that VM/SP HPO does not support. An unsupported device is a device type that is not listed in the DEVTYPE operand of the RDEVICE macro instruction. To use an unsupported device, a user must attach or dedicate the device to a virtual machine. A dedicated device is one that is not shared among users, but is used exclusively by one user. CP supports these dedicated devices only under these conditions:

- No timing dependencies exist in the device or the program.
- No dynamically modified channel programs exist in the access method, except when OS ISAM or OS/VS TCAM Level 5 are used.[2]
- No special functions need to be provided by VM/SP HPO.
- None of the other CP restrictions are violated. (Refer to Appendix D, "VM/SP HPO Restrictions.")
- The device is generated into the CP nucleus (by using the RDEVICE macro instruction with the appropriate CLASS operand).

---

[2] A virtual=real virtual machine can use dynamically modified channel programs in their access methods if (1) no input/output involves page zero and (2) the virtual machine operator issues the CP command SET NOTRANS ON.

I/O devices that are part of a virtual machine's configuration require real device equivalents. However, exceptions to this rule are:

- Unit record devices, which CP can simulate by using spooling techniques.
- Virtual 2311 disks, which CP can map onto 2314 or 2319 disks. One to two full 2311 units can be mapped onto a 2314 or 2319 disk in this manner.

Service Record File

On 303x processors, each console station of the 3036 system console has a 7443 diskette attached to it, accessible when the console station is in SRF mode. In the normal console configuration, one of the processor's console stations is used as an operator's console, and the other console station is used as a service console. It is through the service console that SRF capability is provided. With the console in degraded configuration (i.e. one console station serves as both operator and service console), there is no SRF capability. It is recommended that the SRF address specified on the RIOGEN macro instruction at system generation be the address of the service record file attached to the service console.

In a 3033 attached processor or multiprocessor system, there are two 3036 consoles. This configuration has four service record file devices (one console per station).

3033 attached processor and multiprocessor environments support multiple service record file devices. For VM/SP HPO systems operating on a 3033AP or 3033MP, specify multiple SRF devices at system generation. Code DEVTYPE=7443 in the RDEVICE macro statement and CUTYPE=7443 in the RCTLUNIT macro statement to generate support for the SRF devices. Also code the ADDRESS=ccuu operand in both macro statements. Identify the SRF device addresses in the RIOGEN macro statement as SRF=(ccuu,ccuu,...). The SRF addresses you specify in the RIOGEN macro statement should be the same as the addresses of the SRF devices attached to the service support consoles.

In 3033 AP or MP environments with I/O configured asymmetrically to one processor, in order to access the SRF devices in both 3036 consoles, a channel path must be available from the I/O processor to both SRF devices.

If an SRF device specified on the RIOGEN macro statement is found to be inaccessible during initialization of the error recording cylinders, an error message is sent to the system operator. Processing continues without frames from that SRF device in place on the error recording cylinders.

The RIOGEN macro statement produces an MNOTE warning message if you specify more than 32 SRF devices.

Processor Controller

The 3081 Processor Complex and the 3090 Processor Complex use a processor unit and a processor controller to control system operations. The processor controller is a service processor that defines the I/O configuration to the processor complex. To accomplish this, the processor controller requires information about the real system configuration. You can define this information to the controller by executing the Input/Output Configuration Program. For more information about the Input/Output Configuration Program, see Chapter 10, "Planning for I/O."

Vector Facility

The Vector Facility is a high-speed vector/scalar instruction processor which can be attached to the 3090 Processor Complex. It is especially useful for FORTRAN programming and allows calculation with either fixed-point or floating-point values. A given Vector Facility can be attached only to one processor; but each processor in a configuration can attach one Vector Facility.

Chapter 3. Estimating CP Storage Requirements

This section contains information about:

- Estimating real storage requirements for VM/SP HPO
- Reducing the size of the CP nucleus
- Estimating direct access storage requirements
- Estimating storage requirements for CMS minidisks

"Specifying a Virtual=Real Machine" on page 119 includes information about estimating real storage requirements for a virtual=real machine.

Real Storage Requirements for CP

Figure 3 lists various CP requirements and the amount of real storage required for each.

CP Requirement	Real Storage Allocated
Resident nucleus	Approximately 370K[3]
Internal trace table	Conventionally, 4K of storage is allocated for each 256K of real storage up to 16 megabytes, and 4K per megabyte above 16 megabytes. This storage is set aside at IPL time. See "SYSCOR Macro" on page 349 for details of how to increase the size of the internal trace table.
Real control blocks	There is a control block for each real device, control unit, and channel: • 144 bytes/real device • 80 bytes/real control unit • 96 bytes/real channel • 56 bytes for each remote 3270 or real 3704/3705/3725.

Figure 3 (Part 1 of 2). Real Storage Requirements for CP

[3] An additional 40K of real storage is allocated in AP or MP mode.

CP Requirement	Real Storage Allocated
Virtual device tables	The amount of contiguous storage required for the virtual device tables is given by multiplying the size in bytes of the virtual device block (VDEVBLOK) by the number of virtual devices. (The current size of the VDEVBLOK is 80 bytes.)
Permanently allocated free storage (for installation control of free storage, use the SYSCOR macro). For the format of the SYSCOR macro, see "SYSCOR Macro" on page 349. *Note: The VMBLOK allocation is taken from DPA.*	The default value is a minimum of 12K, plus an additional 4K for each 64K of real storage above 256K[4] up to 16 megabytes, and 4K for each 256K above 16 megabytes. This storage is set aside at IPL time. Each logged-on virtual machine requires a virtual machine control block (VMBLOK), a segment table (SEGTABLE), a page table (PAGTABLE), a swap table (SWPTABLE), and a control block for each virtual device, control unit, and channel. A retrieve buffer (RETBUF) is required if the PF key Retrieve Function is used. The storage requirement is: • 752 bytes for the VMBLOK • 64 bytes/1M of virtual storage for the SEGTABLE • 48 bytes/64K of virtual storage for the PAGTABLE • 136 bytes/64K of virtual storage for the SWPTABLE • 80 bytes/virtual device • 40 bytes/virtual control unit • 48 bytes/virtual channel • 160 bytes for the retrieve buffer on most 3270s; 256 bytes for the retrieve buffer on a 3278-5; 0 bytes on other devices or if the retrieve function is not used.

Figure 3 (Part 2 of 2). Real Storage Requirements for CP

For example, if you have:

1M of real storage
29 real devices
6 real control units
3 real channels

and 12 virtual machines defined, each with:

1 virtual reader
1 virtual printer
1 virtual punch
3 virtual disks
3 virtual channels
1 virtual machine console
3 virtual control units
512K of virtual storage

---

[4] An additional 25% of free storage is allocated in AP or MP mode.

You would estimate CP real storage requirements as follows:

370K	for the CP resident nucleus
16K	for the CP internal trace table (see "SYSCOR Macro" on page 343)
5K	for the real control blocks, calculated as follows: 144 x 29 = 4176 bytes for the real devices 80 x 6 = 480 bytes for the real control units 96 x 3 = 288 bytes for the real channels the sum is: 4176 + 480 + 288 = 4944 bytes (approximately 5K)
60K	for permanently allocated free storage (default value)
451K	real storage required.

Also, as each of the 12 virtual machines defined logs on, a little more than 3K of real storage is allocated to each from the permanently allocated free storage. A breakdown of this 3K of real storage is:

704	bytes for a VMBLOK
64	bytes for the SEGTABLE
384	bytes for the PAGTABLE (512K/64K x 48 bytes)
1088	bytes for the SWPTABLE (512K/64K x 136 bytes)
80	bytes for a virtual reader
80	bytes for a virtual printer
80	bytes for a virtual punch
240	bytes for three virtual disks
144	bytes for three virtual channels
80	bytes for a virtual machine console
120	bytes for three virtual control units
3064	bytes for each of the logged-on users defined

See "Specifying the Virtual=Real Storage Size" on page 117 for an example of estimating storage requirements and determining the maximum virtual=real area size.

Reducing the CP Nucleus Size

There are four load lists, each for a particular system configuration. See Figure 4.

System Configuration	Load List (No V=R Area)	Load List (V=R Area)
UP	HnlCPLOD	HnlVRLOD
AP or MP	HnlAPLOD	HnlAVLOD
The letter n is the release level of VM/SP HPO and l is the modification level. If there is no modification level, l=0.		

Figure 4. Load Lists and System Configurations

If you want a smaller CP nucleus, you can tailor the load list to your own specifications. Edit the load list to remove only modules that are associated with functions that you do not require. For more information, see the *VM/SP HPO System Programmer's Guide.*

Figure 5 shows support and associated modules as well as the names that are undefined if those modules are deleted from the load list.

Exercise caution before deleting the modules from the load list. If you generate a system that includes a device in the I/O configuration, you cannot remove the modules associated with that device from the load list. If you do remove the modules listed, unpredictable results may occur.

If you do not use the NAMENCP macro statement, label DMKSNTRN is undefined during the VMFLOAD procedure.

If you do not use the NAME3800 macro statement, label DMKSNTQN is undefined during the VMFLOAD procedure.

Support	Modules	Names
Local 3270	DMKGRF	DMKGRFEN, DMKGRFIC, DMKGRFIN, DMKGRFMT
Remote 3270	DMKRGA, DMKRGB, DMKRGC, DMKRGD, DMKBSC	DMKRGAIN, DMKRGBIC, DMKRGFMT DMKRGBFM, DMKRGBEN, DMKBSCER DMKRGDOI, DMKRGDOB
3340	DMKTRK	DMKTRKIN, DMKTRKFP, DMKTRKVA
3375/3380 error recovery	DMKDAD	DMKDADER
3850 MSS	DMKSSS DMKSST DMKSSU	DMKSSSMQ, DMKSSSEN, DMKSSSVA, DMKSSS, DMKSSULO, DMKSSTHV, DMKSSUI1, DMKSSSRL, DMKSSVI2 DMKSSSAS, DMKSSSL1, DMKSSSL2, DMKSSSL3, DMKSSSLN, DMKSSSDE, DMKSSUCF, DMKSSSUS, DMKSSSVM
3704/3705	DMKRNH	DMKRNHIC, DMKRNHIN, DMKRNH DMKRNHND, DMKRNHTR
Preferred Machine Assist	DMKPMA	DMKPMAXF, DMKPMACD, DMKPMAEX, DMKPMAI1, DMKPHAXS, DMKPMAI2, DMKPMAIN, DMKPMAMC, DMKPMASS DMKPMASW, DMKPMAWT
V=R area	DMKVSC	DMKVSCSC, DMKVSCVR,
CP Quiesce virtual machine command	DMKQVM	DMKQVMRT, DMKQVMEP, DMKQVMRS, DMKUVMRX
SNA CCS	DMKVCP, DMKVCR, DMKVCT, DMKVCV, DMKVCX	DMKVCPIL, DMKVCRNR, DMKVCRRD, DMKVCRWT, DMKVCTLO, DMKVCTCH, DMKVCTER, DMKVCTEN, DMKVCTDA, DMKVCTCN, DMKVCTSV, DMKVCTQS, DMKVCTRM, DMKVCVER, DMKVCXIO DMKVCR
Missing Interruption Handler	DMKDID	DMKDIDDA, DMKDIDGR, DMKDIDTA, DMKDIDUR, DMKDIDMS, DMKDIDTR, DMKDIDEP
3066 Display	DMKGRH	DMKGRHIN
Special Cases of privileged Instructions	DMKFPS	DMKFPS
Translates Contask data	DMKTTY	DMKTTYOP, DMKTTYIN

Figure 5. Reducing the Size of the CP Nucleus

Direct Access Storage Requirements for CP

In the following paragraphs and in "Part 2. Defining Your VM/SP HPO System" there are many references to the concept of "DASD space."

It is helpful to understand CP's requirements for DASD space in general. It is also helpful to understand the differences and similarities between CP's view of count-key-data (CKD) DASD and fixed block (FB-512) devices:

CKD - (2314, 2319, 3330, 2305, 3340, 3350, 3375, and 3380)
FB-512 - (3310 and 3370)

CP's reference to DASD space is always done in units called DASD pages. A DASD page is 4096 bytes of contiguous DASD storage. This means that CP requires that all its DASD space (nucleus, error recording, warm start data, checkpoint data, directory, saved systems, dump space, paging, and spooling space) be formatted as 4096 byte records (pages). Also CP requires that you identify what specific pages on DASD are allocated to each type of CP reference. For example, you must identify pages for the nucleus, for paging, for the directory, and so forth.

CP provides the format/allocate service program (DMKFMT) to perform these formatting and allocating functions. A DASD volume containing any of the types of space listed above is called a CP volume and must be processed by the format/allocate service program. Space not used by CP on CP owned volumes is available to you. Typically, although not necessarily, this space is used for user minidisks. CP has no format requirements for this space, but does require that it be allocated.

DASD Characteristics

CKD Device Geometry

CP views CKD devices by their geometric characteristics, e.g. as a certain number of cylinders. Each cylinder has a given page capacity, meaning that a given number of 4K records "fit" on a cylinder. This number varies for each CKD DASD type. CP references its data by a cylinder number and a page number on that cylinder. For CP space you must format and allocate pages in groups of cylinders. In "Part 2. Defining Your VM/SP HPO System," you are asked to compute your particular DASD requirements as a number of records or pages, then convert this number to an equivalent number of cylinders. This conversion is simply dividing the page requirement by the number of pages per cylinder for your particular device type. Allocating space for minidisks on CP owned volumes is also done in units of cylinders. Use of space within this allocation is done in units of cylinders via the MDISK directory control statement. This is convenient in that no conversion is required in this case.

FB-512 Device Geometry

FB-512 devices appear as a linear address space of 512-byte blocks. The blocks are consecutively numbered from 0 to n, where n is the highest block number on the volume. CP groups 8 consecutive blocks to form a CP page. CP then views the volume as a collection of pages numbered from 0 to n/8. For example, blocks 0-7 make up page 0. There is no concept of cylinder boundaries in this structure.

You must allocate space on FB-512 volumes in units of pages (contrasted to the unit of cylinder on CKD). When you compute your DASD space requirements as a number of pages, you can use these numbers directly in the SYSRES macro and in the format/allocate service program (DMKFMT). No conversion to other units is required. When allocating space on FBA devices, allocation should be on pseudo-cylinder boundaries. See "FBA Page/Pseudo-Cylinder Conversion" on page 35.

Although convenient for CP DASD space, this causes an inconvenience when assigning minidisks. This is caused by the difference in the unit of input between the format/allocate service program (pages) and the MDISK control statement (blocks). When assigning minidisk space, you must know the extents of the available space in block numbers. Be careful that you input to format/allocate in page numbers and assign minidisks by block number.

To obtain the block number, take the allocation results, which show page numbers, and multiply the page number by 8. For example, in the sample layout for a 3310 shown in Figure 6, pages 2000 through 9999 were allocated as PERM space for use as minidisks. The allocation results would show:

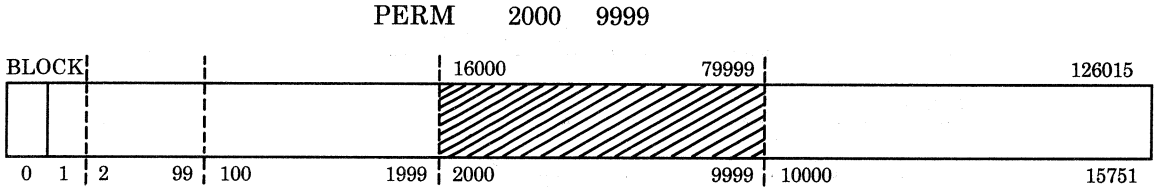


Figure 6. Sample Layout of a 3310

By doing the following calculation this corresponds to block numbers 16000 through 79999.

For the beginning block number:

Multiply 2000 (1st page) X 8 (8 blocks per page) = 1600 (beginning block number)

For the ending block number:

Multiply 9999 X 8 = 79992, which is the 1st block of the last page.

Add 79992 + 7 (remaining blocks in last page) = 79999 (last block in last page)

You can now use these block numbers on the MDISK statement.

FBA Page/Pseudo-Cylinder Conversion

When you code the SYSPAG macro(s), compute your DASD space requirements as a number of pseudo-cylinders. A method of converting pages (or blocks) of an FBA device to pseudo-cylinders is given below for a 3310 and a 3370 device. A pseudo-cylinder corresponds to an access position, a block represents 512 bytes and a page consists of eight blocks.

Device Characteristics	3310	3370-1	3370-2
Number of blocks	126,016	558,000	712,752
Number of access positions	358	750	958
Blocks/access position	352	744	744
Pages/access position	44	93	93

Figure 7. FBA Blocks and Pages per Access Position

To determine which pseudo-cylinder a particular page or block is on, use one of the following algorithms:

pseudo-cyl no. = (page no.) / pages per access position
pseudo-cyl no. = (block no.) / blocks per access position

Below are tables showing the relationship between block and page number ranges and pseudo-cylinder numbers for the 3310 and 3370.

3310

Pages	Blocks	Pseudo-Cylinder
0 - 43	0 - 351	0
44 - 87	352 - 703	1
88 - 131	704 - 1055	2
. .	. .	.
. .	. .	.
. .	. .	.
15,708 - 15,751	125,664 - 126,015	357

3370-1

Pages	Blocks	Pseudo-Cylinder
0 - 92	0 - 743	0
93 - 185	744 - 1487	1
186 - 278	1488 - 2231	2
. .	. .	.
. .	. .	.
. .	. .	.
69,657 - 69,749	557,256 - 557,999	749

3370-2

Pages	Blocks	Pseudo-Cylinder
0 - 92	0 - 743	0
93 - 185	744 - 1487	1
186 - 278	1488 - 2231	2
. .	. .	.
. .	. .	.
. .	. .	.
89,001 - 89,093	712,008 - 712,751	957

Figure 8. Converting Block and Page Numbers to Pseudo-Cylinder Numbers

- Pages 0 and 1 are reserved (see the *VM/SP HPO Operator's Guide* for details)
- Pages 2-99 for directory (DRCT)
- Pages 100-1999 for nucleus and error recording (PERM)
- Pages 2000-9999 minidisks (PERM)
- Pages 10000-15751 for paging and spooling (TEMP)

The following is an example of defining three minidisks within the range of PERM space:

```
mdisk 191 16000 2000 LABEL
mdisk 19F 18000 5000 LABEL
mdisk 296 23000 7000 LABEL
```

Figure 9 shows CP DASD space requirements by DASD type. The following paragraphs describe in detail how you determine the DASD space CP requires for the nucleus, error recording, warm start data, checkpoint data, directory, saved systems data, paging, and spooling space.

	2319/2314	3330	2305	3340	3350	3375	3380	FB-512
CP Nucleus	varies	varies	varies	varies	varies	varies	varies	varies
Error Recording[1]	1	2	2	2	2	2	2	114
Warm Start	1	1	1	1	1	1	1	57
Checkpoint Start	2	1	3	3	1	1	1	57
Directory	2	4	2	10	2	4	2	varies
Override Space	optional	optional	optional	optional	optional	optional	optional	optional
Saved Systems	varies	varies	varies	varies	varies	varies	varies	varies
Paging Space	32	18	40	40	10	10	11	1000
Spooling Space	50	30		70	15	15	20	1700
Total System[2]	89 cyl	56 cyl	48 cyl	126 cyl	31 cyl	33 cyl	37 cyl	3156 pgs

[1]The default is 2 cylinders but up to 9 cylinders may be specified via the SYSERR operand of the SYSRES macro.
[2]These figures do not include space for the nucleus or saved systems. For FB-512 devices, directory space is not included.

Figure 9. DASD Space Requirements by DASD Type

CP Nucleus DASD Requirements

The CP nucleus (without a virtual=real area) requires about 350 pages of disk space for resident and pageable functions.

To determine the number of cylinders required for the CP nucleus, refer to the load map produced during system generation. One DASD page is required for each page of fixed and pageable nucleus. When you have a system with a V=R area, you should subtract the page numbers of the V=R area from the last module page number to find the size of CP in pages.

For example, if the last module entry in the load map ends at page 15E (hexadecimal), and if you have no V=R area, 350 pages of disk space are required for CP nucleus residence. The number of cylinders required depends on the system residence device used; see the "Saved System DASD Requirements" section that follows for the number of pages per cylinder each device can accommodate.

Normally, the number of cylinders required for CP nucleus residence is:

8 cylinders on a 3330 or 3333
3 cylinders on a 3350
4 cylinders on a 3375
3 cylinders on a 3380

Error Recording DASD Requirements

Error recording space varies from 2 to 9 and is established by the SYSERR operand of the SYSRES macro instruction as described in Chapter 23, "Preparing the CP System Control File (DMKSYS)."

Warm Start Data DASD Requirements

Formulas for calculating the warm start space needed are under the discussion of the SYSWRM operand of the SYSRES macro in Chapter 23, "Preparing the CP System Control File (DMKSYS)."

Checkpoint Start Data DASD Requirements

The space required for dynamic checkpointing of the spool file system is discussed in "SYSRES Macro" on page 335.

Dump Space DASD Requirements

This space is optional. If you want to allocate dump space for use by CP, format the area and allocate it as DUMP using the format/allocate program. The size should normally be large enough to contain all of real storage. If no DUMP space is allocated, spooling space (TEMP) is used for dumps.

To calculate the approximate amount of storage needed for DASD dump space, use this formula:

```
number of pages of real memory + 5
```

The result is the number of pages needed. For CKD devices, convert the result to cylinders by dividing it by the number of pages per cylinder.

VM/SP HPO Directory DASD Requirements

The directory normally requires two cylinders so that it can be rewritten without disturbing the active directory and swapped after a successful update. Equations for computing directory sizes are found in "Allocating DASD Space for the Directory" on page 189.

Saved System DASD Requirements

In order to save page image copies of a virtual machine, you must reserve space on a CP-owned volume for the saved system. Saved systems require one page for each page saved, plus an additional information page or pages. The information pages are used by CP to save the virtual machine's register contents, PSW, and storage keys. The number of information pages required depends on the number of virtual machine pages being saved. The following chart shows the relationship.

Number of Virtual Machine Pages Being Saved	Number of Information Pages Required by CP
1 - 1948	1
1949 - 3896	2
3897 - 4096	3

Figure 10. Information Pages Required for Saved Systems

Note: Saving a system on a 3704/3705 may require up to four additional information pages.

For example, to save a complete 16M virtual machine, you must reserve 4099 pages on DASD:

```
4096 pages to be saved + 3 information pages
```

When you code the NAMESYS macro, set the SYSPGCT parameter to the exact number of pages being saved, not counting information pages. In the above example, set SYSPGCT to 4096. The SYSSTRT parameter indicates the starting DASD location where the 4099 pages will be saved.

Warning: To keep the saved system from overlapping destructively with another system, remember that information pages are allocated.

Saving one copy of the CMS system requires:

5 cylinders on a 2413 or 2319
3 cylinders on a 3330 or 3333
6 cylinders on a 3340
2 cylinders on a 3350
2 cylinders on a 3375
1 cylinder on a 3380
138 pages on an FB-512

Paging and Spooling DASD Requirements

Paging and spooling space requirements are installation-dependent. (Values shown in Figure 7 are for average systems.) Paging space is allocated at a rate of:

24 pages/cylinder on a 2305 or 3340
32 pages/cylinder on a 2314 or 2319
57 pages/cylinder on a 3330 or 3333
120 pages/cylinder on a 3350 in native mode
96 pages/cylinder on a 3375
150 pages/cylinder on a 3380
(The 2305 is normally used for paging only.)

Paging space is dependent on the number and size of logged-on virtual machines, processor storage size, and workload. In general, the following calculation yields adequate paging space:

```
number of logged-on users x average virtual machine size
(in pages) = number of pages needed.
```

Thus, a 1M processor with 10 logged-on users having an average virtual machine size of 500K would require 994 pages of DASD space for paging [(10 x 125) - 256 = 994]. This would take 18 cylinders of a 3330 (994/57 = 17.4).

Spooling data is placed in a 4K buffer with the necessary channel programs required for each record. Data capacity of spooling cylinders thus varies with the data and CCWs used.

The examples in Figure 3 assumed a maximum of 200 spool files of 8 to 9 blocks each. If separate DUMP space is not allocated, spooling space (TEMP) will be used for dumps.

The primary system operator is warned when the paging/spooling space becomes 90% full. *VM/SP HPO System Messages and Codes* tells the operator what to do if this warning occurs.

Facilities exist to dump spool files to tape when the spool space is full or nearly full. When spool space is again available, the system operator can restore the dumped spool files to the system for subsequent processing.

VSAM and Access Method Services Requirements

VSAM and access method services support in CMS requires both DASD space and virtual storage.

The DASD space needed is listed under "Loading and Saving Discontiguous Segments" in the *VM/SP HPO Installation Guide.*

VSAM and Access Method Services support adds approximately 2K to the size of the CMS nucleus. In addition, this support uses free storage to execute the VSE logical transients, and for buffers and work areas. VSAM issues a GETVIS macro to request free storage.

If the CMS/DOS environment is invoked with the VSAM option

```
SET DOS ON (VSAM
```

part of the CMS/DOS virtual storage is set aside for VSAM use.

Vector Facility Requirements

For each virtual machine using the Vector Facility, a two-page save area is reserved in real storage, usually above the 16-megabyte line. This save area is returned when the virtual machine logs off or issues the IPL CLEAR or SYSTEM CLEAR command.

IPCS Considerations

External Storage

Disk storage needed by IPCS is divided into two parts. The first part contains the IPCS command modules, the current NUC MAP, problem reports, and the symptom summary. These files occupy less than 5 cylinders on a 3330, allocated as shown:

- 100 problem reports plus symptom summary - 20%
- NUC MAP - 45%
- IPCS modules - 35%

The first part does not vary greatly (only problem reports and symptom summary are affected).

The second part contains dumps. The size of a dump depends mainly on the size of the system being dumped, and the operand of the CP SET DUMP command, either ALL or CP. The table below shows typical space usage by device type for the fixed area and for one dump.

	Cylinders						Blocks
Files	**3330**	**2319**	**3340**	**3350**	**3375**	**3380**	**FB-512**
IPCS modules NUC MAP 100 problem reports symptom summary	5	9	13	3	4	2.5	2090
512K CP	1.5	3	4	1	1	1	628
512K ALL	2.5	5	7	1.5	2	1	1045
1024K CP	3	6	8	1.5	2	1.5	1254
1024K ALL	6	11	16	3	4	2.5	2508

Figure 11. External Storage Requirements for IPCS

Real Storage

The real storage requirement is the normal CP requirement of approximately 2K for control blocks while the IPCS virtual machine is logged on. Other real storage usage is controlled by demand paging implementation.

Estimating DASD Storage Requirements for CMS Minidisks

The following information is intended to help you allocate enough direct access storage space for CMS minidisks.

Device Type	Approximate 80-byte lists
2314	1300 per cylinder
3310	6.4 per 512-byte block
3330	2300 per cylinder
3340	960 per cylinder
3350	5000 per cylinder
3370	6.4 per 512-byte block
3375	3200 per cylinder
3380	6250 per cylinder

Each physical disk contains file control information as well as your data. A given amount of data requires more file control information if put into many small files instead of a few large files.

For an average CMS user, the following minidisk space should be sufficient:

Device Type	Space Required
2314	7 cylinders
3310	1700 512-byte blocks
3330	4 cylinders
3340	11 cylinders
3350	2 cylinders
3370	1700 512-byte blocks
3375	3 cylinders
3380	2 cylinders

Chapter 4. Planning for CMS

The Conversational Monitor System (CMS) provides conversational facilities for virtual machine users. CMS operates only in a virtual machine, and together with CP, provides a time-sharing system suitable for program development, problem solving, and general time-sharing work.

CMS is required to support CP. VM/SP HPO adds no new function to CMS.

This section contains the following information about CMS:

- Storage Requirements
- Device Support
- Libraries
- Command Language
- Program Language Facilities
- Limited Support of DOS and OS
- Disk and File Management
- Tape Support
- Unit Record Support
- Editing
- Batch Facility
- Saving CMS

CMS Storage Requirements

CMS requires virtual storage and auxiliary storage.

Virtual Storage

A minimum of 256K of virtual storage is required for a CMS virtual machine. This virtual storage is distributed as follows:

- CMS buffers, pointers, and control blocks (DMSNUC) – 20K
- Loader tables – 8K (for virtual machines with up to 384K of virtual storage) and 12K (for virtual machines with more than 384K of virtual storage)
- User area – 120K (for application programs or CMS disk-resident commands)
- CMS free storage – 100K
- Transient area – 8K (CMS disk-resident commands).

Auxiliary Storage

The CMS auxiliary storage requirements are:

- System residence for CMS (190 minidisk):

  158 cylinders on a 2314 or 2319
  100 cylinders on a 3330 or 3333
  236 cylinders on a 3340 Model 35 or Model 70
  49 cylinders on a 3350 in native mode
  45 cylinders on a 3380
  74 cylinders on a 3375
  45,568 blocks on an FB-512

- Resident disk space for application programs (CMS commands, user programs, IBM program products). The amount of space is program-dependent. You must assign it.

- Work space for application programs (CMS commands, user programs, IBM program products). The amount of space is program-dependent. You must assign it.

Device Support

CMS supports the virtual machine devices shown in Figure 12.

Under CP, unit record device types and the system console may be simulated and mapped to addresses and device types other than the real ones. For instance, CMS expects a 3215, 3210, 1052, or 3270 type of operator's console, but many terminals are 2741s or 3270s. Regardless of the real device type, the virtual system console is a 3215. The control program (CP) handles all channel program modifications necessary for this simulation. CMS virtual disk addresses are mapped by CP to different real device addresses.

The read-only CMS system disk (S-disk), normally located at virtual address 190, contains the CMS nucleus functions and disk-resident CMS command modules. The CMS nucleus is loaded into virtual storage when you issue the CP IPL command. CMS remains resident until you enter another IPL command or until you log off. The disk-resident modules are loaded into virtual storage only when their services are needed.

In addition to the system disk (S-disk) and primary disk (A-disk), each CMS user can have up to 24 additional disks. The read/write A-disk is the primary user disk. Files that you wish to retain for later use are stored on one of your disks. Information stored on a disk remains there until you erase it. An exception is the temporary disk; files written on this disk are lost when you log off. For more information about CMS disks and their use see the *VM/SP CMS User's Guide.*

Virtual IBM Device	Virtual Address[1]	Symbolic Name	Device Type
3210,3215,1052,3270	cuu[2]	CON1	System console
2314,3310,3330,3340,3350, 3370,3375,3380	190	DSK0	System disk (read-only)
" " "	191[3]	DSK1	Primary disk (user files)
2314,2319,3310,3330,3340, 3350,3370,3375,3380	ccu	DSK2	Disk (user files)
" " "	ccu	DSK3	" "
" " "	192	DSK4	" "
" " "	ccu	DSK5	" "
" " "	ccu	DSK6	" "
" " "	ccu	DSK7	" "
" " "	19E	DSK8	" "
" " "	ccu	DSK9	" "
" " "	ccu	DSKH	" "
" " "	ccu	DSKI	" "
" " "	ccu	DSKJ	" "
" " "	ccu	DSKK	" "
" " "	ccu	DSKL	" "
" " "	ccu	DSKM	" "
" " "	ccu	DSKN	" "
2314,2319,3310,3330,3340, 3350,3370,3375,3380	ccu	DSKO	Disk (user files)
" " "	ccu	DSKP	" "
" " "	ccu	DSKQ	" "
" " "	ccu	DSKR	" "
" " "	ccu	DSKT	" "
" " "	ccu	DSKU	" "
" " "	ccu	DSKV	" "
" " "	ccu	DSKW	" "
" " "	ccu	DSKX	" "
1403,3203,3211,1443,3800, 3262	00E	PRN1	Line printer
2540,2501,3505 00C	RDR1	Card reader	
2540,3525	00D	PCH1	Card punch
2401,2402,2403,2415,2420, 3410,3411,3420,	181-4	TAP1-TAP4	Tape drives

[1]The device addresses shown are those that are preassembled into the CMS resident device table. You can change the virtual machine addresses by using the CP DEFINE command.
[2]The virtual address of the system console may be any valid multiplexer address.
[3]191 is the default user-accessed A disk unless it is dynamically changed by an ACCESS at CMS IPL.

Figure 12. Devices Supported by a CMS Virtual Machine

CMS Libraries

CMS updates simulated partitioned data sets which contain:

- CMS and OS macros to be used at assembly time (macro libraries)
- Object routines to be referred to at execution-load time (text libraries)
- Executable routines that are loaded by OS SVCs that CMS simulates. The CMS filetype for these files is LOADLIB.
- Executable routines that are fetched by DOS SVCs that CMS simulates. The CMS filetype for these files is DOSLIB.

System Macro Libraries

The system macro libraries, located on the CMS system disk, are:

Library	Contents
DMSSP MACLIB	CMS and DOS macro instructions versioned by VM/SP
CMSLIB MACLIB	CMS macro instructions not versioned by VM/SP
OSMACRO MACLIB	The selected OS macros from SYS1.MACLIB that are supported under CMS
OSMACRO1 MACLIB	The remaining distributed OS macros from SYS1.MACLIB
OSVSAM MACLIB	A subset of OS/VS VSAM macros that are supported under CMS
TSOMAC MACLIB	The OS macro instructions distributed in SYS1.TSOMAC
DOSMACRO MACLIB	Macro instructions used for CMS/DOS support

If you have previously created a CMS macro library and called it DOSMACRO MACLIB, you should rename it so that it does not conflict with the DOSMACRO MACLIB supplied with the system.

If you plan to assemble VSE programs containing VSE macro instructions in CMS/DOS, you must first create a CMS macro library that contains all the VSE macro instructions you need. The *VM/SP HPO Installation Guide* shows the procedure for copying an entire macro library. The procedure for copying individual macro instructions is described in the *VM/SP CMS User's Guide.*

If you plan to assemble VSE programs containing VSE/VSAM macros, you must first create a CMS MACLIB containing the VSE/VSAM macros. An EXEC named VSEVSAM is distributed with VM/SP that can be used in conjunction with the VSE/VSAM optional source distribution tape to create such a library. See the *VM/SP Installation Guide* for additional information on the VSEVSAM EXEC.

System Text Libraries

The system text libraries, located on the CMS system disk, are:

Library	Contents
CMSLIB TXTLIB	The CMS system text library
TSOLIB TXTLIB	Selected TSO routines necessary to support certain features of the language program products
EREPLIB TXTLIB	Base text library for CPEREP
ERPTFLIB TXTLIB	Updates to CPEREP text library

Note: EREPLIB and ERPTFLIB may have to be placed on one of the userid CE's minidisks.

Other Libraries

Also located on the CMS system disk is the CMSBAM DOSLIB, used to build the CMSBAM discontiguous saved segment used with CMS/DOS.

Execution-time libraries are available with the program product language processors.

You can generate your own libraries and add, delete, or list entries in them via the MACLIB and TXTLIB commands. You can also specify which libraries (system and user) to use for program compilation and execution via the GLOBAL command. Up to eight libraries may be specified. Although CMS library files are similar in function to OS partitioned data sets, OS macros should not be used to update them.

CMS Command Language

The CMS command language lets you converse with CMS. With this command language, you can use:

- Language compilers
- An assembler
- CMS file management system
- Context editing and line editing
- Execution control
- Debugging capability
- Generalized HELP facility

Additionally, you can invoke the CP commands available to all virtual machines under VM/SP HPO directly from CMS. Using these CP commands, you can send messages to the operator or to other users,

dynamically change your virtual machine's configuration, and invoke spooling facilities. In CMS, the facilities of CP and CMS together appear as those of a single integrated system.

To use CMS, you must first gain access to a virtual machine via the CP LOGON command, and IPL CMS. Then you can enter commands or data from the remote terminal (virtual operator's console). Each command, upon completion, returns control to you. For information about how to use CMS and for a description of all CMS commands, see the *VM/SP CMS Command and Macro Reference* and the *VM/SP CMS User's Guide.*

CMS Program Language Facilities

The language processors available under CMS include:

S/370 Assembler
VS BASIC
PL/I
OS FORTRAN IV
VS FORTRAN Extended
OS/VS COBOL
DOS PL/I Optimizer
DOS/VS COBOL
VS APL
DOS/VS RPG II

The assembler is distributed with VM/SP. The language compilers that are program products must be ordered separately. For a complete list of language processors that can be executed under CMS, see Appendix A, "Program Products, Installed User Programs, Field-Developed Programs, and Emulators" on page 383.

CMS executes the compilers via interface modules. Users should always recompile their programs and compiler interfaces under the system they are running on to insure any interface changes are incorporated (i.e., control block changes). CMS commands are provided to invoke the compilers within the conversational environment of CMS.

OS/VS COBOL programs, using the following facilities, can be compiled under CMS, but must be transferred to a machine (virtual or real) running OS for execution.

QSAM & BDAM spanned records
ISAM
RERUN statement
label-handling options
OPEN REVERSED
Sort feature
Segmentation feature
ASCII code feature
Forced end of volume
TCAM feature

OS PL/I programs, using the following facilities, can be compiled under CMS, but must be transferred to a machine (virtual or real) running OS for execution.

Multitasking
Teleprocessing file support
ISAM
Backwards attribute
Spanned records for buffered files
Sort-merge
Checkpoint-restart
ASCII data sets
Track overflow

The DOS/VS COBOL, DOS PL/I, and DOS/VS RPG II Optimizing compilers execute in the CMS/DOS environment of CMS. The CMS/DOS environment does not support the execution of DOS programs that use:

- Teleprocessing, indexed sequential access method (ISAM), or direct access method (DAM). CMS/DOS supports only the sequential access method (SAM) and virtual storage access method (VSAM).
- Multitasking. CMS/DOS supports only a single partition, the background partition.

Limited Support of OS and VSE in CMS

Object programs (TEXT files) produced under CMS or OS can be executed under CMS provided they use OS functions simulated by CMS. Object programs using nonsimulated OS macro functions must be transferred to an appropriate real or virtual OS machine for execution.

Sequential and partitioned data sets residing on OS disks can be read by OS programs running under CMS. Also, certain CMS commands can be used to process data sets on OS disks.

CMS does not support multibuffering for non-DASD devices. There is one DCB per device, not per file.

CMS simulates the control blocks, supervisor and I/O macro instructions, linkage editor and fetch routines necessary to compile, test, and execute VSE programs under CMS. The support for the DOS user is comparable to that for the OS user.

CMS supports VSAM and Access Method Services for DOS and OS users. Refer to the VM/SP CMS Command and Macro Reference for the restrictions on using VSE/VSAM and Access Method Services in CMS. CMS supports VSAM for:

- OS/VS COBOL
- OS PL/I
- VS BASIC
- VS/APL
- DOS/VS COBOL
- DOS/VS RPG II
- DOS PL/I
- DOS/VS SORT/MERGE

CMS does not support VSAM for assembler language programs.

CMS/DOS support of VSAM is based on the VSE/VSAM program product. Installations that plan to use this facility must order the VSE/VSAM program product.

The application programmer who normally uses CMS to interactively create, modify, and test programs may require facilities not supported in CMS (for example, an OS program using ISAM). He can alternately execute CMS and another operating system in the same virtual machine.

A description of the actual processes for reading OS or DOS files is in the *VM/SP CMS User's Guide.* A description of alternating operating systems is in *VM/SP HPO Operating Systems in a Virtual Machine.*

DL/I in the CMS/DOS Environment

Batch DL/I application programs can be written and tested in the CMS/DOS environment. This includes all batch application programs written in:

- COBOL
- PL/I
- RPGII
- Assembler language

You can execute any data base description generation and program specification block generation. The data base recovery and reorganization utilities must be executed in a VSE virtual machine.

For more information, see the *VM/SP CMS User's Guide*, and *DL/I DOS/VS General Information*, GH20-1246.

CMS Disk and File Management

CMS can manage up to twenty-six virtual disks for each user. These disks may be minidisks or full packs. Moreover, they may be in:

- CMS format
- OS or DOS format
- VSAM format

When the Mass Storage System is installed, and the processor is attached to an MSS, any CMS virtual disk can be located on an MSS 3330V volume.

CMS disks are formatted with the CMS FORMAT command. Files contained on these disks are in a format unique to CMS, and cannot be read or written using other operating systems.

OS and DOS disks or minidisks may be used in CMS. OS or DOS programs executing in CMS may read data sets or files on OS or DOS disks, but may not write or update them. OS and DOS minidisks may be formatted with the Device Support Facility or with an appropriate OS/VS or VSE disk initialization program, if the disk is a full pack.

Minidisks for use with VSAM must be formatted with the Device Support Facility. Full disks must be initialized using the appropriate OS/VS, Device Support Facilities, or VSE disk initialization program.

Disk Access

Disks can be accessed so that files can only be read (read-only), or so that files can be read and written (read/write).

Both CP and CMS can control read/write access. If a disk is designated read/write by CP, then CMS determines if the access is read or write. If a disk is designated read-only by CP, then it can only be read by CMS.

To access a disk, you must:

- Identify the disk to CP as part of your virtual machine configuration. This disk is available if it is defined in your directory entry, or it can be acquired dynamically with the CP LINK or DEFINE commands.

- Identify the disk to CMS by assigning it a filemode letter. You do this using the ACCESS command in CMS.

You may have many virtual disks known to CP in your virtual machine configuration at one time. CMS allows a maximum of twenty-six to be accessed, with filemode letters A through Z. The S-disk (usually at virtual address 190) is the CMS system disk. The A-disk (usually at virtual address 191) is the user's primary read/write work disk. Disks may be dynamically accessed and released during a terminal session.

File Sharing

CP provides for sharing of disks and minidisks among several users. The type of access (multiple users read-only or read/write) is controlled by LINK command operands. Password protection is provided. Since CMS does not provide any control for multiple writes (such as ENQ, DEQ), it is not recommended that CMS disks be used with multiple-write access.

CMS Disk File Format

All the disks that are to contain CMS files must be formatted before being used the first time. The CMS FORMAT command initializes disks in CMS format and writes a label on the disk.

A disk can be formatted into one of five disk block sizes:

512, 800, 1024, 2048, or 4096 bytes.

Count-key-data devices use an 800 byte format to provide compatibility with earlier releases. The volume label is written on record 3 of cylinder 0, track 0 for count-key-data devices, and on block one (origin zero) for FB-512 devices. The volume label contents depends on the formatting block size as detailed in Figure 13.

Field Description	Byte Displacement Length	Contents by Disk Block Size	
		800 byte	1K, 2K, 4K byte
Label identifier	0,4	C'CMS='	C'CMS1'
Volid	4,6	User label	User label
Version identifier	10,2	X'0000'	X'0000'
Disk block size	12,4	Not used, zeros	F'1024', F'2048', or F'4096'
Disk origin pointer	16,4	Not used, zeros	F'4' or F'5'
Number of usable cylinders/blocks	20,4	Not used, zeros	F'n'
Maximum number of formatted cylinders/blocks	24,4	Not used, zeros	F'n'
Disk size in CMS blocks	28,4	Not used, zeros	F'n'
Number of CMS blocks in use	32,4	Not used, zeros	F'n'
FST size in bytes	36,4	Not used, zeros	F'n'
Number of FSTs per CMS block	40,4	Not used, zeros	F'n'
Disk FORMAT date	44,6	Not used, zeros	X'yymmddhhmmss'
Bytes 50-79	50,30	Not used, zeros	Not used, zeros

Figure 13. Volume Label Contents for CMS Formatted Disks

On count-key-data devices, each 512-byte, 800-byte, 1K, 2K, or 4K block (called CMS block in the following discussion) represents one physical record of that size on disk. For FB-512 devices, each FB-512 (512-byte) blocks, logically concatenated to form the correct number of data bytes for that CMS block. The 800-byte disk format is not supported for FB-512 devices.

Files placed on CMS disks can have logical records that are fixed or variable length. In either case, the CMS file system places these file records contiguously into the fixed length CMS blocks, spanning blocks where necessary. As a file grows or contracts, its space is expanded or reduced accordingly.

Files on a CMS disk are identified by means of a file directory. The file directory is updated when a command is issued that changes the status of the file on the disk.

For a minidisk formatted in 512, 1024, 2048, or 4096-byte CMS blocks, a single CMS file can contain up to approximately $(2^{3}+^{1}\text{-}1)$-132,000 disk blocks of data, grouped into as many as $2^{3}+^{1}\text{-}1$ logical records, all of which must be on the same minidisk.

There are more restrictions for a minidisk formatted in 800-byte physical blocks. A minidisk cannot contain more than 3500 files if it is allocated on a 2314/2319, and not more than 3400 files for *all* the other DASDs supported by CMS. A single file can contain up to 12,848,000 bytes of data only, grouped into as many as 65,535 logical records. The number of 800-byte CMS blocks is limited to 65,535 per minidisk. This results in a maximum usable minidisk size in terms of cylinders depending on the DASD type. Figure 14 compares the disk devices supported by CMS in the case of 800-byte CMS blocks.

Other Restrictions:		
Maximum number of files per minidisk (exceptions - 2314/2319)	other DASDs	3400 (3500)
Maximum number of logical records per file		65,535
Maximum number of data bytes per file		12,848,000 bytes or 16,060 CMS blocks
Maximum number of 800-byte CMS blocks per minidisk		65,535
Minidisk allocated on device type	**Number of 800-byte blocks per cylinder**	**Maximum usable minidisk size in cylinders**
2314/2319	150	203
3330	266	246
3340 model 35	96	348
3340 model 70	96	682
3350	570	115
3375	360	182
3380	540	121

Figure 14. CMS Disk File Statistics for 800-Byte CMS Blocks

Identifying Disk Files

CMS commands can list the identifications of files on CMS and non-CMS formatted disks and minidisks. The LISTFILE command lists the entries in the master file directory for CMS disks. The LISTDS command lists the entries in the VTOC (volume table of contents) for OS and DOS disks, or data spaces on VSAM volumes.

CMS Tape Support

Each CMS machine can support up to sixteen magnetic tape units at virtual addresses 181 through 187 and 288 through 28F. They may be 2401, 2402, 2403, 2415, 2420, 3410/3411, 3420, or 8809 drives.

For 7-track and 9-track tape subsystems, the tape-handling commands ASSGN, FILEDEF, and TAPE allow you to specify the modeset of the tape: track and density. For 7-track subsystems only, you can specify the tape recording technique (odd or even parity, converter on or off, and translator on or off).

If you do not specify the modeset for a 7-track tape, CMS issues a modeset indicating 7-track, 800 bpi (bits per inch), odd parity, converter on, and translate off. If the tape is 9-track, the density is assumed to be 1600 bpi (or whatever bpi the tape drive was last set at) for dual density drives; for single density drives, the featured density (800, 1600, 6250 bpi) is assumed.

As an alternative to specifying mode in each command that uses the tape, e.g. FILEDEF, you can issue a CMS TAPE command. For example:

```
TAPE MODESET (181 9TRACK DEN 6250
```

TAPE MODESET sets the mode for the tape, which stays in effect until the command is reissued. You must do this if one of your programs is to use tapes in other than the default mode. The tape must be at load point for the MODESET to take effect.

Read the section "Tape Labels in CMS" in the *VM/SP CMS User's Guide* carefully before using labeled tapes in CMS. The CMS tape label processing features described there allow you to specify tape files with IBM standard or non-standard labels, or to bypass label processing for non-labeled tapes.

Multivolume tape files are not supported by CMS.

Note: These restrictions apply only when you run CMS. DOS and OS systems running in virtual machines can continue to read and write tapes with standard labels, non-standard labels, or no labels on single and multi-reel tape files.

The system operator must attach tape drives to your CMS virtual machine before any tape operation can take place.

For information about tape handling in the CMS/DOS environment, see Chapter 8, "Planning for CMS/DOS" on page 79.

CMS Unit Record Support

CMS supports:

- One virtual card reader at virtual address 00C
- One virtual card punch at virtual address 00D
- One virtual printer at virtual address 00E

Under CP, these devices are spooled. CMS does not support real or dedicated unit record devices, nor does it support a virtual 2520 Card Punch. Figure 12 lists the devices supported as virtual devices by CMS.

Saving CMS

CMS is designed so that it can be saved easily and shared by CMS users. Also, CMS is designed so that CMS/DOS, CMS VSAM and access method services, the System Product Editor, EXEC interpreter, EXEC2 interpreter, and CMS OS simulation routines can be placed in discontiguous saved segments that can be shared among users. The starter system has entries in the system name table (DMKSNT) and CP system control file (DMKSYS) so that you can save CMS (and the CMS discontiguous saved segments) at the end of the system generation procedure for CMS.

The default DMKSNT entry places the CMS nucleus from location X'1D0000' to X'200000'. It is intended that the CMS nucleus be shared at a location above the average virtual machine size for your installation. If a user's virtual machine size extends beyond the start location of the CMS nucleus, then the CMS nucleus will exist in the user's virtual storage, and the FREELOWE pointer (DMSFREE low-extended pointer) will be located at the start of the CMS nucleus. This may prevent the user from acquiring a large contiguous GETMAIN area. Because of this the CMS nucleus should be generated at a higher virtual address, or an alternate saved CMS system should be generated for such users. The default DMKSNT file supplied with your system includes an entry for such a system (CMSL), and a load list (CMSLOADL EXEC) is supplied with which you can generate a CMS nucleus at storage location X'F00000'.

The location of your saved CMS nucleus used by the majority of your users should not be excessively high, since CP constructs a segment table which has entries for all segments in the user's virtual machine, whether the segments exist or not. For example, if a user has a 512K virtual machine size and IPLs a CMS system located from X'1D0000' to X'200000', the user's segment table will have entries for all segments from X'0' to X'200000'. These segment tables are built in real storage by CP. An excessively high nucleus location will cause real storage to be wasted in the construction of

these segment tables. You may therefore wish to relocate the CMS nucleus, based on your installation's requirements, to a higher or lower address than the default. To do this, you should modify the load list for CMS, CMSLOAD EXEC, and modify the SLC entry in the list that immediately precedes DMSALP, and the entry preceding DMSOME to the address which you want for your CMS nucleus. You should then create SLC files, which will be included in the CMS nucleus load deck to change the loader location counter when the nucleus is loaded via IPL from your reader. These addresses should be chosen based upon your installation's needs.

An additional load list, CMSLOADL EXEC, is included with the CMS system. Using this load list, a CMS nucleus may be built a CMSL located at address X'F00000'. In addition, the sample DMKSNT files include, as a comment, the entry for a saved CMS system at this location. You may use this as an example of generating a CMS system at an alternate location.

For more information about saved systems, see Chapter 25, "Defining Saved Systems and Discontiguous Saved Segments" on page 373 and the *VM/SP HPO System Programmer's Guide.*

Chapter 5. Minidisks

Suppose each of many virtual machines had one real DASD for each configured virtual DASD. If they all ran at the same time, external storage needs would be excessive. To avoid this waste, the physical disk pack is divided into logical subsets, called minidisks. A unique sequence of whole cylinders or blocks is assigned to each minidisk as defined in the MDISK directory record. That allocated space is a complete virtual disk device. Each minidisk has its own virtual device address and virtual cylinders or blocks. To identify each minidisk to CP, there must be a volume table of contents or disk label identifier (VTOC) with data about each minidisk.

Minidisks are controlled and managed by CP. If a virtual machine attempts to use DASD space beyond the boundaries defined for its minidisks, CP presents a command reject (seek check) to the virtual machine. If a system is to be run on both a virtual and a real machine, minidisks for that system must start at real cylinder or block zero. For a detailed list of minidisk restrictions, see Appendix D, "VM/SP HPO Restrictions."

The remainder of this section describes how to:

- Define minidisks
- Allocate space
- Initialize minidisks
- Specify alternate tracks/blocks
- Label minidisks
- Share minidisks

Defining Minidisks

The VM/SP HPO directory entry for a virtual machine defines each minidisk by means of an MDISK statement. Any minidisk so defined is a permanent part of the virtual machine configuration. You can access data on the minidisk whenever you log on.

Temporary need for direct access space requires using a pool of reserved disk space, called T-DISK space. You specify T-DISK space in an MDISK statement in the directory. You choose its size when you allocate disk space with the stand-alone Format/Allocate program. Minidisks created from the T-DISK area must be initialized when you log on and exist in the virtual machine for only that session. When the virtual machine logs off or issues a CP command to release the T-DISK, the area is returns to CP.

When you allocate minidisks on disk volumes, you must take care to reduce:

- The amount of motion forced on the read/write heads
- The degree of physical overlap for the arm.

To find out how to reduce these problems, read Chapter 23, "Preparing the CP System Control File (DMKSYS)" on page 311.

Note: The VM/SP HPO directory function neither checks nor flags overlapped or duplicate minidisk extents. Nor does the function provide DASD space records for unused or used space.

Figure 15 illustrates the use and definition of minidisks. The disk labeled OSDOS1 contains several minidisks, some formatted to OS requirements and others to VSE requirements. OSDOS1 is a 2314 volume. The directory entry for userid ABC, an OS user, describes:

- Virtual device 230 as a 50-cylinder disk area on real volume OSDOS1
- Virtual device 231 as a 20-cylinder disk area on real volume OSDOS1
- Virtual device 232 as a 30-cylinder disk area on real volume CPVOL1

The directory entry for userid XYZ, a VSE user, describes:

- Virtual device 130 as a 50-cylinder disk area on real volume OSDOS1
- Virtual device 131 as a 20-cylinder disk area on real volume CPVOL1.

Note: On a 3330, 3340, 3350, 3375, or 3380, an OS/VS or OS minidisk must start at real cylinder 0 unless the VTOC is limited to one track. "Minidisk Restrictions" on page 404 provides more detail.

Minidisk Space Allocation

A minidisk always begins at virtual cylinder or block zero. For count-key data (CKD) devices, its minimum size is one cylinder. An exception is a 2314 or 2319 disk formatted by the Device Support Facility. In that case, the minimum number of cylinders is two, and the second cylinder is used as the alternate track cylinder. Except for the 3350 in 3330-1 or 3330-11 compatibility mode or in native mode, a minidisk must exist on its real counterpart. For example, a virtual 3340 minidisk must reside on a real 3340. An FB-512 minidisk can be any number of blocks up to the maximum of the volume.

A DASD volume with multiple minidisks has some tracks where the cylinder address in the count fields of records R0 and R1 disagree. If an attempt is made to read this volume by IEHDASDR, you may get messages IEH813I or IEH869I. To prevent this, initialize the disk with the FORMAT function of IEHDASDR before using it. This function rewrites R0 and R1 on each track so that the count fields agree with each other.

CP controls the boundaries of minidisks. For a DASD address outside the boundaries specified in the MDISK directory statements, CP presents a command reject (seek check) I/O error to the virtual machine.

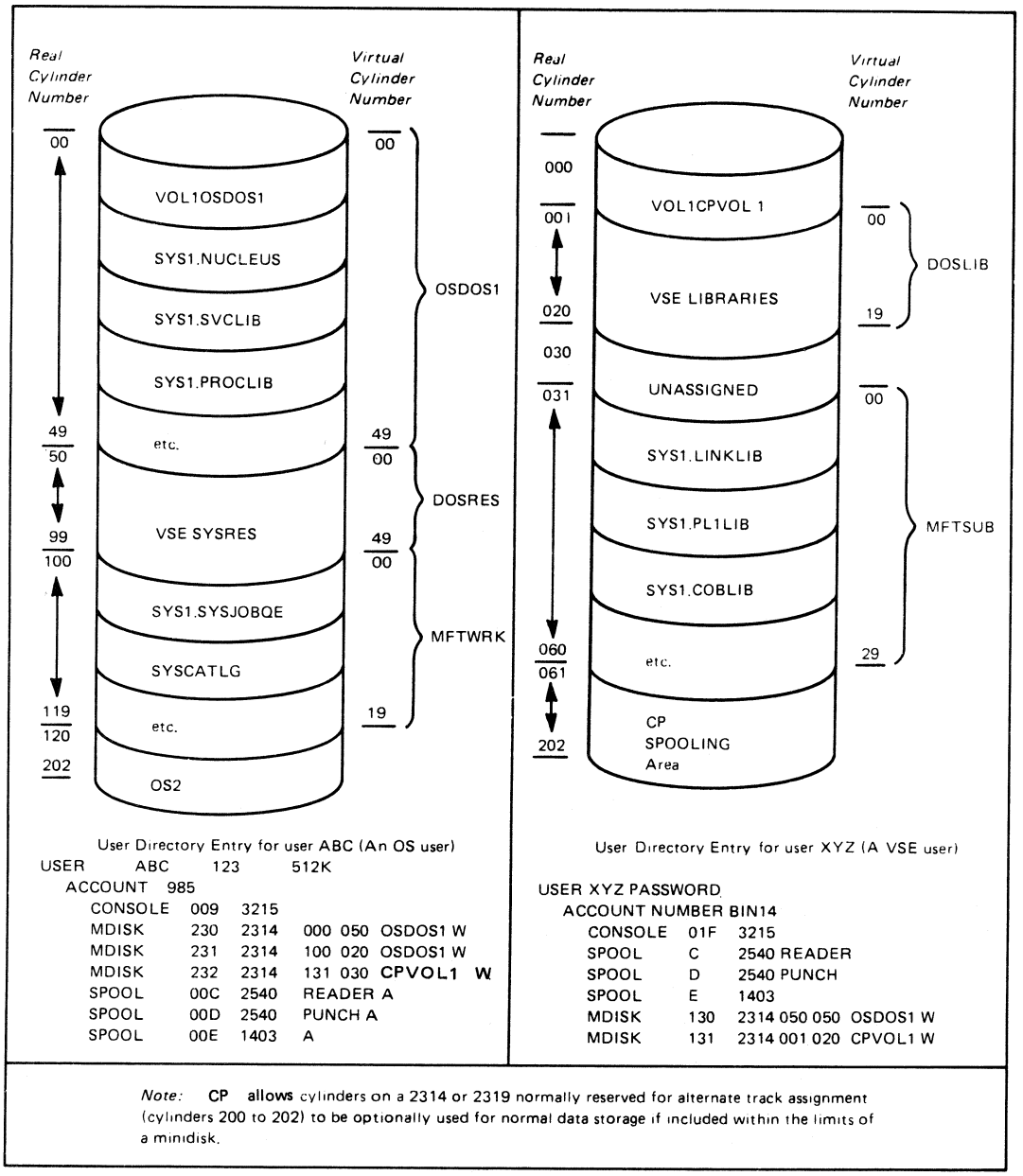


Figure 15. Use and Definition of Minidisks

Note: If the cylinder or block addresses in the MDISK statements inadvertently overlap each other, data in them may be compromised with no obvious error.

OS Minidisks

OS bases all of its space allocation parameters on the volume table of contents (VTOC) label written on each disk. The amount of space provided for that volume is determined from the format-5 (space accounting) data set control block (DSCB). Thus, for OS to support minidisks, a VTOC must be written whose format-5 DSCB reflects the desired minidisk size. OS treats the remaining disk space on the real disk as permanent dedicated space and cannot use it as format-5 DSCBs. The Device Support Facility service program should be used to format minidisks for use by OS or VSE. VSE does not support 3380.

VSE Minidisks

VSE allocates space as specified in the EXTENT job control card. It is your responsibility to see that the EXTENT cards refer to valid minidisk cylinders. On a 2314 or 2319 volume, the last minidisk cylinder for Device Support Facility is always reserved as an alternate track cylinder. Therefore, a VSE minidisk on a 2314 or 2319 must have at least two cylinders. For example, if you specify a ten-cylinder minidisk, the EXTENT card must refer to cylinders 0 through 8 only. This leaves the last cylinder for alternate track assignment. On a 3330, 3333, 3340, or 3350 minidisk, however, no cylinder is reserved for alternate tracks within each minidisk. Therefore, a ten-cylinder minidisk must be defined in the EXTENT card as cylinders 0 through 9.

MSS Minidisks

When MSS minidisks are defined on MSS 3330V volumes, minidisks are virtual 3330-1 disks. The presence of MSS and 3330V system volumes is transparent to a virtual machine accessing minidisks.

Minidisk Initialization

Like real disks, minidisks must be formatted for use by the appropriate service program.

You format directory-defined minidisks only once and temporary minidisks each time they are used. For CKD, you may use any program to initialize disks that supports the operating system's use of the DASD type. To initialize a minidisk, run one of the following service programs in a virtual machine:

Service Program	Disk Type	Formatting Information
CMS FORMAT command	CMS (except CMS/VSAM)	Formats the specified tracks into 512-, 800-, 1024-, 2048-, or 4096-byte blocks or physical records
CP Format/Allocate program	CP	Formats specified tracks into 4096-byte blocks
Device Support Facility	OS, VSE, and CMS/VSAM (on CKD devices)	Writes read-only track descriptor records for each track, and sets the remaining portion of each track to binary zeros. It also writes a format-5 DSCB whose contents reflect the minidisk size (the amount of free space available for allocation).

Alternate Tracks/Blocks

As a preventive action when disks are made, alternate tracks are assigned as replacements for defective primary tracks. Device Support Facility can assign these replacements in the field.

3330/3350 DASD

On the 3330 or 3350, the control unit automatically assigns alternate tracks. The 3830 control unit assigns primary cylinders:

- 0 through 403 as minidisks on the 3330 Model 1 or 2. The remaining cylinders (404 to 411) are assigned for alternate tracks.
- 0 through 807 as minidisks on the 3330 Model 11. The remaining cylinders (808 to 814) are assigned for alternate tracks.
- 0 through 554 as minidisks on the 3350. The remaining cylinders (555 to 559) are assigned for alternate tracks.

3340/3344 DASD

The 3340 DASD uses a hardware logic that lessens the dependence on alternate track usage. The 3340 can bypass the defective portion of a data track and write the balance of the record in the space remaining. When an alternate track is required, it can be assigned by the Device Support Facility. A dedicated 3340 device uses the 3348 Data Module to reserve cylinders:

- 0 through 347 as minidisks on Model 35. Cylinder 348 is reserved for alternate tracks.
- 0 through 695 as minidisks on Model 70. Cylinders 696 and 697 are reserved for alternate tracks.

The same values hold for 3344 devices as for the 3340 Model 70.

Once the Device Support Facility has assigned the alternate track, the disk may be used for any normal purpose. There are three restrictions:

- A minidisk should not be located where its track 0, cylinder 0 falls on a defective track of a 3340 or 3344 storage device. If it is, the CP IPL command will not work for that minidisk.
- Any operating system doing SIO to a disk with a defective 3340 or 3344 track must be able to do normal alternate track error recovery.

  Note: CMS qualifies here because it uses DIAGNOSE in place of SIO.

- VM/SP HPO does not perform alternate track recovery for overflow record processing on 3340 or 3344 storage devices. If a defective track is encountered during overflow record processing, a fatal I/O error results.

Error Recovery Support

When an I/O issued to a defective 3340 or 3344 track results in a track condition check, an alternate track is needed. Software error recovery procedures provide the means to switch to an alternate track.

- For CP I/O and for DIAGNOSE I/O issued from a virtual machine, switching is fully automatic and the issuer of the I/O request is not aware of it.
- For SIO issued from a virtual machine, the track condition check is sent to the virtual machine. This allows the operating system in the virtual machine to run its own error recovery procedures.

Since alternate tracks are assigned from the high-order cylinders at the end of the real 3340, the virtual machine will attempt to seek outside of the minidisk to recover. The CP CCW translation process allows seeks outside of the minidisk to an alternate track provided that the particular alternate track is assigned to a defective track within that minidisk. After the alternate track is assigned, any attempts at head switching to an unowned track in this cylinder are prevented.

Allocation Conversion

Previously, "alternate track" cylinders of 3340/3344 devices were often used as primary data cylinders. Now these cylinders must be reserved exclusively for alternate track use. Therefore, changing from an old system (prior to VM/370 Release 5 PLC 6) to a current system requires some revisions. You must revise space allocation and minidisk layouts on any 3340/3344 disk where primary data areas are on "alternate track" cylinders.

System Residence Devices: The system residence device containing "alternate track" cylinders formerly used as the primary data area needs revising. Files of existing control statements should be revised before you generate a new system. In particular, the allocate function performed on the system residence disk and other CP-owned disks may have to be revised. After this revision, review the specification of the SYSRES, NAMESYS, and NAMENCP macros.

Minidisk Devices: Any 3340/3344 minidisks extending into the alternate track cylinders need to be moved. Use the DASD dump restore (DDR) utility to copy them to another area of the disk or to another disk. In the past, when a 3340/3344 had a defective track, the cylinder with the bad track not usable. Minidisks would be allocated adjacent to that cylinder, but not including it. In such a case, use any version of the DDR utility to dump all cylinders of the real disk to tape.

Take care when revising alternate track cylinders which have been used as a primary data area with the new version of the DDR utility. Make sure that you specify the cylinder range exactly. For example, enter:

```
DUMP 0 TO 697
```

rather than specifying ALL. An exact specification dumps only tracks that have been assigned as alternates from the final cylinders. Next, run the Device Support Facility program to assign alternate tracks to the defective tracks so that all cylinders become usable. Then, use the new DDR utility to restore minidisks from the tape. It may be possible to reorder them into the previously unusable cylinders.

Note: Whenever the location or size of a minidisk is changed, you must revise the related MDISK entries in the system directory.

After alternate tracks have been assigned, use only the most current versions of the DDR, DIR, and FMT utilities with 3340/3344 devices.

2314/2319 DASD

On 2314 and 2319 devices, CP and CMS (except CMS/VSAM) do not recognize or support alternate track techniques for their own use. VSE, OS, and VMS/VSAM minidisks, however, do recognize and support alternate tracks on these types of DASD. The Device Support Facility automatically assigns the last cylinder in any minidisk as an alternate track cylinder. When you initialize 2314/2319 devices, you can assign all 203 cylinders for virtual machine and system use.

If a track assigned to a virtual machine minidisk area becomes defective:

- Run the standalone CP Format/Allocate service program if the minidisk is CP-owned. Flag the whole cylinder containing the defective track as permanent (PERM). This prevents CP from ever allocating that cylinder for CP paging, spooling, or temporary files. Remember to exclude this cylinder when you allocate disk space for any minidisk in the directory entry for a virtual machine.

- Reformat the minidisk, including the defective track, with the Device Support Facility program if the minidisk is used by either VSE, OS, or CMS/VSAM. An alternate track is assigned at the end of the minidisk.

- Set up the entire volume containing the defective track as an OS, VSE, or CMS/VSAM volume. Format it with either the Device Support Facility or IEHDASDR for OS or CMS/VSAM disks. For DOS disks, use the VSE Initialize Disk utility program (INTDK). Alternate tracks are assigned in the standard manner.

3375/3380 Disks

The control unit automatically handles defective tracks encountered on 3375/3380 DASD volumes. The control unit provides necessary hardware recovery for handling defective tracks. When a defective track is encountered, the hardware switches to an alternate track. If the end of the alternate track is reached, the hardware switches back to the first track following the defective track.

FB-512 DASD

When FB-512 disks are made, alternate blocks are flagged and replacements assigned for defective primary use. In the field they are assigned by the Format Defective Block procedure used with:

- Device Support Facility or

- LOCATE CCW as described in the hardware manual.

Given the number of the defective block, hardware replaces it with an alternate and sets up the appropriate pointers.

Labels

All disks to be handled by CP (as a whole or as a combination of logical disks) must have a label. The label must be on real cylinder 0, track 0, record 3 for count-key-data devices or on block 1 for FB-512 devices. It identifies the physical volume and must be in the form:

VOL1xxxxxx *or*
CMS=xxxxxx (for disks using an 800-byte format) *or*
CMS1=xxxxxx (for disks using a 1K, 2K, or 4K format)

where xxxxxx is a 6-character volume label. (VOL1xxxxxx is also the form of labels created by the Device Support Facility, IEHDASDR, or INTDK.)

In addition, all virtual machine minidisks should have a label at the same respective *virtual* locations for CKD or FB-512 devices.

A physical volume that holds only virtual machine minidisks can have the first of those minidisks starting at real cylinder or block 0. CP recognizes the physical volume if the first minidisk has a valid label.

In Figure 15, volume OSDOS1 has its real cylinder 0 allocated to a minidisk formatted for use by OS. The volume serial number of that minidisk must be OSDOS1, the label of the real volume. Changing the label affects the directory entries of all users who have minidisks on that volume.

Do not assign real cylinder or block 0 to a user as a data area. If that user has read/write access to the disk, he can destroy the label.

Also, do not assign user minidisks to begin with real cylinder or block 0 of any physical volumes to contain CP controlled areas (for paging, spooling, and so on). On these volumes, cylinder 0 track 0 record 4 contains control information required by CP. The VTOC labels for them are compatible with OS, but indicate to OS that there is no space on that DASD. If the space is assigned to a user minidisk, the programs that format OS, VSE, and CMS/VSAM minidisks write over it. The necessary control information is destroyed and CP system failures occur.

Sharing Minidisks

A minidisk can be shared by many virtual machines. One virtual machine is assigned as the owner. It has an MDISK control statement in its directory entry describing the minidisk. Other virtual machines can link to the minidisk in two ways:

- LINK control statement in their own VM/SP HPO directory entry
- CP LINK command with the proper password issued during a terminal session.

For example, assume a virtual machine called USERA owns a minidisk at address 150. The directory entry for USERA contains:

```
MDISK 150 3330 050 010 SYS003 W READPASS
```

USERA's virtual disk is on the volume labeled SYS003 and occupies real cylinders 050-059.

Any other virtual machine that issues the CP LINK command with the proper password can read the 150 minidisk belonging to USERA.

Any other virtual machine can also read USERA's 150 minidisk if it has the following control statement in its directory:

```
LINK USERA 150 cuu RR
```

The virtual device address cuu shows where the 150 minidisk belonging to USERA is linked to another virtual machine.

Assume another virtual machine, USERB, with the following statement in its directory entry:

```
LINK USERA 150 151 RR
```

USERB can read data from USERA's 150 virtual disk whenever it issues a read for data on its own 151 virtual disk.

You can link to any minidisk defined in the directory if both of the following conditions are met:

1. The minidisk being linked to has a password specified in the MDISK control statement. The password must represent the type of link you desire.

2. The type of access requested (R, RR, W, etc.) is feasible at the time of the link.

Note: *See the description of the CP LINK command in the* VM/SP HPO CP Command Reference for General Users *for more information about linking to minidisks.*

Chapter 6. Planning and Requirements for IPCS

This chapter identifies the effect of the VM/SP Interactive Problem Control System (IPCS) Component on the following:

- Machine requirements
- Programming requirements
- Conversion considerations
- Storage estimates

Machine Requirements

VM/SP IPCS operates on any IBM system that meets the minimum requirements for VM/SP Release 4 or VM/SP HPO Release 4.0.

VM/SP IPCS runs in a CMS virtual machine with a minimum virtual storage size of 512K (K equals 1024) and uses any terminal supported by CP as a logon device.

If the output device is a display terminal, the output of DUMPSCAN is written in blocks of twenty-two 80-character lines. The two 80-character lines at the bottom of the display are used as a user input area of 136 characters and a screen status display area of 24 characters. Therefore, for a proper display, the user's terminal must be capable of accepting this format.

Programming Requirements

VM/SP IPCS is designed for use with the VMDUMP facility. IPCS operates under the CMS component of VM/SP Release 4.

Conversion Considerations

The format of various internal files and data areas as well as internal interfaces for reading and writing dumps, creating problem reports, etc., have been changed for the VM/SP IPCS. Therefore, IPCS cannot view dumps or update symptom summary status for dumps taken by the IPCS supplied with the VM/370 system (Component 5749-DMM), and vice-versa.

If your installation is using the VM/370 system IPCS, or the VM/IPCS Extension, and plans to upgrade to VM/SP IPCS, the system IPCS symptom summary file and PRBnnnnn dumps can be converted to the format required by VM/SP IPCS by the use of the CONVERT command. Any existing system IPCS problem report files need not be converted for use with the PRB, PROB, or APAR commands of the VM/SP IPCS. The format of the problem report files varies from the VM/370 IPCS to the VM/SP IPCS, but the data is accurate.

If your installation has existing PVM Release 2 or RSCS Release 3 IPCSE, these help files must also be converted to the proper format for IPCS use. The CONVIPCS EXEC will aid you in this conversion.

For details on these conversion utilities, see the *Interactive Problem Control System Guide.*

IPCS can diagnose dumps from systems either with or without the Enhanced Subsystem support.

Storage Requirements

External Storage

The disk storage needed by VM/SP IPCS can be divided into two parts. The first part, which contains the current IPCS map, problem report files, and the symptom summary file, varies in size only slightly as the number of problem reports and symptom summaries vary. For example, for CP dumps, the following amount of data can be stored in 4 cylinders on a 3330:

- 75% (3 cylinders) = 100 problem reports plus symptom summary
- 25% (1 cylinder) = CPIPCS MAP

The second part of disk storage contains the dumps and supplemental data (for example, console files, trace output). The size of the dump depends mainly on the size of the system being dumped, and the operand of the CP SET DUMP command, either ALL or CP. Figure 16 on page 69 shows approximate space usage by device type or the for the fixed area and for one dump.

Files	Cylinders			
	3330	2319	3340	3350
Fixed Area				
IPCS CPNUC MAP 100 problem reports and symptom summary	4	8	10	2
One Dump				
512K CP	1.5	3	4	1
512K ALL	2.5	5	7	1.5
1024K CP	3	6	8	1.5
1024K ALL	6	11	16	3

Figure 16. Approximate Space Usage for Fixed Area and One Dump by Device Type

Calculating DASD Space

The amount of DASD space on the CMS file system required for a processed virtual machine dump can be calculated from the following formula:

$$C = \frac{S/4 + 5 + R}{P}$$

where:

C = Number of cylinders required. If there is a remainder, C must be rounded up.

S = Virtual machine storage size in K-bytes.

R = Number of pages required for appended load map. R is the number of 4096-byte records occupied by the output of the MAP command of IPCS.

P = Number of pages per cylinder.

For:

2305/3340, P = 24

2314/2319, P = 32

3330/3333, P = 57

3350, P = 120

3375, P = 96

3380, P = 150

Figure 17 is a table showing the number of cylinders needed to contain a dump from various sizes of virtual machines. The table assumes that the entire virtual storage is included in the dump, and a five-page load map is appended. In the table, K equals 1024 and M equals 1024K (1,048,576).

Virtual Machine Storage Size	2314/2319 cyls.req.	3330/3333 cyls.req.	2305/3340 cyls.req.	3350 cyls.req.	3375 cyls.req.	3380 cyls.req.
64K	1	1	2	1	1	1
128K	2	1	2	1	1	1
256K	3	2	4	1	1	1
512K	5	3	6	2	2	1
768K	7	4	9	2	3	2
1M	9	5	12	3	3	2
2M	17	10	22	5	6	4
4M	33	19	44	9	11	7
8M	65	37	86	18	22	14
12M	97	55	129	26	33	21

Figure 17. Cylinders Required to Contain a Dump Depending on Virtual Storage Size

Chapter 7. Planning for CMS VSAM and Access Method Services

CMS supports interactive program development for OS and VSE programs using VSAM.

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.

CMS also supports access method services to manipulate OS, DOS VSAM, and SAM data sets, and VSAM for use with DOS/VS SORT/MERGE.

Under CMS, VSAM data sets can span up to 25 DASD volumes. CMS does not support VSAM data set sharing, but does support the sharing of minidisks or full pack minidisks. Only one user may have write access to the VSAM master catalog, but many other users may concurrently read and reference the catalog.

VSAM data sets created in CMS are not in the CMS file format. Therefore, CMS commands currently used to manipulate CMS files cannot be used for VSAM data sets that are read or written in CMS.

Because VSAM data sets in CMS are not a part of the CMS file system, CMS file size, record length, and minidisk size restrictions do not apply. The VSAM data sets are manipulated with access method services programs executed under CMS, instead of with the CMS file system commands. Also, all VSAM minidisks and full packs used in CMS must be initialized with the Device Support Facility program or an appropriate VSE or OS/VS disk initialization program (if the minidisk is a full pack). The CMS FORMAT command must not be used.

In its support of VSAM data sets, CMS uses RPS (rotational position sensing) wherever possible. CMS does not use RPS for 2314/2319 devices, or for 3340 devices that do not have the feature.

Hardware Devices Supported by VSE

CMS support of VSAM data sets is based on VSE/VSAM. Disks supported by VSE can be used for VSAM data sets in CMS or for CMS disk files used as input by Access Method Services. These disks are:

- IBM 2314 Direct Access Storage Facility
- IBM 2319 Disk Storage
- IBM 3310 Direct Access Storage
- IBM 3330 Disk Storage, Models 1 and 2

- IBM 3330 Disk Storage, Model 11
- IBM 3340 Direct Access Storage Facility
- IBM 3344 Direct Access Storage
- IBM 3350 Direct Access Storage
- IBM 3370 Direct Access Storage, Models A1, A2, B1, and B2
- IBM 3375 Direct Access Storage

Note: CMS/VSAM also 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.

When the processor is attached to an MSS, the CMS disk may be defined as a 3330 Model 1 that is mapped by CP to all or part of a 3330V volume.

CMS disk files used as input to or output from Access Method Services may reside on any disk supported by CMS.

Programming Languages Supported

CMS supports VSAM for programs written for the following compilers:

Compiler	Program No.
OS/VS COBOL Compiler and Library	5740-CB1
OS COBOL Interactive Debug	5734-CB4
VS Basic Processor	5748-XX1
OS PL/I Optimizing Compiler and Libraries	5734-PL3
OS PL/I Checkout Compiler	5734-PL2
DOS/VS COBOL Compiler and Library	5746-CB1
DOS PL/I Optimizing Compiler and Library	5736-PL3
DOS/VS RPG II Compiler	5746-RG1
VS/APL	5748-AP1

Data Set Compatibility Considerations

CMS can read and update VSAM data sets created under VSE/VSAM or OS/VS. VSAM data sets with physical record sizes .5K, 1K, 2K, or 4K created under CMS can be read and updated by OS/VS VSAM. For complete information regarding VSE/VSAM and OS/VS VSAM, consult the *VSE/VSAM General Information Manual.*

If you perform allocation on a minidisk in CMS, you cannot use that minidisk in an OS virtual machine in any manner that causes further allocation. VSE/VSAM (and thus CMS) ignores the format-5, free space DSCB on VSAM disks when it allocates extents. If allocation later occurs

in an OS machine, OS attempts to create an accurate format-5 DSCB. The format-5 DSCB created by OS, however, does not correctly reflect the free space on the minidisk because OS expects it to be a full pack. In CMS, allocation occurs whenever data spaces or unique data sets are defined, and space is released whenever data spaces, catalogs, and unique data sets are erased.

ISAM Interface Program (IIP)

CMS does not support the VSAM ISAM Interface Program (IIP). Thus, any program that creates and accesses ISAM (indexed sequential access method) data sets cannot be used to access VSAM key sequential data sets.

There is one exception to this restriction. If:

1. You have OS PL/I programs that have files declared as ENV(INDEXED)

   and

2. The library routines detect that the data set being accessed is a VSAM data set

your programs will execute VSAM I/O requests.

Planning Considerations for Installing VSAM Under CMS

CMS support of VSAM and Access Method Services is based on the VSE/VSAM program product. You must order the supported level of the VSE/VSAM program and use the VSAMGEN EXEC to install VSAM under CMS.

Support of VSAM under CMS also requires that the CMSDOS and CMSBAM discontiguous saved segments be generated. For complete information on the installation of VSAM under CMS, see the *VM/SP HPO Installation Guide.*

Chapter 8. Planning for CMS/DOS

Those of you who use CMS/DOS must, in certain situations, have available a VSE SYSRES. If you wish to use either the DOS/VS COBOL or PL1 compilers under CMS/DOS, you must first order and install a VSE system (most current level) and install the compilers on this system.

If you plan to use CMS/DOS, you must also generate the CMSDOS and CMSBAM discontiguous saved segments. These segments contain simulated VSE services that are necessary for execution of VSAM and other VSE programs under CMS. Running VSAM under CMS depends on the generation of CMSDOS and CMSBAM.

For complete details on the installation of the CMSDOS and CMSBAM discontiguous saved segments, see "Loading and Saving Discontiguous Saved Segments" in the *VM/SP HPO Installation Guide.*

VSE System Generation Considerations

CMS/DOS support in CMS uses a real VSE system disk in read-only mode. CMS/DOS provides the necessary interface, and then fetches VSE logical transients and system routines directly from the VSE system libraries. Also, CMS/DOS fetches the DOS/VS COBOL and DOS PL/I compilers directly from the VSE system or private core image libraries.

It is your responsibility to order the most current VSE system and then generate it. Also, if you plan to use VSE compilers, you must order the DOS/VS COBOL and DOS PL/I optimizing compilers and install them on this VSE system.

When you install the compilers on the VSE system, you must link-edit all the compiler relocatable modules using the linkage editor control statement:

```
ACTION REL
```

You can place link-edited phases in either the system or the private core image library.

When you later invoke compilers from CMS/DOS, the library (system or private) containing the compiler phases must be identified for CMS. You identify all system libraries to CMS using the filemode letter that corresponds to that VSE system disk. Do this by specifying the filemode letter on the SET DOS ON command when you invoke the CMS/DOS environment. You identify a private library by coding ASSGN and DLBL commands that describe it. These VSE system and private disks must be linked to your virtual machine and accessed before you issue the commands to identify them for CMS.

CMS/DOS has no effect on the update procedures for VSE, DOS/VS COBOL, DOS/VS RPGII, or DOS PL/I. You should follow the normal update procedure for applying IBM-distributed coding changes to them.

When the VSE System Must Be Online

Much of what you do in the CMS/DOS environment requires that the VSE system pack and/or the VSE private libraries be available to CMS/DOS. In general, you need these VSE volumes whenever:

- You use the DOS/VS COBOL or DOS PL/I compilers. These compilers are executed from the system or private core image libraries.
- Your DOS/VS COBOL or DOS PL/I source programs contain COPY, LIBRARY, %INCLUDE, or CBL statements. These statements copy books from your system or private source statement library. This function requires that the CMSBAM shared segment be generated and available to CMS/DOS.
- You invoke one of the librarian programs: DSERV, RSERV, SSERV, PSERV, or ESERV.
- You link-edit VSE programs that use non-disk LIOCS modules. CMS/DOS link-edits LIOCS routines with the VSE program from VSE system or private relocatable libraries.
- You execute VSE programs that fetch phases directly from VSE system or private core-image libraries.

A VSE system pack is usable when it is:

- Defined for your virtual machine
- Accessed
- Specified, by mode letter, on the SET DOS ON command

A VSE private library is usable when it is:

- Defined for your virtual machine
- Accessed
- Identified via ASSGN and DLBL commands

The VSE system pack and private libraries may reside on full packs or minidisks.

CMS/DOS Tape Handling

You can use the CMS tape label processing features described in the *VM/SP CMS User's Guide* to process tapes defined with a DTFMT. The features described there allow you to define input and output tapes that have standard or non-standard labels and to define unlabeled tapes. They also allow you to specify your own exits for processing user standard or non-standard labels. Before CMS prepares your tape files for processing, it returns control to the tape label processing routines.

The CMS LABELDEF command, which is described in the *VM/SP CMS User's Guide*, is equivalent to the VSE TLB control statement for standard label tapes.

When a tape is defined as a work file, it is treated as unlabeled. Any labels encountered on the tape are written over.

Tape labels are not supported on tape files defined with DTFCP or DTFDI. Existing IBM standard header labels are bypassed on such tapes when they are used for input. Any existing labels are written over when the tapes are used for output.

Note: To use tape label processing you must have the CMSSEG shared segment available. Otherwise, the modules which belong in CMSSEG must be loaded into the user storage area.

Label Information Area

CMS/DOS does not support a label information area. If the real VSE system pack used by CMS/DOS has a label information area, it is not used.

In CMS/DOS, ASSGN and DLBL commands provide functions similar to those provided by the VSE ASSGN, DLBL, and EXTENT control statements. In VSE those control statements are in effect only for one job. Thus, it is convenient to place often-used DLBL and EXTENT control statements on the label information area.

In CMS/DOS, however, there is no such thing as a job. Consequently, ASSGN and DLBL commands remain in effect for an entire CMS/DOS session, unless they are reset by another ASSGN or DLBL command. Also, in CMS, you can place all the commands you need to compile and execute a program in an EXEC file and invoke that EXEC file by its filename.

Chapter 9. Planning for Virtual Machine Operating Systems (Other Than CMS)

This section contains information about:

- Operating systems that can run in virtual machines
- VM/VS handshaking feature
- Multiple virtual machines using the same operating system
- Remote Spooling Communications Subsystem (RSCS)
- IX/370 handshaking feature

Operating Systems that Execute in Virtual Machines

Processors use storage keys to protect information in real storage from unauthorized use. A storage key contains a four-bit control field in storage that is associated with either 2K or 4K blocks of real storage. With 2K key support, one storage key protects a 2K block of real storage and two keys protect a 4K block of storage. With 4K key support, one storage key protects a 4K block of real storage. On processors using 4K key support, the appropriate level of the virtual machine operating system is required.

2K Key Operating Systems

Batch or single user interactive:

DOS
DOS/VS
DOS/VSE
DOS/VS Advanced Functions
VSE/Advanced Functions
OS/PCP
OS/MFT
OS/VS1
OS/MVT
OS/VS1 BPE
OS/VS2 SVS
OS/VS2 MVS
MVS/SP
OS-ASP
RSCS

Multiple-Access:

VM/370
VM/System Extensions
VM/SP
VM/SP HPO
Time Sharing Option of OS (TSO)
VSE with VSE/ICCF

Conversational:

CMS

4K Key Operating Systems

Batch or single user interactive:

VSE/Advanced Functions Release 3.1
OS/VS1 Basic Programming Extensions Release 3
MVS/System Product Release 1 Enhancement

Multiple-Access:

VM/SP Release 2 and subsequent releases
VM/SP HPO2 and subsequent releases

The VM/VS Handshaking Feature

The VM/VS handshaking feature is a communication path between CP and certain other system control programs (such as OS/VS1) that makes each system control program aware of certain capabilities and requirements of the other. The VM/VS handshaking feature consists of:

- Processing pseudo-page-faults
- Closing CP spool files when the system control program's output writer operation is complete
- Providing an optional nonpaging mode for operating systems executing under the control of CP
- Providing miscellaneous enhancements for an operating system's virtual machine executing under the control of CP.

A page fault is a program interruption that occurs when an instruction in an active page refers to a page marked "not in storage". The virtual machine requesting that page is placed in the wait state while the requested page is brought into real storage. With the VM/VS handshaking feature, however, a multiprogramming operating system, executing under the control of CP in a virtual machine, may dispatch one task while waiting for CP to honor a page request for another task. When the pseudo-page-fault portion of VM/VS handshaking is active, CP sends a pseudo-page-fault

(program interrupt X'14') to the guest system. When the guest system recognizes a pseudo-page-fault, it places only the task requiring the page in wait status. It then can dispatch any of its other tasks.

Since no paging is done by the operating system using VM/VS handshaking, ISAM programs are treated by CP as if they are being run from fixed storage locations. In order to execute ISAM programs successfully, the virtual machine directory must therefore include the ISAM option.

When the handshaking feature is active, the operating system using VM/VS handshaking closes the CP spool files by invoking the CP CLOSE command when a task or job has completed. Once these spool files are closed, CP can process them without operator intervention.

Operating systems using VM/VS handshaking can execute in nonpaging mode. Nonpaging mode exists when

1. The handshaking feature is active,

   and

2. The operating system's virtual storage size equals the virtual storage size of the virtual machine.

When the guest operating system runs in nonpaging mode, fewer privileged instructions are executed and duplicate paging is eliminated. Such a virtual machine may have a larger working set when it is in nonpaging mode rather than when it is in paging mode.

Also, there are some other aids for guest systems using VM/VS handshaking while executing under the control of CP. With the handshaking feature, the guest system avoids some of the instructions and procedures that would be inefficient in the virtual machine environment.

When the VM/VS handshaking feature is active, the operation of a system control program closely resembles the stand-alone operation because much redundancy of function between CP and the operating system is eliminated. For instance:

- One VS1 task can be dispatched while another is waiting for a page to be brought into real storage.

- There is less need for the virtual machine operator to intervene because output files are automatically closed and processed.

Note: Even when the handshaking feature is active for a virtual machine, the pseudo-page-fault portion of the handshaking feature is not available until it is set on with the CP SET PAGEX ON command. This command can set pseudo-page-fault handling on and off.

Refer to *VM/SP HPO Operating Systems in a Virtual Machine* and *VM/SP as a Host System*, GC19-6212, for detailed information on handshaking.

Multiple Virtual Machines Using the Same Operating System

In general, an operating system which is to run in a virtual machine should have as few options generated as possible. This is also true when several virtual machines share a system residence volume. Very often, options that improve performance on a real machine have no effect (or possibly a negative effect) in a virtual machine. For example, seek separation, which improves performance on the real machine, is not needed in a virtual machine. CP itself issues a stand alone seek for all count-key-data disk I/O.

Sharing the system residence volume makes it unnecessary to keep more than one copy of the operating system online. The shared system residence volume should be read-only so that it can be shared among virtual machines. CMS discontiguous saved segments can also be shared among all virtual machines, since they are outside the virtual storage of each of the sharing virtual machines. The CMS/DOS environment of CMS simulates VSE/AF supervisor and input/output functions, thus allowing execution of many DOS programs. DOS and OS systems can be shared among users if all data sets with write access are removed from the system residence volume. Refer to the *VM/SP HPO System Programmer's Guide* for more details.

Remote Spooling Communications Subsystem

The RSCS Virtual Machine

The Remote Spooling Communications Subsystem (RSCS), operating in a virtual machine, handles the transmission of files between VM/SP HPO users and remote terminals and stations. Figure 18 illustrates a typical RSCS configuration.

Three lines of the real 3705, operating in 2703 emulation mode, are shown dedicated to the RSCS virtual machine. The communication lines are shown attached to a 3780 Data Communication Terminal, a System/3, and an OS/HASP processor running in a remote System/360 or System/370.

The RSCS machine uses the spooling facilities of CP to communicate with virtual users. Any user who wishes to have an output file transmitted to a remote location must associate tag information (such as destination and priority) with his file via the TAG command and spool the file to the RSCS machine's virtual reader. RSCS analyzes the tag data, enables the appropriate line, and transmits the data using the line protocol required by the receiving station.

Remote locations can submit card files to the RSCS machine and address them to either a VM/SP HPO user or to RSCS itself for transmission to another remote station. RSCS produces a CP spool file by writing that data to virtual unit record devices and, if the file is destined for a VM/SP HPO user, sends it to the user's virtual reader via the CP SPOOL command. If

the file is addressed to RSCS, it is queued on RSCS's virtual reader and then handled in the same manner as a file spooled to RSCS by a VM/SP HPO user. In this case, it is the responsibility of the remote station that originated the data to supply the tag information.

RSCS can also function as a remote workstation of a HASP/ASP type batch processor. VM/SP HPO users and remote stations can submit jobs to RSCS for transmission to the HASP system. After processing, HASP can return printed and/or punched output to RSCS for spooling to the real system printer or punch. You can expand the capabilities of RSCS in a VM/SP HPO system by installing the RSCS Networking program product (5748-XP1). For more information about RSCS Networking, see the *RSCS Networking Program Reference and Operations Guide.*

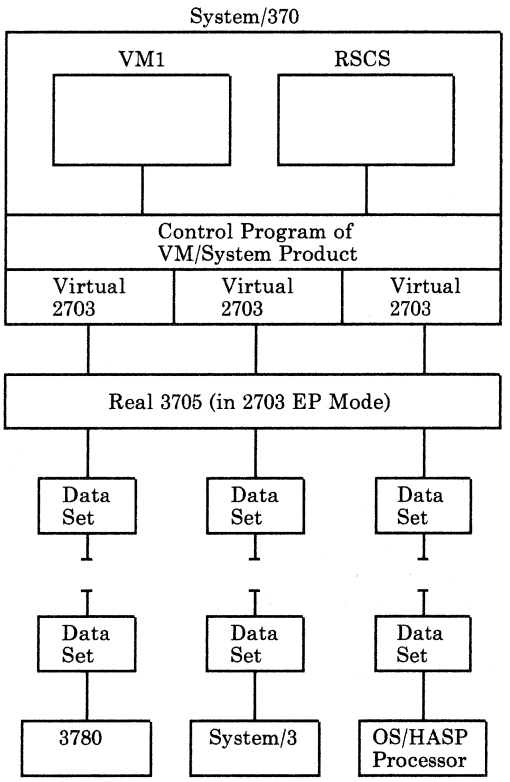


Figure 18. A Remote Spooling Communications Subsystem

RSCS Planning Considerations

Before you perform the RSCS generation procedure, be sure you have a virtual machine defined for RSCS. The virtual machine you intend to run RSCS should have:

- 512K of virtual storage
- A reader
- A console
- A minidisk for the RSCS system residence volume. The RSCS system disk must have a write password.

You can define more than one RSCS virtual machine. Also, you can have more than one RSCS virtual machine running at the same time. When multiple RSCS virtual machines are running at the same time, each must have a unique user identification (userid) and local location identification (ID = linkid).

When you code the GENLINK macros during the RSCS generation procedure, you must code the local location identification on the first GENLINK macro.

The IX/370 Handshaking Feature

The IX/370 handshaking feature improves the performance of IX/370 systems running under VM/SP or VM/SP HPO. Installations running IX/370 under VM incur additional overhead because of VM's handling of Supervisor Call instructions (SVC) 9, 10, and 11. The IX/370 handshaking feature reduces the overhead by accelerating SVCs 9, 10, and 11 issued by authorized IX/370 virtual machines.

To invoke this support, the IX/370 virtual machine must:

1. Specify the SVCACCL parameter of the OPTION control statement in the directory. For example:

   ```
   OPTION SVCACCL
   ```

2. Issue the class G SET command to enable SVCACCL. For example:

   ```
   set svcaccl on
   ```

You can issue the CP QUERY SET command to verify the SVCACCL setting.

Chapter 10. Planning for I/O

Some of the planning you must do before you generate VM/SP HPO concerns the configuration of input and output devices. The topics covered here are:

- Channel Switching
- Alternate Path Support
- 3880 Storage Subsystem
- Channel Set Switching Facility
- Monitoring and Service Support Facility
- Input/Output Configuration Program
- Operating Systems Using Reserve/Release
- Missing Interruption Handler
- Logical Device Support Facility
- Inter-User Communication Vehicle
- Virtual Machine Communication Facility
- Extended Channel Support

Channel Switching

The two- or four-channel switch can be used in the following cases:

- Two processors, one running VM/SP HPO, the other running an operating system that supports channel switching.

- Two virtual machines running under VM/SP HPO. Each virtual machine operating system must support the channel switch feature (CMS does not).

- A single virtual machine running under VM/SP HPO. The virtual machine operating system must support the channel switch feature.

- A processor running VM/SP HPO and managing multiple paths to devices through VM/SP HPO alternate path support. Refer to "Alternate Path Support" on page 88.

You can use the two- or four-channel switch for devices attached to two processors. For example, one processor could be running VM/SP HPO and the other could be running OS, as shown in Figure 19.

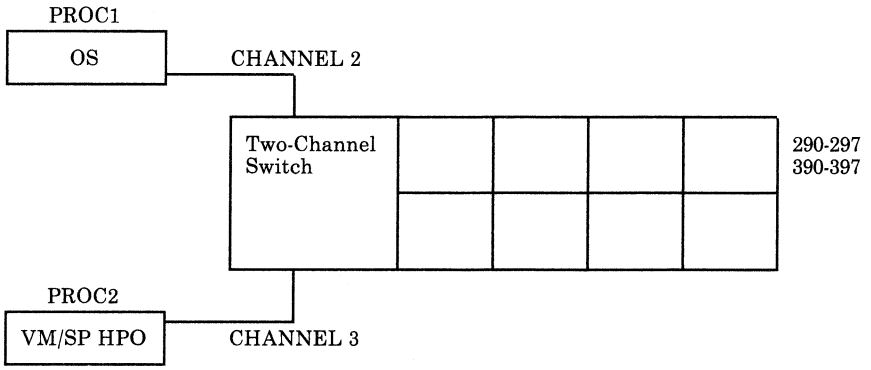


Figure 19. Channel Switching between Two Processors

VM/SP HPO requires the following RDEVICE and RCTLUNIT macro instructions to support this configuration:

```
RDEVICE  ADDRESS=(290,8),DEVTYPE=3330
RDEVICE  ADDRESS=(390,8),DEVTYPE=3330
RCTLUNIT ADDRESS=290,CUTYPE=3830
RCTLUNIT ADDRESS=390,CUTYPE=3830
```

These macro instructions make it possible for you to run VM/SP HPO on PROC1 or PROC2. If you are always going to run VM/SP HPO on PROC2, you can eliminate one path (eliminate one set of RDEVICE and RCTLUNIT macro instructions).

If any I/O devices controlled by CP for its own exclusive use are attached to a control unit by a two- or four-channel switch, the processor controlling the other channel interface must vary the CP-owned devices offline. For example, if all eight disks in the preceding configuration are mounted, and two of those disks are CP-owned volumes (such as CP system residence and CP paging and spooling volumes), the OS system running on PROC1 must vary the CP-owned volumes offline. This procedure protects volumes that CP needs.

You can also use the two- or four-channel switch for devices attached to one processor that is running VM/SP HPO. For example, one processor could be running VM/SP HPO with OS running in a virtual machine as shown in Figure 20. In this case, the virtual machine operating system supports channel switching.

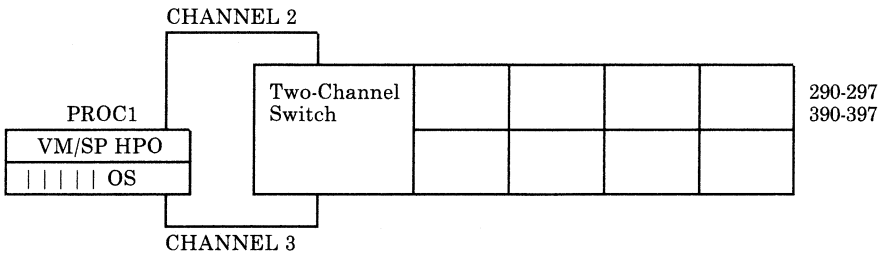


Figure 20. Channel Switching on One Processor

VM/SP HPO requires the following RDEVICE and RCTLUNIT macro instructions to support this configuration:

```
RDEVICE  ADDRESS=(290,8),DEVTYPE=3350
RDEVICE  ADDRESS=(390,8),DEVTYPE=3350
RCTLUNIT ADDRESS=290,CUTYPE=3880
RCTLUNIT ADDRESS=390,CUTYPE=3880
```

For this example, you should have all devices associated with one path offline when you load VM/SP HPO. Otherwise, the following message is displayed:

```
DMKCPI954E DASD raddr VOLID volid NOT MOUNTED,
           DUPLICATE OF DASD raddr
```

The 3880 Storage Subsystem contains two storage directors. Each storage director acts as a control unit providing input/output operations to a string of devices. Since each storage director can be addressed separately, you must code one RCTLUNIT macro statement for each module. The optional ALTCH operand of the RCTLUNIT macro allows specification of up to three alternate channel interfaces to a single storage director. The two- or four-channel switch feature allows up to four channels to have access to a storage director. VM/SP HPO supports a maximum of four channel paths to a single storage director.

DASDs can be used by OS running in a virtual machine if they are dedicated to that virtual machine via the ATTACH command or the DEDICATE control statement in the virtual machine directory entry. Device addresses generated for the virtual machine operating system need not be the same as those defined for the real machine.

As another example, consider channel switching for tapes. If the real configuration includes a 2816 Switching Unit or a two- or four-channel switch feature, it can be made to operate under control of a virtual machine operating system. For example, if 580 and 680 are the alternate device addresses for a particular tape drive, then:

- Generate the virtual machine operating system for the appropriate hardware (in this case a 2816 Switching Unit on channels 5 and 6).
- Generate CP as though 580 and 680 are different devices (with different control units and channels).
- Issue the CP ATTACH command for both device addresses (580 and 680) whenever the real device is to be attached to the virtual machine.

Device addresses generated for the virtual machine operating system do not need to be the same as those on the real machine.

The devices must be used by the virtual machine as dedicated devices (attached, or defined with a DEDICATE statement in the directory entry).

Alternate Path Support

Alternate path logic provides support for the Two-Channel Switch, the Two-Channel Switch Additional Feature, and the String Switch Feature. This support allows up to four channels on one control unit to be attached to VM/SP HPO and/or one device to be attached to two logical control units. This allows the control program up to eight paths to a given device when the maximum number of alternate channels and alternate control units are specified. When an I/O request is received for a device, CP can select a free path from any of the available paths to the device. With this support, even though the primary path to a device is busy, there may exist an alternate path(s) that is available. Instead of the I/O request being queued, it can be started immediately on an alternate path. In the case where no available path to the device exists, alternate path I/O scheduling is implemented in such a way that the request is queued off multiple busy/scheduled paths. The first path to become available will be the path the I/O is started on. This approach has some distinct advantages over approaches used by other operating systems:

1. The I/O starts on the first available path to the device. This eliminates the arbitrary choice of queuing based on number of IOBLOKs already queued, primary path, last busy scheduled path encountered, etc.

2. No user is penalized more than any other user.

3. The first in, first out (FIFO) principle is maintained.

Figure 21 shows how the real I/O control block structure is coded, and how it logically appears when an alternate control unit is specified.

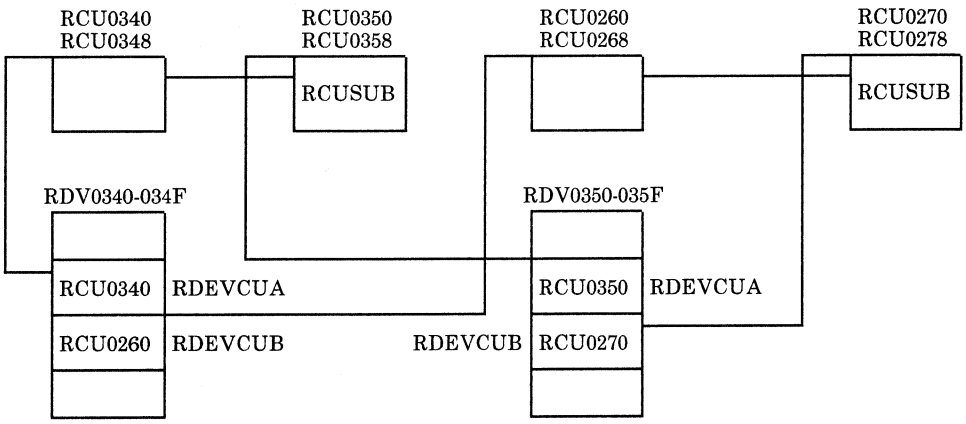


Figure 21. Real I/O Control Block Structure for Alternate Control Unit Specification

To specify an alternate control unit on the RDEVICE macro, code:

```
RDEVICE ADDRESS=ccuu,DEVTYPE=nnnn,MODEL=n,ALTCU=ccuu
RDEVICE ADDRESS=(340,32),DVTYPE=3330,MODEL=1,ALTCU=260
RCTLUNIT ADDRESS=340,CUTYPE=3830,FEATURE=32-DEVICE
RCTLUNIT ADDRESS=260,CUTYPE=3830,FEATURE=32-DEVICE
```

To specify alternate channel addresses on the RCTLUNIT macro, code:

```
RCTLUNIT ADDRESS=ccuu,CUTYPE=nnnn,FEATURE=xxx-DEVICE,    X
         ALTCH=(1,2,4)
```

Figure 22 shows how the real I/O control block structure would be coded and logically appear when alternate channels are specified. Note that the subordinate control unit blocks do not contain pointers to the alternate channel blocks. Only the prime control unit block contains pointers to the alternate RCHBLOKS. This is consistent with the current CP block structure.

```
RCTLUNIT ADDRESS=340,CUTYPE=3830,FEATURE=32-DEVICE,     X
         ALTCH=(1,2,4)
RCHANNEL ADDRESS=1,CHTYPE=MULTIPLEXOR
RCHANNEL ADDRESS=2,CHTYPE=MULTIPLEXOR
RCHANNEL ADDRESS=3,CHTYPE=MULTIPLEXOR
RCHANNEL ADDRESS=4,CHTYPE=MULTIPLEXOR
```

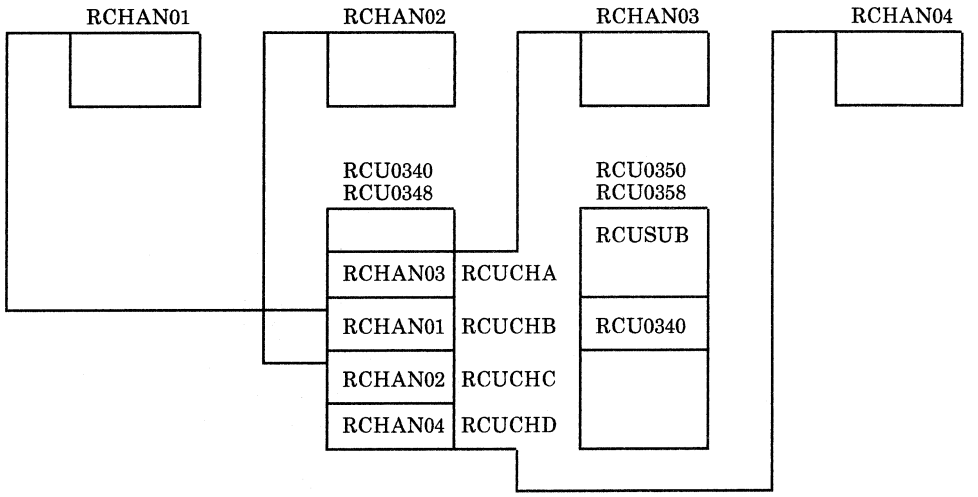


Figure 22. Real I/O Control Block Structure for Alternate Channel Specification

Restrictions

The following restrictions apply directly to Alternate Path processing:

- VM/SP HPO does not support alternate paths for devices that issue attention interruptions to invoke a read response from the host; for example, the 3851 Mass Storage Control (MSC) unit.
- Only one alternate channel can be specified for MP configured systems.

3880 Storage Subsystem

VM/SP HPO can use a 3880 Model 11, 21, 13, or 23 Storage Subsystem to manage DASDs. In addition, the Model 11 and Model 21 can be used to perform preferred system paging and swapping.

Configuration

The 3880 Storage Subsystem must consist of a 3880 Control Unit, Model 11, 21, 13, or 23, and one or two strings of DASDs. The table below shows the kind and maximum number of DASDs that can be attached to each model of the 3880 Storage Subsystem.

3880 Model Number	Type of DASD	Number of Strings	DASD Per String	Total DASD
11	3350	1	8	8
21	3350	2	4	8
13	3380	2	16	32
23	3380	2	16	32

All models of the 3880 Storage Subsystem consist of three functional components: two storage directors, a cache, and the 3350 or 3380 DASDs. The storage directors and the cache reside together in the control unit. In all models, the storage directors control the storage subsystem; and the cache and DASDs comprise two levels of storage.

All eight devices attached to a 3880 Model 11 Storage Subsystem are handled by one storage director. The Model 21 has two storage directors. Both storage directors can be logically attached to one or two strings of one to four storage devices each. In the Model 13 and Model 23, two storage directors are each logically attached to one or two strings of one to sixteen storage devices.

Note: The two DASD strings that can be attached to each of the storage directors in a Model 13, 21, or 23 are the same *two strings.*

The two storage directors in a 3880 Model 21, Model 13, or Model 23 Storage Subsystem may be in the same physical 3880 (*single-frame configuration*) or in two different physical 3880s (*dual-frame configuration*). In a dual-frame configuration, both physical 3880s must be the same model: both 3880-21s, both 3880-13s, or both 3880-23s. Figure 23 on page 91 shows a dual-frame configuration. Dual-frame configuration is preferable because it permits access to the data on all the DASDs even when one of the 3880s is unavailable.

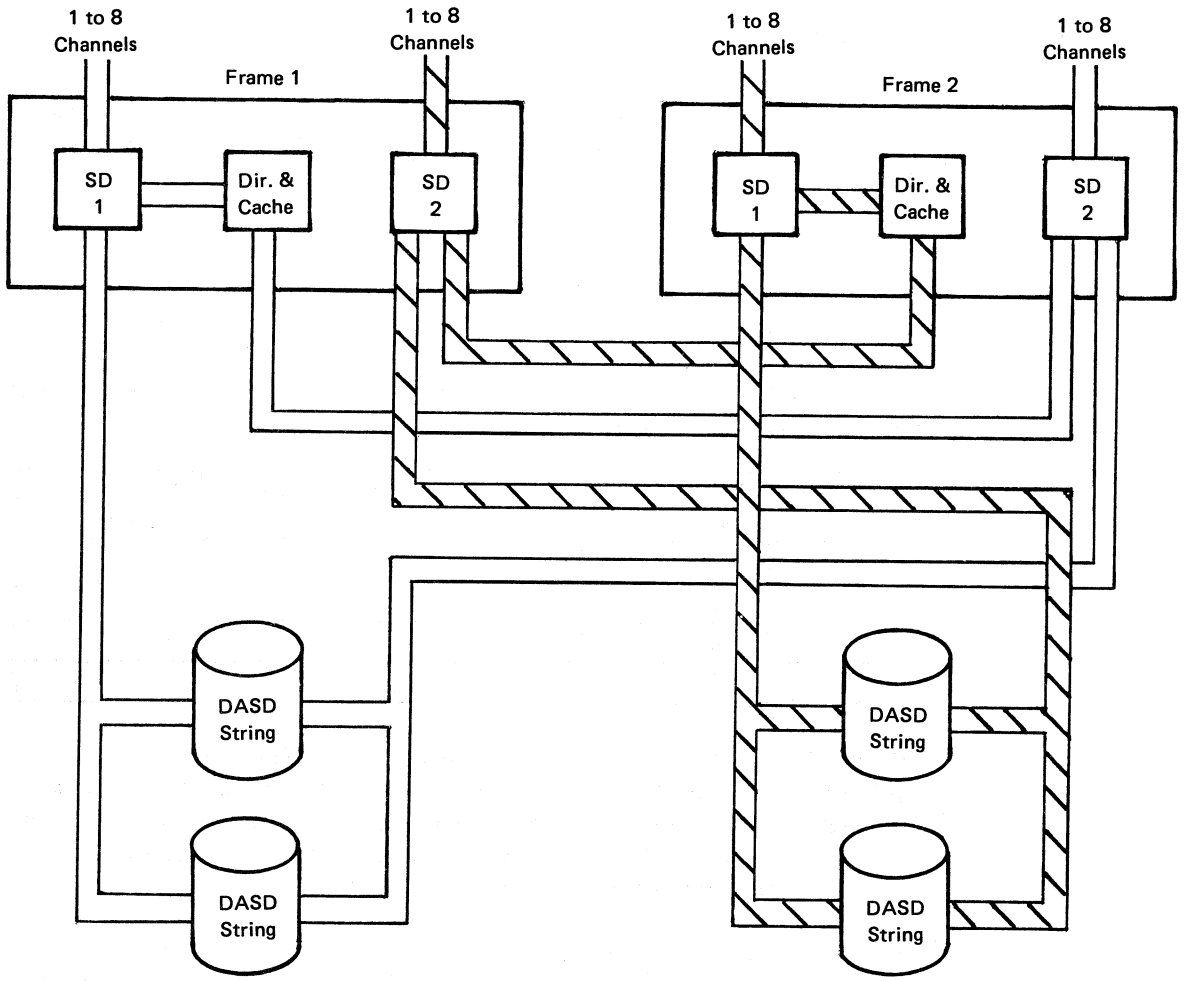


Figure 23. 3880 Storage Subsystem Dual-Frame Configuration

The Cache

Model 11 and Model 21

The Model 11 and Model 21 function in either of two modes: paging or direct. Direct mode I/O does not go through the cache and is used to access the volume label and VTOC on the 3350 devices. Paging mode goes through the cache. On the Model 11, only one storage director functions in paging mode.

Model 13 and Model 23

The cache of the Model 13 and Model 23 is a write-through cache, not intended for paging. All data written to the cache is also written to one of the 3380 DASDs. Consequently, writing to cache is slower than writing to an ordinary DASD and should be avoided. Reserve the cache for read-only data.

The MDISK directory control statement determines whether data written to a given minidisk is also written to the cache. You can also force all minidisks to bypass the cache by issuing the CACHE OFF command to disable caching at the device level. You should use a Model 13 or 23 for paging, spooling, or swapping only if the cache is disabled in this way.

A minidisk on a 3880 Model 13 or Model 23 Storage Subsystem may use the cache under the following conditions:

- The CACHE ON command with the SUBSYS option must have been issued to enable the subsystem for caching.
- The CACHE ON command must have been issued to enable the DASD on which the minidisk is located for caching.
- The minidisk must be defined with a DEVTYPE of 3380C on the MDISK control statement.
- In a 3880-13 Storage Subsystem, both storage directors must be available. That is, the ALTCU operand of the RDEVICE macro must be coded. (Only one storage director need be available on the 3880-23).

Note: Powering a 3380-AA4 off and then back on can affect the caching status not only of the device but of the whole 3880 Storage Subsystem. In such a case, you should issue the CACHE QUERY command and, if necessary, reissue the CACHE ON/OFF command to make sure that the caching status is what you want.

The minidisks allowed to access the cache should have a:

- High read-to-write ratio
- Very high I/O rate from interactive users

The following minidisks usually meet these requirements:

- CMS S-disk
- CMS Y-disk
- PROFS 399 program disk (Program Number 5664-176)
- CMS tools disks containing programs and files accessed by most of the users

Any read-only minidisk of which the system has several copies is a good candidate for caching on a 3880 Model 13 or 23.

Addressing

The 3880 Storage Subsystem is invoked by specifying it in the RDEVICE, RCTLUNIT, and SYSPAG macros at system generation time. You must understand the addressing schemes of the different models of the 3880 Storage Subsystem in order to code the DMKRIO files correctly. See Part 2 of this book for details of coding the RDEVICE, RCTLUNIT, and SYSPAG macros.

Model 11 and Model 21

There are four addresses associated with each 3880 Model 11 or Model 21 Storage Subsystem: one base address and three paging addresses. As a result, four channel programs can address the same device at the same time, as long as all the data required is in the cache.

When you code the RDEVICE macro for the Model 11, you will code the base mode address of the first storage device you plan to attach. The digit indicating the control unit in each base address must be an even number. The last digit must be between 0 and 7. Since the base addresses are generated in sequence, if you are planning to attach eight storage devices, the address for the first storage device should end in 0, so that the address of the last storage device will end within the permitted range.

The paging addresses are generated in increments of eight from the base address. For example, if you address your first storage device at 240, its base address will be 240 and its paging addresses will be 248, 250, and 258. A second storage device can have a base address of 241 and paging addresses of 249, 251, and 259.

For the Model 21, the base address of each storage device must end in a digit between 0 and 3. Since base addresses are generated in sequence, remember to make the address of the first storage device in the string end in 0 if you plan to attach four devices, so that the address of the fourth device will not end in a digit greater than 3.

The paging addresses for storage devices attached to the Model 21 are generated from the base address in increments of four. If the base address of the first storage device attached to a 3880 Model 21 Storage Subsystem is 240, the paging addresses will be 244, 248, and 24C. Three more storage devices can be attached at 241, 242, and 243. No DASD can be attached to the Model 21 at 244, since 244 is a paging address; but a second string of one to four DASDs can be attached beginning at 250. IBM recommends, however, that you attach the second string to some other channel.

In the Model 21, the 3350 DASDs must have the *same address* from the point of view of either storage director. In other words, the device portions of the ADDR and ALTCU operands of the RDEVICE macro must be the same.

Figure 24 illustrates the 3880 Model 11 and Model 21 Storage Subsystem addressing schemes.

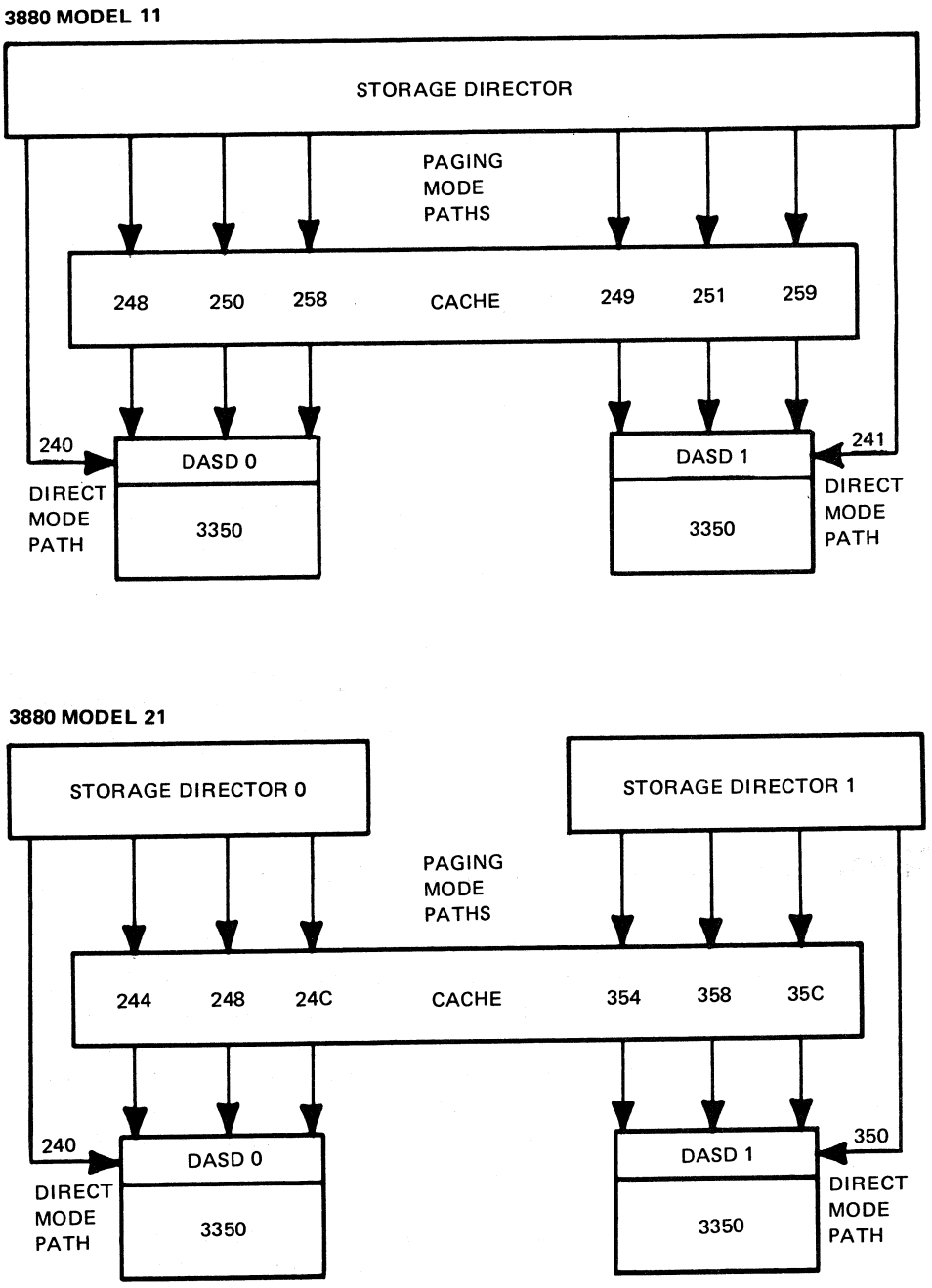


Figure 24. Examples of the 3880 Model 11 and Model 21 Storage Subsystem

Model 13 and Model 23

There is only one special restriction on addressing a 3880 Model 13 or Model 23 Storage Subsystem. The 3380 DASDs must have the *same address* from the point of view of either storage director. In other words, the device portions of the ADDR and ALTCU operands of the RDEVICE macro must be the same.

Error Conditions

If you attempt to generate a 3880 Model 11 or Model 21 Storage Subsystem at an improper address, the following MNOTE is generated:

```
MNOTE:  INVALID DEVICE ADDRESS.
```

If you generate an incorrect type of 3880 Storage Subsystem, the following message is issued at initialization or vary on time:

```
DMKxxx328E raddr1 raddr2 raddr3 raddr4 IMPROPERLY GENNED -
VARIED OFFLINE
```

xxx
is CPJ if the message is issued at initialization and CPZ if it is issued at vary on time.

If you attempt to put a CP nucleus onto a 3880 Model 11 or 21 Storage Subsystem device or IPL a VM/SP nucleus from a 3880 Model 11 or 21 Storage Subsystem device, an error condition results and a wait state 11 is returned.

If you install a 3880 Storage Subsystem with the 308x processor, you will have to perform an IOCP generation. See "Considerations for Coding the Input/Output Configuration Source File" on page 294 for further information on generating the IOCP.

Channel-Set Switching Facility

The channel-set switching facility is available on the 3033 attached processor and multiprocessor systems and the 308x and 3090 Processor Complexes. (It is not available on the 4381 Model 3 Processor). This feature permits a set of channels to be switched from one processor to another in a multiprocessor or attached processor environment. A channel-set is the collection of channels that are switched as a group. On a 3033 attached processor system, all online channels comprise the channel-set.

VM/SP HPO, when generated for AP operation, uses the channel-set switching facility. The switching operation directs the execution of I/O instructions and I/O interruptions from the main processor to an attached processor, thus permitting an operator to vary the main processor offline. The switching operation does not control other channel activity, such as data-transfer operations and chaining.

In 3033 and 308x attached processor environments, channel-set switching is used to continue system operation in uniprocessor mode when the main (I/O) processor is taken offline as the result of a VARY OFFLINE PROCESSOR command or a main processor failure. This support switches the channel-set from the main processor to the attached processor.

There are no required system generation macro instructions to support channel-set switching. In the event of a failure on the main (I/O) processor, the automatic processor recovery routine determines if channel-set switching capability exists. If there is no channel-set switching capability in the system, CP enters the wait state with a code of X'0001'. If the error is TOD clock damage or a malfunction alert on the main (I/O) processor and the processor is in problem state, the failing processor is taken offline, provided that the attached processor is equipped with the channel-set switching facility. The channel-set switching feature is used to disconnect the channel-set from the failing processor and to reconnect the channel-set to the attached processor. Processing continues on the attached processor in uniprocessor mode. This message is issued:

```
DMKCPU623I CHANNEL-SET CONNECTED TO PROCESSOR nn
DMKMCT623I
```

Monitoring and Service Support Facility

The 308x and 3090 Processor Complexes use processor controllers to coordinate central communications. Each processor controller performs the following functions automatically:

- Validates storage during system initialization
- Configures channel groups
- Detects and corrects recoverable channel errors
- Monitors power and coolant levels

The monitoring and service support facility (MSSF) is a hardware component of the 3082 processor controller (the processor controller for the 308x Processor Complex). MSSF provides I/O configuration and storage information for the 308x processor complex. The service processor is the corresponding component of the 3090's processor controller. Virtual machine operating systems that are able to communicate with the MSSF or the service processor can use the MSSF command word SCPINFO, or the READ SCP INFORMATION service call, to obtain information about the configuration and storage allocation of the 308x or 3090 Processor Complex.

The MSSF and service processor also process VARY PROCESSOR commands. When you issue the CP command VARY PROC ONLINE or VARY PROC OFFLINE VPHY on a 308x or 3090 processor, CP generates an MSSFCALL instruction to the MSSF, or a service call to the service processor. MSSF or the service processor then physically brings the processor online or takes the processor offline. When the request finishes, MSSF or the service processor returns a completion status code.

To get information for the 3090 Processor Complex on hardware configuration, CP generates the READ SCP INFORMATION service call. CP uses the MSSFCALL diagnose instruction and MSSF command words to communicate with the MSSF. The MSSFCALL diagnose instruction (function code X'80'), service call (X'B220'), MSSF command words and

completion status codes are described in the *VM/SP HPO System Programmer's Guide*. The *VM/SP HPO System Logic and Problem Determination Guide—CP* describes the response codes that CP simulates for a V=V virtual machine.

Neither MSSF nor the service processor requires system generation macro definitions. You *must*, however, define your *real* I/O configuration to the processor controller using the Input/Output Configuration Program. General information about the Input/Output Configuration Program follows.

Input/Output Configuration Program

The processor controller contains the I/O configuration definition for the 308x and 3090 Processor Complexes. You supply this information to the controller by executing the Input/Output Configuration Program.

The Input/Output Configuration Program (IOCP) processes IOCP macro definitions that contain I/O configuration information for the 308x or 3090 Processor Complex. Using IOCP macro definitions, IOCP records the information in an input/output configuration data set. The input/output configuration data set (IOCDS), which is stored in the processor controller, supplies the information that defines the I/O configuration for the Processor Complex. A 3084 Processor Complex has two sides, called side A and side B. Each side has four I/O configuration data sets. The I/O configuration data sets on side A are called A0, A1, A2, and A3; on side B they are called B0, B1, B2, and B3. Except for the 3081 Model D Processor Complex, all of the other 308X Processor Complexes have only one side A and four I/O configuration data sets, as does the 3090 Processor Complex.

The IOCDS contains information that describes:

- Channel paths to the processor
- Control units assigned to the channel paths
- I/O devices assigned to the control units

There are three versions of IOCP:

Stand-Alone version
VM version
MVS version

You should run the appropriate version of IOCP as determined by the following descriptions.

The VM version of IOCP means that version that runs under all releases of the following operating systems: VM/System Product (5664-167), VM/System Product High Performance Option (5664-173), and the VM/Extended Architecture Migration Aid (5664-169).

Stand-Alone Input/Output Configuration Program

If you are installing a 308x or 3090 processor for the first time, it may be necessary to execute the stand-alone version of IOCP. The stand-alone version of IOCP defines your initial I/O configuration to the Processor Complex. Using the 3081 service support console or the system console, you can run the stand-alone version of IOCP to:

- Read the input/output data set (IOCDS) from the processor controller
- Display, add, alter, and delete I/O configuration data from the IOCDS
- Obtain I/O configuration reports
- Write a new or updated IOCDS

The stand-alone version of IOCP is shipped with the processor complex as a software program. It resides in a partitioned data set on the integrated processor controller file in the processor controller. You should examine the starter IOCDS to determine whether a sufficient configuration of I/O devices, control units, and channels is defined and whether it matches your system configuration. A listing of the starter IOCDS appears in the *Input/Output Configuration Program User's Guide and Reference* for your processor.

If there are enough entries in the starter IOCDS for you to generate VM/SP HPO, it is not necessary to execute the stand-alone version of IOCP. Simply power-on reset the processor controller and generate your VM/SP HPO system. The processor controller uses the starter IOCDS defined in the level 0 IOCDS of the processor controller. After VM/SP HPO is generated, and CMS is operating, use the VM/SP Input/Output Configuration Program to fully define your configuration to the processor controller.

If the starter IOCDS does not meet your needs, execute the stand-alone version of IOCP.

VM Input/Output Configuration Program

VM IOCP executes under the VM/System Product version of CMS. Using the CMS command IOCP with the necessary options, you can generate new I/O configuration data or obtain I/O configuration reports. If you have an operating VM/SP HPO system (that is, you have defined a sufficient configuration to the processor controller in order to generate VM/SP HPO) you can use the IOCP command to define your entire system configuration.

The IOCP command writes the new input/output configuration source file to IOCDS in the processor controller that you do not write-protect. You can write-protect each IOCDS individually by using the IOCDMS (SYS021) frame on the system console. To test the new IOCDS:

1. Generate a new VM/SP HPO system that has a real I/O file (DMKRIO) modified to reflect the new I/O configuration.
2. Shut down the VM/SP HPO system.
3. Power-on reset the processor controller, using the appropriate level IOCDS.
4. Load (IPL) the new VM/SP HPO system.

MVS Input/Output Configuration Program

The MVS version of IOCP runs as a job under control of the OS/VS2 MVS system control program with the OS/VS2 MVS/System Product. You use JCL statements to execute IOCP. By coding options on the PARM= parameter of the EXEC statement, you can generate a new input/output configuration data set (IOCDS) or produce I/O configuration reports from the IOCDS in storage.

Using IOCP in Single Processor Mode

If VM/SP HPO is operating in single processor mode and any virtual machine (CMS or MVS/SP) causes IOCP to read from or write to the processor controller, CP locates and stops (SIGP STOP) the processor dedicated to the virtual=real virtual machine. Then CP executes the read or write function. When the service signal external interruption occurs, CP restarts the dedicated processor.

References

For more information about IOCP source file and matching real I/O configuration file, refer to Chapter 21, "Preparing the Real I/O Configuration File (DMKRIO)."

For details concerning IOCP macro instructions, the CMS IOCP command and the starter IOCDS, refer to the *Input/Output Configuration Program User's Guide and Reference.*

Operating Systems Using Reserve/Release

Shared DASD is the capability of accessing direct access devices from two or more systems. The systems can be in virtual machines, on the same real processor, or on different real processors. Device access by the sharing systems is sequential.

Sharing of DASD volumes can occur when:

- A two- or four-channel switch attaches a device's control unit to two or four channels.
- String switching is used and the control units to which the DASD volumes are switched are on channels of two different systems.

With shared DASD, an I/O operation may be started to a shared device from any of the systems able to access the device using the switch. Each sharing system waits for the programmable switch to gain device access. The first requesting system gets the switch set to its interface so that it may perform I/O operations to a shared device. When the switch returns to neutral, any other system, or the same one, may select the shared device and have the switch set to its interface.

It is important to note that none of the sharing systems is aware of what the other is doing with data on the shared devices. Data integrity is the responsibility of the using program. For this reason, a program may issue the RESERVE hardware command to retain exclusive use of a shared device while a critical update to data is being performed. Device RELEASE is issued to terminate exclusive reservation. If a shared device has been reserved for exclusive use, the system channel through which RESERVE was issued will lock out any other channel, on the same or different system, from accessing the device.

There are several reasons why you would elect to share devices between systems:

- Scheduling of jobs is simplified, and operator intervention is minimized. Instead of being moved from one system to another, the volume remains mounted and available to each system able to access the data by means of the two- or four-channel switch or string switch.
- Updating of data is reduced. One update to a shared data set is needed, instead of the multiple updates required if each of several systems had its own copy of the data set.
- Backup and switchover in the event of hardware failure is made easier in a multi-system environment if the needed data is accessible to surviving systems without moving it.
- Direct access storage space may be saved because one copy of the data is required instead of multiple copies.

Two assembler language macros, RESERVE and DEQ, are available for reserving and releasing of a device. Data integrity of shared devices can be maintained by application program use of the RESERVE macro, or by operating system components that automatically issue the RESERVE macro if the target of their update operation is to a shared device. CMS does not make use of these macros in its CMS file system. In addition, CMS does not support these macros in OS simulation or CMS/DOS. The SHAREOPTIONS operand on the Access Method Services control statement has no function in CMS. No attempt is made by CMS VSAM to reserve or release system resources. Use of shared DASD by virtual machines should be limited to guest operating systems that will maintain the integrity of shared data, such as catalogs, VTOCS, program libraries, etc.. These guest operating systems should also support the use of the RESERVE and DEQ macros used by application programs running under these systems. The only other alternative is use of hardware reserve or release CCWs by an application program running under CMS. In this case, the application program issues hardware reserve and release CCWs in a SIO or DIAGNOSE operation to the shared device.

Reserve/release support can be addressed in two forms:

- Hardware-supported reserve/release
- Virtual reserve/release

Hardware-supported reserve/release is the use of reserve/release CCW strings by virtual machine or processor operating systems to preserve data integrity. Data integrity is preserved by the hardware on a device basis during the interval of time between the reserve and release CCWs by not allowing access to the reserved device via any other path.

Virtual reserve/release is software simulation of reserve/release CCWs for minidisks. Since virtual devices associated with a minidisk all map to the same real channel interface to the device, hardware protection is lost, and a software locking structure is required to maintain data integrity during reserve/release sequences.

CP and CMS do not issue reserve CCWs. The use of reserve/release is the responsibility of the virtual machine operating system. The VM/SP HPO initialization routine issues a release CCW to tape and DASD volumes to determine if the two- or four-channel switch feature is installed.

Hardware-Supported Reserve/Release

Operating systems with hardware-supported reserve/release use reserve/release CCWs to preserve data integrity in the following cases:

- Two virtual machines with each operating system having a separate channel path to the device to be shared. Each virtual operating system uses reserve/release CCWs to preserve data integrity.

  Reserve/release CCWs are recognized by the CP CCW translation routine and are executed by the hardware to preserve data integrity. In this case devices should be generated, at system generation time in

DMKRIO, as separate devices. Each device should be dedicated to a virtual machine by means of the ATTACH command or DEDICATE control statement in the directory.

- A virtual machine runs under VM/SP HPO and shares a device with another processor. The operating system in the virtual machine uses reserve/release CCWs to preserve data integrity. The operating system running on the other processor can be VM/SP HPO, in which case the virtual machine operating system uses reserve/release CCWs to maintain data integrity. It can also be a non-VM/SP HPO operating system with reserve/release capability.

  To support this environment, the device should be dedicated to the virtual machine by means of the ATTACH command or DEDICATE control statement in the directory.

  The device can be defined as a minidisk on the VM/SP HPO processor, which begins at real cylinder 0. Do not use virtual reserve/release support in this environment. The volume being shared should not contain more than one minidisk or be used for CP paging, spooling, etc., since reservation by the other processor could lock out virtual machine users or VM/SP system I/O requests to the same device.

In the above shared DASD environments, the use of reserve/release by virtual machine operating systems and alternate path support are mutually exclusive. CP changes a reserve CCW to a sense CCW when an alternate path is defined for the device. In a multiprocessor (MP) system, an alternate path is defined if both processors have a channel path to the device. The protection offered by hardware reserve is lost. A single path should be defined in VM/SP HPO for devices that will be dedicated to virtual machines, and then shared between other virtual machines or processors.

When an I/O error occurs on a device, the processor maintains a contingent connection for that device until a SENSE CHANNEL command is executed and sense data is recorded. No other I/O activity can take place on the device during this time. Under VM/SP HPO, the contingent connection is maintained until the SENSE command is executed, but I/O activity from other virtual machines or another processor can begin on the device while the sense data is being reflected to the virtual machine. The user should therefore be aware that, on a shared disk, the access mechanism may have moved during this time.

The performance of virtual machine operating systems may be degraded when sharing DASD. If not running in single processor mode, when a device is being used by another virtual machine or system, I/O requests may be queued and the virtual machine left in an I/O wait state. If running in single processor mode, a device busy state is reflected up to the V=R guest when a device is in use and a device end interrupt is reflected when an interruption occurs. Depending on contention for devices, the V=R guest may get multiple device busy states and device end interrupts before the device is available to it.

Virtual Reserve/Release

Reserve/release software simulation in VM/SP HPO provides reserve/release protection at the minidisk level, including full volume minidisks. Virtual reserve/release is intended for use by virtual machines that support shared DASD (not CMS) running on the VM/SP HPO processor. Virtual reserve/release simulation is requested by appending a character "V" to the mode operand on the MDISK directory statement. All future links to this minidisk are subject to virtual reserve/release processing. A software locking structure is created to manage the reservation status by minidisk. CP then examines virtual machine channel programs and manages reserve/release CCWs presented by sharing virtual machines. CP simulates hardware reserve by reflecting a "device busy" condition in response to a virtual machine SIO when the minidisk is already reserved by another virtual machine. When the minidisk is released, a "device end" interrupt is reflected to all virtual machine users who received a "device busy" indication. DIAGNOSE users can also issue reserve/release CCWs. No "device busy" or "device end" status, however, is reflected to the virtual machine. If a minidisk is already reserved, a subsequent DIAGNOSE request for another virtual machine is queued until the minidisk is released, when the DIAGNOSE request will be redriven.

Control Program Handling of a Reserve CCW

Reserve/release support and alternate path support are mutually exclusive. The CCW translation routine changes a reserve CCW to a sense CCW when alternate paths have been defined to the device from the VM/SP HPO processor. Data integrity is not preserved when sharing a device between processors or virtual machines if alternate paths are defined. In an MP system, an alternate path is defined if both processors have a symmetrical channel path to the device. When using virtual reserve/release to share a minidisk between virtual machines on the processor, CP still changes a reserve CCW to a sense CCW when alternate paths are defined to the real device; but since hardware reserve/release is simulated when virtual reserve/release is being used, data integrity is preserved when alternate paths are defined. Figure 25 on page 104 identifies those cases when CP changes a reserve CCW to a sense CCW.

Restrictions

Device Sharing Between Real Processors

- When a device is shared between processors and at least one of the processors is running VM/SP HPO, the shared volume cannot contain more than one minidisk. The single minidisk may encompass the entire volume or a small portion of the volume. The remainder of the volume must not be referenced by CP for use as paging, spooling, etc., or by any virtual machine.

Note: This restriction applies only both real and virtual reserve/release are used. Assume that two virtual machines use separate minidisks on the same real volume and that both minidisks are defined for virtual reserve/release.

1. *Virtual machine A issues a reserve to minidisk A, resulting in a RESERVE CCW to the real volume.*
2. *Virtual machine B issues a release to minidisk B, resulting in a RELEASE CCW to the real volume.*
3. *Another real machine can now write to that volume, including the minidisk A area.*

- Devices shared between processors must not be generated in DMKRIO as having alternate paths. If there are multiple paths from the VM/SP HPO processor to the shared devices, as well as a path from the same devices to another processor, the paths from the VM/SP HPO processor cannot be generated in DMKRIO as alternate paths via the ALTCH or ALTCU macro operands. *This means that the definition of alternate paths in DMKRIO and the use of real reserve/release are mutually exclusive.*

Type of Device	Alternate Path Support	Reserve/Release Executes in the Hardware (2-4 Channel Switch)	Virtual Reserve Release Requested (V Added to Mode in MDISK)	Sent by CP to Device	CCW Comnd Note
Dedicated DASD or Tape	Not defined	Not applicable	Not applicable	Reserve	1
	Defined	Not applicable	Not applicable	Sense	2, 6
Minidisk	Not defined	Yes	No	Reserve	1
	Not defined	Yes	Yes	Reserve	1
	Not defined	No	No	Reserve	3
	Not defined	No	Yes	Sense	4
	Defined	Not applicable	Not applicable	Sense	5

[1]Normal Operation. The command is passed unchanged to the hardware.

[2]When the VM/SP HPO system has been generated with alternate path support for those devices, it prevents the devices from being reserved. This action causes CP to avoid a possible channel lockout. CP does not return any indication that the device was not reserved to the operating system issuing the CCW command.

[3]Without the two-channel switch special feature, CP sends the reserve/release CCW command unchanged to the hardware. However, the hardware rejects the command and does not reserve the device.

[4]Before sending the command to the hardware, CP changes the reserve CCW command to a SENSE CCW command, and places a virtual reserve on the minidisk. The real device is not reserved. The virtual reserve prevents other operating systems running under the same VM/SP HPO system from accessing the minidisk. However, these same virtual operating systems may virtually reserve other minidisks located on the same real volume. Because the two-channel switch feature is not installed on the channels, only one address path goes to the device from the VM/SP HPO processor. This path allows virtual reserve/release processing to send a SENSE CCW to the device, although the reserve CCW command is rejected by the hardware.

[5]When alternate paths to a device have been defined (by the ALTCU operand on the RDEVICE macro instruction and the ALTCH operand on the RCTLUNIT macro instruction), CP changes reserve/release CCW commands to SENSE CCW commands to prevent a possible channel lockout.

[6]In an MP system, an alternate path is defined if both processors have a symmetrical channel path to the device.

Figure 25. Summary of Reserve/Release Support

Device/Minidisk Sharing on a Single Processor

- If more than one path to a volume exists, DMKRIO may be generated so that each path is defined as a separate path, not as an alternate path. When this is done, each path can be attached or dedicated to a different user, and reserve/release CCWs issued by such users preserve data integrity. In this case, integrity is preserved by the hardware, not by the software reserve/release support. *Again, the definition of alternate paths in DMKRIO and the use of real reserve/release are mutually exclusive.*

- A volume may be defined through the directory to contain one or more minidisks. Such minidisks must be identified through the MDISK statement as requesting virtual reserve/release support. These minidisks may then be shared between virtual machines that support shared DASD and data integrity is preserved by the use of reserve/release CCWs in the virtual machine channel program. Alternate paths may be defined to the device when using virtual reserve/release. The reserve CCW is still changed to a sense CCW, but data integrity is preserved by the virtual reserve/release code.

No Dynamic Path Selection on the 3380

MVS is the only operating system that supports Dynamic Path Selection on the 3380. VM/SP HPO does not use this facility nor does VM/SP HPO allow an MVS virtual machine without preferred machine assist to use it. An MVS guest virtual machine with preferred machine assist, however, may use Dynamic Path Selection if it accesses the 3380-AA4 unit only across preferred channels (channels not defined in DMKRIO).

Missing Interruption Handler

Hardware errors, timing conditions, or software errors may prevent I/O interruptions from being passed to the control program. The missing interruption handler monitors system I/O activity to detect interruptions not occurring within specified time intervals. In order for CP to monitor I/O, module DMKDID must be present in the system. If MIH is set on and a missing interrupt is detected, CP will attempt to correct the conditions. CP sends an informational message to the operator and writes a record of the missing interruptions to the system error recording area regardless of whether the action is successful or not.

The default for the missing interrupt handler is that MIH is set off, the interrupt is detected and a message is written to the operator, but no corrective action is taken. MIH can be turned on by a directory option or by issuing the SET command.

Interruption timing varies widely among devices. Certain devices are more critical than others. In order to allow an installation greater flexibility for monitoring I/O activity, CP allows the installation to specify a different time interval for each class of device. The time interval is set using the

IBM-supplied defaults (provided in module DMKSYS), or as you determine. You may change the default intervals by coding the SYSMIH macro statement and reassembling DMKSYS.

The SET MITIME command dynamically changes the time intervals specified in DMKSYS. Time intervals used in the SET MITIME command remain in effect until you issue another SET MITIME command, or until you reload the system. A description of the SET MITIME command is in the *VM/SP HPO Operator's Guide*. See Chapter 23, "Preparing the CP System Control File (DMKSYS)" for a description of the SYSMIH macro statement.

You can use the missing interruption handler with all I/O devices supported by CP except terminal devices (CLASTERM), System Network Architecture devices, pass-through virtual machine (logical) devices, and special devices (CLASSPEC). The missing interruption handler supports Mass Storage System devices generated with a 3851 Mass Storage Controller (CLASSPEC TYP3851) and 3330V DASD (CLASDASD FEATURE=VIRTUAL or FEATURE=SYSVIRT).

Logical Device Support Facility

The Logical Device Support Facility allows an application executing in a virtual machine to communicate with CP as if it were a real terminal supported by CP. An example of such an application is the VM/Pass-Through Facility. The application manages data flow to and from CP. This function is implemented through the creation of logical devices by the facility in the host system.

The Logical Device Support Facility is made up of two data transfer subfunctions, three control subfunctions, a special external interrupt (code X'2402') to asynchronously alert a virtual machine of pending logical device status, and an external control word for passing control information with the external interrupt.

The facility is invoked by issuing the DIAGNOSE (X'7C') instruction. Operands on the DIAGNOSE instruction are used to specify:

- The register containing the logical device identification (Rx)
- The register identifying the subfunction to be performed (Ry)
- The DIAGNOSE function code (X'7C')

The data buffer address and the data buffer length are in register Rx and Rx+1 respectively when DIAGNOSE (X'7C') is issued for an ACCEPT or PRESENT function.

CP uses the external interrupt code X'2402' to signal asynchronously the virtual machine of pending logical device status. When the virtual machine receives the external interrupt, a full word of data is stored at location 128 (decimal) in virtual storage. This data gives the reason for the interruption and the associated logical device address. Control register 0, mask bit 22, must be on to receive this external interruption.

Subfunctions that may be requested via DIAGNOSE (X'7C') are summarized in Figure 26. More complete descriptions of these subfunctions are in the *VM/SP HPO System Programmer's Guide.*

Logical device support is not designed to simulate all aspects of real device support. Some instances are:

- Logical device support always passes channel end and device end to the virtual machine together.
- The Program Controlled Interruption (PCI) bit in the CCW is not handled by logical device support.
- Ending status on I/O only is passed back to the virtual machine (not initial).

Programming for Logical Device Support Facility is contained primarily in two pageable CP modules, DMKHPS and DMKHPT. If an application program does not invoke the Logical Device Support Facility, then modules DMKHPS and DMKHPT need not be included in the CP load list.

Subfunction	Description
INITIATE	Initiate CP/host application communications
ACCEPT	Transfer data written to logical device from virtual machine storage
PRESENT	Transfer data from virtual machine to CP as input from logical device
TERMINATE	Drop a specific logical device
TERMINATE ALL	Drop all logical devices created for this virtual machine

Figure 26. Logical Device Support Facility Subfunctions

Inter-User Communication Vehicle

The Inter-User Communication Vehicle (IUCV) provides a communication capability between virtual machines. The facility also supports communication within the same virtual machine and between a virtual machine and CP.

IUCV communication takes place between two communicators. Every communication has a source communicator and a target communicator. A communication occurs over a predefined linkage called a path. Messages are created, transmitted over the path, and then eliminated by IUCV. IUCV functions include the following:

- Communications paths and messages are initiated by either CP or a virtual machine.
- A communicator can selectively establish and terminate communication paths.

- Two communicators can establish more than one communication path between them.
- More than one message can be transmitted in either direction at the same time, using the same path.
- All IUCV functions are privileged.
- All IUCV functions are invoked with the IUCV macro.
- Directory authorizations allow an installation to control the establishment of IUCV communication paths between virtual machines and CP system services.

For a detailed description of IUCV functions and the IUCV macro instruction, refer to the *VM/SP HPO System Programmer's Guide*.

The IUCV directory control statement defines authorizations for establishment of IUCV communication paths. The MAXCONN keyword of the OPTION directory control statement defines the maximum number of IUCV connections allowed for a virtual machine. For more information, see the "The Directory Program" on page 190.

Although IUCV is similar to the Virtual Machine Communication Facility, IUCV does not replace VMCF. Both communication facilities are available and can coexist. IUCV should be considered for new applications requiring inter-user communication. IUCV is used for communication between System Network Architecture Console Communications Services (SNA CCS) and VCNA.

Virtual Machine Communication Facility

The Virtual Machine Communication Facility (VMCF) allows one virtual machine to communicate and exchange data with any other virtual machine operating under the same VM/SP HPO system. The VMCF external interruption masking is controlled by PSW bit 7 and CR0 bit 31. It is to your advantage to always have CR0 bit 31 set to 1 (while VMCF is in use) and to control the interruptions with PSW bit 7 only. This reduces the number of LCTL instructions.

Messages and data directed to other virtual machines are logically identified via the virtual machine's userid. Data is transferred in 2048-byte blocks from the sending virtual machine's storage to the receiving virtual machine's storage. The amount of data that can be moved in a single transfer is limited only by the storage sizes of the respective virtual machines.

Use of real storage is small. Only one real storage page need be locked during data transfer. A special interrupt is used to notify one virtual machine of a pending transfer of data. This interrupt is also used to synchronize sending and receiving of data.

Under the Special Message Facility, CP acts as a virtual machine in behalf of a virtual machine that issues SMSG. The receiving virtual machine, properly programmed to accept and process special messages, authorizes itself to CP. Data (message) transfer is from CP, via the message and VMCF modules.

Summary of VMCF Functions

VMCF includes five data transfer operations and seven control functions. Figure 27 contains a brief summary of VMCF functions. A more detailed description of these functions, including how they are implemented in the control program (CP), is in the *VM/SP HPO System Programmer's Guide.*

Data transfer operations involve movement of data from one virtual machine's storage to another virtual machine's storage. VMCF control functions allow a user to manage data transfer operations from the virtual machine console.

VMCF is implemented via the CP DIAGNOSE instruction. It is not hardware dependent. VMCF is available to any logged-on, authorized virtual machine.

Function	Code[1]	Comments
AUTHORIZE	Control	Initialize VMCF for a given virtual machine. Once AUTHORIZE is executed, the virtual machine can execute other VMCF functions and receive messages or requests from other users.
UNAUTHORIZE	Control	Terminates VMCF activity.
SEND	Data	Directs a message or block of data to another virtual machine.
SEND/RECV	Data	Provides the capability to send and receive data on a single transaction.
SENDX	Data	Directs data to another virtual machine on a faster but more restrictive protocol than the SEND function.
RECEIVE	Data	Allows you to accept selective messages or data sent via a SEND or SEND/RECV function.
CANCEL	Control	Cancels a message or data transfer directed to another user but not yet accepted by that user.
REPLY	Data	Allows you to direct data back to the originator of a SEND/RECV function, simulating full duplex communication.
QUIESCE	Control	Temporarily rejects further SEND, SENDX, SEND/RECV, or IDENTIFY requests from other users.
RESUME	Control	Resets the status set by the QUIESCE function and allows execution of subsequent requests from other users.
IDENTIFY	Control	Notifies another user that your virtual machine is available for VMCF communication.
REJECT	Control	Allows you to reject specific SEND or SEND/RECV requests pending for your virtual machine.

[1]Data indicates a data transfer function. Control indicates a VMCF control function.

Figure 27. Virtual Machine Communication Facility (VMCF) Functions

Extended Channel Support

Extended channel support permits you to increase the number of channels attached to a dyadic 3090 Processor Complex from 24 to 48. You can increase the number of channels in a channel set from 16 to 32. The larger number of channels minimizes channel contention and improves overall system performance.

Since extended channel support allows more than 16 channels in a channel set, channel addresses may have two digits. As a result, real addresses may have four digits: two for the channel, one for the control unit, and one for the device.

Chapter 11. Improving System Performance

The performance characteristics of an operating system when it is run in a virtual machine environment are difficult to predict. This unpredictability is a result of:

- The processor model
- The system configuration (uniprocessor, attached processor, or multiprocessor)
- The total number of virtual machines executing
- The type of work being done by each virtual machine
- The speed, capacity, and number of the paging media
- The use of fixed-head cylinders for preferred paging
- The use of Paging Storage for paging and/or swapping
- The amount of real storage available
- The degree of channel and control unit contention, as well as arm contention, affecting the paging medium
- The type and number of virtual machine performance options in use by one or more virtual machines
- The existence of hardware assists
- The favored priority and V=R options in effect
- The virtual machine's channel mode (block multiplexer or selector)
- The use of a cache for read-only data that is accessed by many users
- The order in which media are selected for paging, swapping, and spooling

Note: *The performance of an MSS being accessed by the operating system and shared with other systems depends on the total MSS utilization and contention.*

Performance Measurement and Analysis

The control program (CP) has two commands that measure system performance to help you identify problem areas.

The MONITOR command controls collection of performance data and the recording of it on system spool files or tapes. Both summary and trace data can be collected. You may specify classes of data collected using either the operands of the MONITOR command or the SYSMON macro instruction. Classes selected depend on the nature of the analysis to be performed. The Virtual Machine Monitor Analysis Program (VMMAP) Version I can be used to reduce the data collected. Guidelines for using this program provide you with information that will aid in determining overall load environment and performance profile of your system. The Virtual Machine Monitor Analysis Program (VMMAP) Version I should enable you to analyze utilization of and contention for major system resources such as the processor, storage, and I/O paging subsystems.

The INDICATE command displays, at a terminal, key information about the system, showing current performance indicators. Invoking INDICATE displays system conditions existing at the time the command is issued. This includes attached processor (AP) and multiprocessor (MP) utilization measurement when operating in an AP or MP environment. If, after using the INDICATE command, you want more extensive data collection and reduction, use the MONITOR command.

Specify automatic data collection with the SYSMON macro in DMKSYS. Coding considerations are in Chapter 23, "Preparing the CP System Control File (DMKSYS)." See the *VM/SP HPO System Programmer's Guide* for:

- Directions on using the MONITOR command to collect performance data on a dedicated tape drive or spool file
- The format and contents of the various classes of data collection available with MONITOR
- Details of the INDICATE command options

Note: The VM Real Time Monitor (SMART), listed in Appendix A, "Program Products, Installed User Programs, Field-Developed Programs, and Emulators," is another program designed to help you monitor and tune your system.

Using Performance Options

The performance of a specific virtual machine can be improved by assigning it one or more performance options. This section discusses those options. Topics are:

- Specifying a virtual=real machine
- Virtual machine assist
- Extended storage support
- VM/370 Extended Control-Program Support (ECPS:VM/370)
- Shadow table bypass assist
- MVS/System Extensions support
- MVS/System Product as a virtual machine
- Improving the performance of an MVS/SP virtual machine
- Shared segment protection
- Enhanced paging subsystem
- Enhanced performance for dyadic and dual processors
- Paging Storage

The *VM/SP HPO System Programmer's Guide* covers information about performance options.

Specifying a Virtual=Real Machine

A virtual=real machine runs in a special area of storage reserved exclusively for it. This area of storage is not available for use by CP or other virtual machines. CP provides no paging function for the virtual=real machine, which is why the virtual=real option is valuable for virtual machine operating systems that do their own paging.

Although the virtual=real option eliminates paging for the affected virtual machine, its main function is to bypass CCW translation. This is possible because I/O from a virtual machine occupying a virtual=real space contains a list of CCWs whose data addresses reflect the real storage addresses.

The only exception is virtual page 0. Except when running SP mode or with a PMA guest, virtual page 0 does not exist as real page 0; it is relocated to the highest page of the virtual=real area. Because of the relocation of page 0, CCW translation must remain on if the virtual machine performs I/O to page 0.

When CP loads an operating system into a virtual=real area, it turns on CCW translation. Once the operating system is loaded, the operator of the virtual machine may issue the CP command SET NOTRANS ON to turn CCW translation off.

When the virtual machine is operating with CCW translation off, it must not perform I/O into virtual page 0. Most operating systems can be generated so that they do not use this area for input/output. Violation of this restriction may cause damage to the entire VM/SP HPO system.

The size of the virtual=real area is specified during CP system generation. It must be large enough to contain the entire address space of the largest virtual machine that you execute in the virtual=real area. You must also be aware of the following requirements:

- Only one virtual=real area can be defined.
- Only one virtual machine at a time can occupy the virtual=real area.
- Since the virtual=real option removes pages from the dynamic paging area, it affects the performance of the other virtual machines.

The virtual=real area is set up at CP initial program load (IPL). It can be released by the primary system operator to be used as part of the dynamic paging area. Once released, it cannot be reclaimed except by reloading CP. The virtual=real area must be released in total. Unused pages of the area cannot be selected for release.

In processors using extended storage support (greater than 16M of real storage), the CP command UNLOCK V=R causes preferred machine assist V=R guest storage from RMSIZE or 16M (whichever is higher) to RSSIZE to be returned to CP for paging. (See "Preferred Machine Assist and Preferred Machine Assist with Control Switch Assist" on page 127 for further discussion of a preferred V=R guest above the 16-megabyte line.)

Each virtual machine logged on by CP requires some of CP's fixed free storage. If a very large virtual=real area is released after CP initialization, system performance degradation may occur as more and more users logon and use the released space. The reason for this is that the number of pages allocated for CP fixed free storage during system initialization is based on real machine size minus virtual=real size. As a result, the number of fixed free pages allocated for a system with a virtual=real area may not be enough to accommodate the larger number of users of the released space, and system overhead may increase as CP extends to get dynamic free storage pages.

This problem may be corrected by using the FREE operand in the SYSCOR macro instruction in the system control (DMKSYS) file at system generation. The SYSCOR macro is described in Chapter 23, "Preparing the CP System Control File (DMKSYS)." The examples used in the following discussions assume that you are allowing VM/SP HPO to determine the number of free storage pages to allocate.

To use the virtual=real option effectively on a multiport teleprocessing system with no CCW translation (SET NOTRANS ON), lines must be dedicated to that system via the ATTACH command or the DEDICATE control statement in the directory entry for the virtual machine. Conversely, on a multiport teleprocessing virtual=real operation, virtual 2701/2702/2703 lines, (that is, lines assigned and used by CP's DEFINE and DIAL commands) operate with CCW translation. If you issue the DIAL command while SET NOTRANS ON is in effect, CCW translation is done for I/O involving that line.

You cannot run programs with dynamic or self-modifying channel programs in a virtual=real area if you also use the DIAL command. Also, you cannot load (IPL) a shared system into a virtual machine running in the virtual=real area. For a virtual=real machine, you must issue the IPL command with either a device address or the name of a nonshared system.

To generate CP so that it properly supports a virtual=real area, do the following:

- Specify VIRT=REAL in the OPTION control statement in the directory entries for all virtual machines in your installation that you plan to run in the virtual=real area.
- Reserve enough DASD space for the CP nucleus.
- Reserve enough real storage space to contain the CP nucleus, virtual=real area, and other virtual machine requirements. Real storage space considerations are critical. If storage space requirements for the nucleus and virtual=real area exceed the size of real storage, the real IPL operation on a VM/SP HPO system supporting virtual storage preservation will result in a hardware load error.
- Specify the amount of storage you want reserved for a virtual=real area.

Chapter 19, "Creating Your Directory" describes the Directory program, including information about the VIRT=REAL operand of the OPTION control statement.

Note: For VM/SP HPO, extra DASD space is not required for a virtual=real system.

Real Storage Validation

VM/SP HPO automatically validates real storage on the 308x and 3090 processor by means of the TEST BLOCK hardware instruction. TEST BLOCK (TB) is an RRE format instruction (four bytes long) that has an operation code of X'B22C'. When CP is loaded or initialized on a 3081 processor, CP issues TB instructions to validate 4K blocks of real storage. This ensures that all page frames to be occupied by system modules are valid.

In the 308x and 3090 processors, real storage is not necessarily contiguous. One or more storage frames are used as a hardware system area to contain channel microcode, control blocks, and usage information. Since the hardware system area (HSA) is not addressable by the control program, an attempt to access the HSA causes an addressing exception. Thus, VM/SP HPO uses TB on the 308x and 3090 processors to detect noncontiguous blocks of real storage and recognize unusable storage frames.

At VM/SP HPO load, the loader uses the TB instruction as it relocates itself to the high end of storage and while loading the system modules into storage. The nucleus must reside in contiguous storage. If an unusable or non-addressable frame is detected within the area reserved for the nucleus, the system load is terminated with a disabled wait state code X'AAAAAA'. There is one exception. Non-addressable frames and frames having errors encountered in the virtual=real area do not cause a disabled wait state at load time. Instead, informational messages are sent to the system operator and the load continues. For this reason, virtual machine operating systems that execute in the V=R area on a real 3081 should use the TB instruction to validate their storage. When the V=R area is unlocked, VM/SP HPO automatically validates the area using TEST BLOCK.

When VM/SP HPO is initialized on a 308x or 3090 processor, those modules involved in the IPL process issue TB instructions to determine the status of every frame of real storage. In processors using extended storage support (greater than 16M of real storage), TB instructions are issued from the end of the V=R area until RMSIZE is reached, and then ISKE is used until RSSIZE is reached. DMKSYSRM, the amount of real storage, is then set to the address of the last good ISKE or TEST BLOCK. The amount of real storage available is defined as either the value specified as the RMSIZE operand in the SYSCOR macro or the real storage size indicated in DMKSYSRM, whichever is lower. The upper bound of storage CP can use is defined by either the value specified in the RSSIZE parameter of the SYSCOR macro or the real storage size found at initialization, whichever is lower.

If a nonaddressable frame, for example, at location 0 of real storage or the real storage area reserved for resident CP modules, or a frame containing errors is detected within the area reserved for the nucleus (excluding the V=R area), system initialization is terminated with a disabled wait state code X'14'. Storage frames reserved for the V=R area are not validated at initialization. V=R frames are validated only at load time as described earlier. In both cases, VM/SP HPO load and VM/SP HPO initialization, non-addressable or invalid frames encountered outside the nucleus area are identified to the system operator by a series of informational messages.

CP simulates TB for any virtual machine with EC mode capability regardless of real processor type, however, VM/SP, VM/SP HPO, and MVS/SP are the only operating systems that issue TB. When a virtual machine is executing V=V on a 308x or 3090, or on any other processor regardless of virtual machine mode, CP simulates all requested TB instructions by setting the storage block and storage key to zero, returning a condition code zero, and setting reference and change bits. A protection exception is reflected to a virtual machine that attempts to issue a TB instruction to a shared page. For a V=R virtual machine executing on a

real 308x or 3090, CP performs a real TEST BLOCK and reflects the result to the virtual machine. If the storage block is usable, the storage block is set to zero, the storage key is unpredictable, and condition code zero is returned. If the storage block is unusable, the storage block and storage key remain unchanged. A condition code one is returned.

V=R Virtual Machine Storage Requirements

When generating VM/SP HPO you have three constraints on the maximum virtual=real size you can specify: real storage, virtual storage, and the size of your nucleus.

Before you load the CP nucleus, be sure the virtual machine you are using has enough virtual storage to contain:

- The loader
- The CP nucleus (including the virtual=real area)

```
Loader + nucleus being loaded + V=R area = total storage requirement.
```

You must have an area larger than this total storage requirement to use the loader. If your virtual machine does not have enough virtual storage, redefine storage and IPL again before continuing.

Specifying the Virtual=Real Storage Size

If you are generating a system to include a virtual=real machine, in the system generation procedure you respond "yes" to the system message:

```
VIRTUAL=REAL OPTION REQUIRED (YES,NO):
```

You are then prompted to enter the size of the virtual=real machine size:

```
STORAGE SIZE OF VIRT=REAL (MINIMUM IS 32K):
```

Normally, you would not want to specify the largest virtual=real machine possible, since that would leave few page frames available for other virtual machines. For further planning and system generation considerations in using preferred machine assist and extended storage support with a virtual=real area, see "Preferred Machine Assist and Preferred Machine Assist with Control Switch Assist" on page 127.

At IPL time, the virtual=real area is locked in storage immediately following CP page 0. The system operator can issue the UNLOCK command with the VIRT=REAL option to free the virtual=real area for additional dynamic paging space for other virtual machines. The area cannot be relocked. It remains unlocked until another system IPL.

Calculate the maximum amount of virtual=real storage available on your processor as follows:

- Use Formula 1 below to calculate the amount of real storage above the minimum required by CP at IPL time. If available real storage (ARS) is negative or zero, CP will not IPL.
- Use Formula 2 below to calculate the maximum virtual=real size (VRS) for any real machine size. If VRS is negative or zero, a virtual=real area is not supported.

Calculating Available Real Storage (Formula 1)

Calculate available real storage (ARS) by subtracting the amount of storage required by CP from the real machine size. Formula 1 is:

$$ARS = RM - \left[I + T + 12K + F\right]$$

where:

RM
is the real machine storage size.

Note: In processors using Extended Storage Support, Formula 1 applies only to the area below 16 megabytes. RM in Formula 1 must be less than or equal to 16M.

I
is the storage needed to IPL CP. Refer to the load map produced when the CP nucleus is generated. The amount of storage needed to IPL CP is all of storage up to, and including, the module DMKSAV.

T
is the storage allocated for the CP internal trace table. CP allocates 4K of storage for each 256K of real storage up to 16M, and 4K of storage for each megabyte above 16M for the CP internal trace table:

$$4K\left\lceil\frac{RM}{256K}\right\rceil + 4K\left\lceil\frac{RMSIZE-16M}{1024K}\right\rceil$$

If the calculation RM/256K results in a fraction, the result should be rounded upward to the next higher integer.

If the calculation RMSIZE-16M results in a negative number, replace it with zero.

12K
is the fixed free storage allocated for the first 256K of real storage.

F
is the fixed free storage allocated for real storage beyond the first 256K (if there is no virtual=real area).

$$4K \left\lceil \frac{RM - 256K}{64K} \right\rceil + 4K \left\lceil \frac{RMSIZE-16M}{256K} \right\rceil$$

If the calculation enclosed in brackets results in a negative value, replace it with zero. If the same calculation results in a fractional number, disregard the fraction.

The result obtained from Formula 1 is the available real storage (ARS) for a particular real machine size. This result is needed to calculate the maximum size of a virtual=real area in Formula 2.

Calculating the Maximum Size of the Virtual=Real Area (Formula 2)

Calculate the maximum size of the virtual=real area for a particular real machine size by recalculating the real storage required by CP and subtracting that value from the real machine size. When you calculate the real storage required by CP, you do not permanently allocate free storage for the portion of storage that is available for the virtual=real area (according to Formula 1). The result of Formula 2 is the maximum size virtual=real area (VRS) you can specify for a particular real machine size. Formula 2 is:

$$VRS = RM - \left\lceil I + T + 12K + 4K \left\lceil \frac{RM - 256K - ARS}{64K} \right\rceil + 4K \left\lceil \frac{RMSIZE-16Mb}{256K} \right\rceil + 16K \right\rceil$$

Use the same value for RM, I, and T as you used in Formula 1. ARS (the available real storage) is the result calculated from Formula 1. If the calculation

$$\frac{RM - 256K - ARS}{64K} \quad \text{or} \quad RMSIZE - 16M$$

results in a negative value, replace it with zero. If the same calculation results in a fractional number, disregard the fraction (see Example 1). 16K is the storage needed at IPL time for the dynamic paging area. After VM/SP HPO is loaded (via IPL), the size of the dynamic paging area is the number of pages from DMKCPE to DMKSAV plus 16K.

The following table shows the maximum size virtual=real area you can specify for some real machine sizes.

Real Machine Size	Maximum VIRT=REAL Size
4M	2,914K
8M	6,925K
12M	10,932K

The values in this table assume that the value of I is equivalent to 1024K[5].

Example 1

Determine the maximum size virtual=real area for a real machine with 8192K (8M) of real storage. The system requires 1024K to IPL.

Formula 1:

$$ARS = 8192K - \left\lceil 1024K + 4K\left\lceil \frac{8192}{256K} \right\rceil + 12K + 4K\left\lceil \frac{8192 - 256K}{64K} \right\rceil \right\rceil$$

$$ARS = 8192K - 1024K + 4K \times 32 + 12K + 4K \times 124$$

$$ARS = 8192K - 1024K + 128K + 12K + 496K$$

$$ARS = 8192K - 1660K$$

$$ARS = 6532K$$

Formula 2:

$$VRS = 8192K - \left\lceil 1024K + 128K + 12K + 4K\left\lceil \frac{8192K - 256K - 6532K}{64K} \right\rceil + 16K \right\rceil$$

$$VRS = 8192K - \left\lceil 1164K + 4K\left\lceil \frac{1304K}{64K} \right\rceil + 16K \right\rceil$$

$$VRS = 8192K - 1267K$$

$$VRS = 6925K$$

Note that the fraction 1304/64 resulting from the calculation

$$\frac{RM - 256K - ARS}{64K}$$

in Formula 2 is rounded to the next lower integer.

---

[5] Since the amount of storage required to load VM/SP HPO varies with the inclusion of optional features and the number of devices in DMKRIO, this figure is used in the following example for illustrative purposes only.

Calculating DASD Space

To assess the relationship of DASD requirements to real storage space for saved systems on DASD, use the following formulas:

(program size/4)/32 = number of 2314/2319 cylinders
(program size/4)/57 = number of 3330/3333 cylinders
(program size/4)/24 = number of 2305/3340/3344 cylinders
(program size/4)/120 = number of 3350 cylinders
(program size/4)/96 = number of 3375 cylinders
(program size/4)/150 = number of 3380 cylinders
(program size/4) = number of FB-512 pages

Program size is the real storage size (in K bytes). K represents 1024 bytes.

Virtual Machine Assist

Virtual machine assist is a combination of a processor feature and programming. It improves the performance of VM/SP HPO. Virtual storage operating systems that run in problem state under control of CP use many privileged instructions and SVCs that cause interruptions which CP must handle. When virtual machine assist is used, many of these interruptions are intercepted and handled by the processor. Consequently, performance is improved.

Certain interruptions must be handled by CP. Consequently, virtual machine assist is not available if it

- Has an instruction address stop set
- Traces SVC and program interrupts

Since an address stop is recognized by an SVC interruption, CP must handle SVC interrupts while address stops are set. Whenever you issue the ADSTOP command, CP turns off the SVC handling portion of the assist feature for your virtual machine. The assist feature is turned on again after the instruction is encountered and the address stop removed.

Whenever a virtual machine issues a TRACE command with SVC, PRIV, BRANCH, INSTRUCT, or ALL operands, the virtual machine assist feature is turned off for that virtual machine. The assist feature is turned on again when tracing is completed.

If virtual machine assist is available on a processor, the operator, using the CP SET command, can turn the function off, and on again, for the entire VM/SP HPO system. Also, if the function is available to CP, each virtual machine operator can turn the function off, and on again, for his own virtual machine. When you create virtual machine directory entries, you can set off the SVC-handling portion of the virtual machine assist function for various virtual machines by specifying SVCOFF on the OPTION control statement.

Extended Storage Support

You can use extended storage support in VM/SP HPO to access storage above 16M in those processors that have extended storage capacity. Extended storage support improves performance in VM/SP HPO system paging rate, response time and throughput in systems experiencing heavy paging. Contention for CP pages in the dynamic paging area will be reduced and processor utilization will improve.

System generation considerations:

- To use Extended Storage Support, more than 16 megabytes of storage must be configured and on line. During system generation you should specify more than 16M in the RMSIZE or RSSIZE operand in the SYSCOR macro.
- CP treats real storage as two distinctly different entities, the area below 16 megabytes and the area above 16 megabytes.
- At IPL time, the CORTABLE entries that map free storage below the 16-megabyte line are initialized based on the following:
  - Allocating free storage pages according to real machine size up to 16 megabytes
  - The RMSIZE and FREE operands specified in the SYSCOR macro
  - Real storage above 16 megabytes
- CP allocates contiguous trace table pages based on the following:
  - Real storage above 16 megabytes
  - Real storage below 16 megabytes
  - The TRACE operand in the SYSCOR macro
- The CORTABLE entries for storage above 16 megabytes are initialized for a dynamic paging area.

VM/370 Extended Control-Program Support

VM/370 Extended Control-Program Support (ECPS:VM/370) is a hardware assist function that provides support over and above that provided by the virtual machine assist feature described previously, and consequently reduces CP's real supervisor state time needed to support virtual machines. ECPS:VM/370 provides the following functions:

- Expanded virtual machine assist
- CP assist
- Virtual interval timer assist

Whenever VM/SP HPO is loaded on one of the supported processors, all three hardware assist functions plus virtual machine assist are activated unless turned off by the system operator.

Expanded virtual machine assist includes a more comprehensive emulation of the SSM, LPSW, STNSM, and STOSM privileged instructions. Additional privileged instructions are also emulated.

CP assist provides a hardware assist for the high-use portions of the following CP functions:

- Virtual machine I/O
- Storage management
- Page management
- SVC handler
- Privileged instruction handler
- Dispatcher
- Standard DASD I/O (DIAGNOSE X'18')

The appropriate CP software routine is used if:

- CP assist is turned off
- Hardware assist does not support the specific service required
- An error condition occurs

Virtual interval timer assist provides for hardware updating of the location 80 interval timer for each virtual machine that has the virtual timer assist function turned on. This timer assist provides a more accurate and repeatable interval timer value for virtual machines than was previously possible through CP software.

Both virtual machine assist and expanded virtual machine assist are automatically turned off if you invoke certain TRACE functions. In addition, virtual interval timer assist is turned off if external interrupts are traced. When the tracing function is ended, CP automatically restarts these hardware assist functions.

For more details on VM/370 Extended Control-Program Support, refer to the *VM/SP HPO System Programmer's Guide.*

Shadow Table Bypass Assist

Shadow table bypass assist is a hardware assist that improves the performance of a virtual=real virtual machine. This feature assists several privileged instructions and one interruption.

To use shadow table bypass assist, issue the class G command SET STBYPASS VR.

MVS/System Extensions Support

The MVS/System Extensions support in CP enables an MVS system running in a virtual machine to use the MVS/System Extensions Program Product.

To use MVS/System Extensions support for your virtual machine:

- The hardware must be part of the real machine.
- The system operator must enter SET 370E ON for your VM/SP HPO system.
- The option 370E must be in the directory OPTION statement for your virtual machine, or you must enter SET 370E ON for your virtual machine.

Note: The SET 370E command is invalid when running VM under a VM system. Therefore, if you are attempting to run MVS in a virtual machine that is running under a VM system, do not set 370E on. Since MVS/SP requires 370E, you cannot run MVS/SP at third level.

When this support is enabled, an operating system running in your virtual machine can use these functions of the System/370 Extended Facility:

- Low address protection
- Common segment support
- Invalidate page table entry (IPTE) instruction
- Test protection (TPROT) instruction
- Virtual-machine extended-facility assist

MVS/System Product as a Virtual Machine

This section describes:

- Support required to run MVS/SP in a virtual machine
- Items that improve the performance of an MVS/SP virtual machine

MVS/System Product Support

The MVS/System Product support, which is part of CP, lets you run the MVS/System Product as a virtual machine operating system.

To use MVS/System Product support for your virtual machine:

- The hardware must be part of the real machine.
- The system operator must enter SET 370E ON for your VM/SP HPO system.
- The option 370E must be in the directory OPTION statement for your virtual machine, or you must enter SET 370E ON for your virtual machine.

Note: The SET 370E command is invalid when running VM under a VM system. Therefore, if you are attempting to run MVS on a VM which is running under a VM system, do not set 370E on. Since MVS/SP requires 370E, you cannot run MVS/SP at third level.

When the system and your virtual machine meet these requirements, the MVS/SP virtual machine can use these functions of the System/370 Extended Facility:

- Low address protection
- Common segment support
- Invalidate page table entry (IPTE) instruction
- Test protection (TPROT) instruction
- Virtual-machine extended-facility assist

Improving the Performance of an MVS/SP Virtual Machine

This section covers several items that improve the performance of an MVS/SP system running in a virtual machine. These items are:

- MVS/SP hardware-assisted cross memory services
- Single processor mode enhancements
- MVS/SP V=R virtual machine recovery
- Preferred machine assist
- Preferred machine assist with control switch assist

MVS/SP Hardware-Assisted Cross Memory Services

VM/SP HPO improves the performance of MVS/System Product (Release 3 or later) cross memory services for virtual machines. This support allows an MVS/SP virtual machine to use cross memory services for data movement, data access, and program sharing. The support also improves page fault processing for the MVS/SP virtual=real virtual machine.

To use cross memory services:

- Using the class A CP command SET 370E ON XMEM, the VM/SP HPO system operator must activate MVS cross memory services for system-wide availability or for availability on a specific processor. If MVS cross memory services are to be available on a specific processor of an attached processor or multiprocessor system, use the class A CP command SET 370E ON XMEM PROC addr (where addr is the address of the processor).
- The options 370E and XMEM should be in the OPTION statement in the directory entry for your MVS virtual machine. If 370E and XMEM are not in the OPTION statement of your directory entry, issue SET 370E ON XMEM.

  If MVS cross memory services are available on a specific processor of an attached processor or multiprocessor system, the AFFINITY option must appear in the OPTION control statement of your directory entry or the primary system operator must set affinity for your virtual machine (class A command SET AFFINITY userid nn).
- To improve page fault processing for the virtual=real virtual machine issue the class G CP command SET STBYPASS VR.
- For virtual=virtual machines, and virtual=real machines that have not eliminated shadow tables (through the command SET STBYPASS VR), use the class G command SET STMULTI to set the number of shadow tables available for the virtual machine. Cross memory services require a minimum of two (if you issue SET STMULTI 1 or SET STMULT1 OFF, CP overrides these options and provides two). The command SET STMULTI ON defaults to three. You can specify up to six shadow tables. The greater the number of shadow tables, the better will be the performance of the MVS virtual machine.

Single Processor Mode Enhancements

A description of single processor mode is in *Virtual Machine Running Guest Operating Systems*. To generate VM/SP HPO for single processor mode and to take advantage of the single processor mode enhancements:

- Be sure module DMKMPO is on the load list
- Generate VM/SP HPO with a virtual=real area

Note that CP rejects the command SET STBYPASS VR when issued in single processor mode. Even though you cannot invoke shadow table bypass, single processor mode uses a subset of the functions of shadow table bypass assist.

MVS/SP V=R Virtual Machine Recovery

V=R virtual machine recovery attempts to preserve an MVS/SP virtual=real virtual machine environment when CP abends. The virtual machine environment includes I/O control blocks, pending interruptions, dedicated devices, real storage, and single processor mode, if active. The recovery process turns off any options that the SET command turned on before the abend.

Note: Because MVS/SP does not support the use of the Vector Facility in 370 mode, V=R recovery does not save vector status or vector registers.

To use V=R virtual machine recovery:

- The module DMKVRR must be present in the system.
- The system resource operator (privilege class B) must issue the SET DUMP AUTO command.
- You (the virtual=real user) must be logged onto the system.
- After MVS is loaded into the virtual=real area, you must issue the class G CP command SET NOTRANS ON.
- You must be sure that all critical input and output occurs on dedicated devices, since the support recovers only dedicated devices.
- You must adjust any MVS parameters that control timing dependencies to prevent them from timing out while CP shuts down, dumps, and re-IPLs.

Preferred Machine Assist and Preferred Machine Assist with Control Switch Assist

An introduction to preferred machine assist and preferred machine assist with control switch assist is found in *VM/Running Guest Operating Systems*. The *VM/SP HPO System Programmer's Guide* discusses the impact of these features on CP commands, how CP interacts with a virtual machine using preferred machine assist or preferred machine assist with control switch assist, and performance considerations.

Information for planning and generating a system that uses preferred machine assist follows. Unless otherwise stated, "preferred machine assist" refers to both preferred machine assist and preferred machine assist with control switch assist.

Planning Considerations

- Any processor with preferred machine assist with control switch assist may also run preferred machine assist without control switch assist.

- The MVS system required for preferred machine assist depends on the processor. A 3090 processor requires MVS/SP Release 1.3.5, while other processors require an MVS/SP Release 1 Enhancement or later release. Note that if you want to use storage greater than 16 megabytes, you must use MVS/SP release 3 or a later release. Results are unpredictable if you load an improper version of MVS/SP as the preferred virtual machine.[6]

- Preferred machine assist does not support the MVS Program Product MSS Central Server Application Program. Preferred machine assist with control switch assist does.

- If you IPL your virtual machine with the PMA or PMAV option, you must also IPL the MVS Stand-Alone Dump Program with either the PMA or PMAV option. If you do not do this, the dump program cannot access:

  - Real storage above 16 megabytes

  - Any device that contains MVS paging data sets that are not on CP-owned channels

  IBM recommends that you IPL the MVS Stand-Alone Dump program with the PMAV option, because DIAGNOSE X'40' is supported in PMAV mode. DIAGNOSE X'40' allows CP to save and restore the contents of the virtual machine in which the IPL simulator runs, so that all of the preferred virtual machine's virtual storage can be dumped.

  You must also IPL the MVS Stand-Alone Dump with the PMA or PMAV option if it is used to dump all of real storage (V=R and CP storage). In this case, it must dump to a device on a preferred-machine-assist-only channel, that is, a channel that is *not* defined to CP in DMKRIO. If the MVS Stand-Alone Dump dumps to a shared device, that is, a device on a channel defined in DMKRIO, it will not be able to access CP real storage.

---

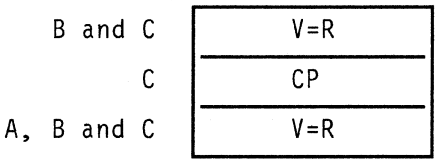
6 A preferred virtual machine is a virtual=real virtual machine that uses preferred machine assist or preferred machine assist with control switch assist.

The following diagram shows which areas will be dumped when the MVS Stand-Alone Dump is used in each of the three possible environments:

A V=R without the PMA or PMAV option

B V=R with the PMA or PMAV option, dumping to a shared device

C V=R with the PMA or PMAV option, dumping to a device on a preferred-machine-assist-only channel



- An MVS guest operating system using
  - Single processor mode
  - Nondisruptive transition

  on the 3090 processor must use preferred machine assist or preferred machine assist with control switch assist.

- An MVS guest operating system using
  - Preferred machine assist
  - Single processor mode
  - Nondisruptive transition

  on the 3090 processor must use the release of MVS/SP currently supported on the 3090 processor (Release 1.3.5).

VM/SP HPO System Generation Considerations

- The preferred virtual machine must run in the virtual=real area. The PMA option must *immediately* follow the VIRT=REAL option on the OPTION statement of the directory entry for the preferred virtual machine. The PMA and VIRT=REAL options must be on the same card-image.

- Use the largest virtual=real area possible for the preferred virtual machine. See "Specifying the Virtual=Real Storage Size" on page 117 to calculate the virtual=real area.

- For the most efficient use of storage in processors using extended storage support, allow an adequate balance of space for the V=R preferred virtual machine and dynamic paging area above 16 megabytes. The value specified in the RMSIZE operand of the SYSCOR macro will control this.

- When using extended storage support, the value specified in the RMSIZE operand of the SYSCOR macro determines the amount of storage above 16 megabytes used for the CP dynamic paging area and the preferred machine guest. If you specify RMSIZE as 16 megabytes or less, CP will not configure any dynamic paging area above the 16-megabyte line and will use less than 16 megabytes in the storage area below the 16-megabyte line. When running V=R, the entire area above the 16-megabyte line will be used for the PMA guest.

  If you unlock the V=R area, CP reclaims the V=R area from the value specified by RMSIZE to the value specified in the RSSIZE operand of the SYSCOR macro. This may leave unused storage area above 16 megabytes from RSSIZE to the end of real storage, and from RMSIZE to 16 megabytes in the area below 16 megabytes. For a discussion of storage maps illustrating Extended Storage Support, see the *VM/SP HPO System Programmer's Guide.*

- To maximize performance, leave some channels undefined in the real I/O configuration (DMKRIO) file. Leaving certain channels undefined allows the preferred virtual machine to define and exclusively own those channels. CP or other virtual machines cannot use those channels.

  Note: Any attempt to detach devices owned exclusively by the preferred virtual machine and then to attach those devices to CP will fail (an error occurs, since the device is not defined in the real I/O (DMKRIO) file).

- The module DMKPMA must be in the CP load list if the system is to use preferred machine assist.

- To ready the virtual machine for preferred machine assist:

  1. Set affinity to one processor, if you log onto VM/SP HPO in attached processor or multiprocessor mode. The AFFINITY option must be in the OPTION statement of your directory entry or the primary system operator must set affinity for your virtual machine (class A command SET AFFINITY userid nn).

     If VM/SP HPO is in uniprocessor mode when you log on, subsequent affinity settings are automatic. If, for instance, the system operator issues the CP class B command VARY ONLINE PROCESSOR nn, the preferred virtual machine's affinity is automatically set to the processor on which the machine was initialized.

2. Set on ECMODE. The option ECMODE must appear in the OPTION control statement in your directory entry or you must issue the class G CP command SET ECMODE ON.

- Observe the restrictions that apply to the LINK, ATTACH and DEFINE commands when preferred machine assist is active. The restrictions ensure that (1) if CP is running and if the preferred virtual machine issues an I/O instruction for a virtual device, that CP simulates I/O on the device defined in the real I/O (DMKRIO) file, and (2) if CP abends, and MVS gets control as a native system control program, that MVS does not inadvertently use a device defined in the real I/O (DMKRIO) file when it attempts to issue instructions to a virtual device.

  A LINK, ATTACH, or DEFINE command *cannot* give the preferred virtual machine:

  1. A virtual disk that is not a full-extent virtual disk (for this reason, a temporary disk (TDISK) cannot be specified)

  2. A dedicated device *or* full-extent virtual disk for which:

     - The virtual and real addresses are not the same *and* there is a real device control block (RDEVBLOK) for the virtual address specified on the command.

     - The virtual channel address does not match a channel defined in the DMKRIO file.

  3. A non-dedicated device for which:

     - There is a real device control block (RDEVBLOK) for the specified virtual address.

     - The virtual channel address does not match a channel defined in the DMKRIO file.

- Load MVS/SP by issuing the IPL command with the PMA or PMAV option (see *VM/SP HPO CP Command Reference for General Users*).

  If the device used for IPL is owned exclusively by the preferred virtual machine (the device is undefined in the DMKRIO file), then the device must be a count-key-data DASD, else results are unpredictable.

  If the device used for IPL is a virtual device, it cannot be a device which violates the restrictions on the LINK, ATTACH or DEFINE commands (see above).

Shared Segment Protection

VM/SP HPO uses one of the following to ensure that virtual machines do not alter shared segments:

- CP routines that protect shared segments
- The segment protection feature

The PROTECT = parameter on the NAMESYS macro instruction controls the use of the segment protection feature and CP's segment protection routines. On the 308x, 3090, and 4381-3, PROTECT = ON specifies that the hardware segment protection feature is to protect segments. On other processors, PROTECT = ON specifies that CP is to protect segments. To make either form of segment protection inactive, code PROTECT = OFF.

Note: In attached processor or multiprocessor mode without the segment protection feature, each processor has its own copy of every shared segment. CP obtains sufficient storage to construct duplicate page and swap tables. Plan for this additional storage space if your system is to be in attached processor or multiprocessor mode and the segment protection feature is not part of the hardware.

An installation that runs VM/SP HPO in attached processor or multiprocessor mode on a processor with hardware segment protect and that has many CMS users can reduce the size of free storage allocated at system initialization. Make this reduction by coding a lower number on the FREE = parameter of the SYSCOR macro instruction.

The SHRTABLE SHRPAGE pointer points to the page and swap tables for the main (IPL) processor. The page and swap tables for the attached (non-IPL) processor will be at a fixed offset from the page and swap tables for the IPL processor. DMKCFG initializes both sets of page and swap tables. At first, the swap tables for the IPL processor and non-IPL processor will point at the DASD locations specified in DMKSNT. As pages are read into storage and then stolen, each shared page is allocated its own DASD slot and pointed to by only one swap table entry. The last user to purge a shared system causes both sets of page and swap table to be freed. See the *VM/SP HPO System Programmer's Guide* for a description of shared segments.

Enhanced Paging Subsystem

The contiguous DASD allocation, in conjunction with the transfer of pages in groups (swapping), reduces the number of SIOs required to support paging operations. The enhanced paging subsystem is supported by the SYSPAG macro, the real storage interactive buffer, and some commands. A detailed description of the SYSPAG macro is provided in Chapter 23, "Preparing the CP System Control File (DMKSYS)." A description of the interactive buffer and monitor facility is provided in the *VM/SP HPO System Programmer's Guide.*

Paging Storage

Paging Storage, also called Expanded Storage, is a high-speed storage facility that serves as a paging assist for the 3090 Processor Complex. Paging Storage reduces system response time while providing increased system security, since only the control program has access to it. No system interfaces permit virtual machines to access the data in Paging Storage. Any virtual machine attempting to move a page into or out of Paging Storage will receive an operation exception.

Enhanced Performance for Dyadic and Dual Processors

This support improves the performance of VM/SP HPO when running on a dyadic or dual processor with store-in cache. Performance improvement is achieved by reducing the contention for cache between two processors. Refer to the *VM/SP HPO System Programmer's Guide* for a description of this enhancement.

Other Desirable Features

- The Extended Floating Point feature improves execution of programs that use Extended Floating Point instructions under VM/SP HPO on Models 155 II and 158.
  - For Model 155 II, feature #3700
  - For Model 158, feature #3700
- The ECPS Expansion Feature (#1601) is available on the IBM 4341 Processor Model Group 2. It increases performance when the MVS/System Product is run in conjunction with the VM/System Product. It allows concurrent operation of ECPS:MVS and ECPS:VM/370 and includes the functions of the shadow table bypass assist.
- Channel-to-channel Adapter (#1850) connects two channels (either S/360, S/370, or 4341 processor). Only one of the processors requires this feature.
- Data Streaming (#4850) is available on the 303x and 3042 AP-2 processors. This feature modifies the first two block multiplexer channels of a channel group to permit each to operate at a substantially higher data rate. This feature is standard on the 308x and 4341 processors.

Chapter 12. Attached Processors and Multiprocessor Systems

Attached Processors

To generate an attached processor system, use the control file and load lists shown in Figure 28 during the system generation procedure:

	System with no V=R area	System with V=R area
control file	DMKHnlA	DMKHnlA
load list	HnlAPLOD	HnlAVLOD
The letter n is the release level of VM/SP HPO and l is the modification level. If there is no modification level, l=0.		

Figure 28. Attached Processor Systems

During assembly of system definition files, DMKHnlA creates TXTAP object modules (rather than the usual TEXT) if attached processor support affects those modules.

Modules Containing AP Support

Seven modules exist exclusively for AP and MP support. Nucleus-resident modules are DMKLOK, and DMKMCT. Pageable modules are DMKAPI, DMKCLK, DMKCPO, DMKCPP, and DMKCPU. These modules have only an AP text file and their names are contained only in the AP and MP load lists (APLOAD and AVLOAD).

The modules that have TXTAP decks and are versioned for AP support can be found using the following steps:

1. List all the TXTAP decks off the VM/SP base tape.
2. List all the TXTAP decks off the latest VM/SP PUT tape.
3. List all the TXTHnlA decks off the VM/SP HPO product tape. (n is the release level of VM/SP HPO and l is the modification level. If there is no modification level, l=0).
4. List all the TXTHnlA decks off the latest VM/SP HPO PUT tape.
5. Combining the lists should produce a complete list of all TXTAP decks (all AP versioned modules).

Multiprocessors

To generate a multiprocessor system, use the control file and load lists shown in Figure 29 during the system generation procedure:

	System with no V=R area	System with V=R area
control file	DMKHnlM	DMKHnlM
load list	HnlAPLOD	HnlAVLOD
The letter n is the release level of VM/SP HPO, and l is the modification level. If there is no modification level, l=0.		

Figure 29. Multiprocessor Systems

During the assembling of system definition files, DMKHnlM creates TXTHnlM object modules (rather than the usual TEXT) if multiprocessor support affects those modules.

Modules Containing MP Support

DMKIOS and DMKMNL are the only modules used in MP support that are different from AP support modules. DMKSPM MACLIB contains OPTIONS COPY with the "&MP" variable set on. The DMKRIO sample supplied with the starter system contains a COPY OPTIONS statement following the CSECT. If you are using your own version of DMKRIO, be sure to include the COPY OPTIONS statement prior to assembly. DMKRIO must be assembled using the DMKHnlM control file, which will pick up the DMKSPM MACLIB. To assemble, use the command:

```
VMFASM DMKRIO DMKHnlM
```

nl is the release level of VM/SP HPO. To accommodate modification levels in release numbers, ignore decimal points and enter both numbers. For example, with VM/SP HPO Release 4.2, use 42 as the value of nl.

Chapter 13. Planning for 3270 Information Display System Terminals

In this section, "3270" refers to 3270 Information Display System Terminals. See Chapter 2, "Configurations" for a list of supported terminals.

You can attach terminals to the system as either local or remote. Local attachments do not require telecommunication lines. Remote attachments are attached to binary synchronous lines. Such configurations usually include:

- A designated channel
- A designated communication controller or transmission control unit
- The device or control unit (for terminal attachment and/or RJE systems)

This section does not include planning considerations for VTAM-supported terminals attached to a VTAM service machine. See Chapter 15, "Planning for SNA Console Communications Services."

Remote Terminal Attachments

3270 support includes remote cluster and stand-alone configurations. This support includes:

- Nonswitched point-to-point binary synchronous transmission
- Switched binary synchronous transmission for 3275 terminals equipped with the Dial feature only
- Cluster configurations of up to 32 display stations and/or printers
- The local 3270 copy function
- EBCDIC (Extended Binary Coded Decimal Interchange Code) transmission code only
- 3270s supported as virtual machine operator consoles
- CP commands allowing the operator to activate and deactivate the teleprocessing lines, display stations and printers
- CMS Editor and System Product Editor support
- The recording of MDR (Miscellaneous Data Recorder) records and OBR (Outboard Recording) records on the error recording cylinder. The MDR records are for the station and the OBR records are for the line. The CPEREP program edits and prints these records.

3270 copy support allows you to assign a screen copy function to a 3270 program function key. Pressing that key transfers the current display image, in its entirety, to an available printer attached to the same control unit. If the printer is busy or otherwise not available when the copy function is invoked, you receive a NOT ACCEPTED message in the screen's status area.

The following restrictions apply to VM/SP HPO's support of remote 3270s:

- Remote 3270 terminals cannot be used as primary or alternate VM/SP HPO system consoles.
- The number of binary synchronous lines supported for 3270 use is 256 minus the number of 3704/3705 Communication Controllers.
- Connections with multipoint clusters are not supported.

3270 Support on Binary Synchronous Lines

Supported display devices on binary synchronous lines have the same flexibility and usefulness as locally attached 3270 devices, except for the following limitations:

- *Display Information Inquiry and Retrieval Speed.* Because the 3270 remote stations are subject to relatively slow teleprocessing transmission speeds and to the mechanics of polling operations, screen display and data entry are not as rapid for remote 3270 devices as they are for locally attached 3270s.
- *Hard Copy of 3270 Screen Image.* Just as users of locally attached 3270 display devices can spool their virtual console input and output to the system printer, so can users of remote 3270 display devices. For remote 3270 users and local 3270 users whose terminals may be physically distant from the system printer, VM/SP HPO provides a limited hard-copy facility at the local and remote locations. The RSCS Networking program product provides support for printing spooled output.
- *TEST REQUEST and SYSTEM REQUEST Keys.* These keys are not supported for remote 3270s. The Test Request function on locally-attached 3277s is supported by the TEST REQUEST key. It is supported on locally-attached 3278s by the SYSTEM REQUEST key.

Remote Configurations Supported

Remote 3270 support requires:

- A binary synchronous line
- A transmission control unit
- Terminal devices (display stations and/or printers) and associated control units

The binary synchronous line must be in 2701/2703 mode. Transmission control units supporting remote 3270s on binary synchronous lines are:

- IBM 2701 Data Adapter Unit with Synchronous Data Adapter Type II
- IBM 2703 Transmission Control with Synchronous Data Adapter Type II
- IBM 3704 and 3705 Communications Controllers in emulation mode. A 3704/3705 line is in emulation mode when it is controlled by the Emulation Program (EP).

Cluster and stand-alone control units are supported for remote 3270s as follows:

Control Unit	Devices Supported
3271 Control Unit Model 2	Clusters of up to 32 display stations and/or printers
3274 Control Unit Model 1C, 21C, 31C	Clusters of up to 32 display stations and/or printers
3274 Control Unit Model 51C	Clusters of up to 12 display stations and/or printers
3276 Control Unit Models 2, 3, and 4	Clusters of up to 8 display stations and/or printers
3275 Display Station Model 2	A stand-alone 3270 device that can be remotely attached

Note: With a minimum configuration, the 3276 can also be considered a stand-alone 3270 device.

The following devices are supported when attached to the 3271 control unit:

- IBM 3277 Display Station Model 2
- IBM 3284 Printer Model 2
- IBM 3286 Printer Model 2
- IBM 3287 Printer Models 1 and 2
- IBM 3288 Line Printer Model 2.

Note: The 3271/3272 Attachment Feature #8330 is a prerequisite when the IBM 3287 Printer is attached to the 3271 control unit.

The following devices are supported when attached to the 3274 Control Unit Model 1C and 51C:

- IBM 3277 Display Station Model 2
- IBM 3278 Display Station Model 2, 3, 4, and 5
- IBM 3279 Color Display Station Models 2A, 2B, 2X, 3A, 3B, 3X, S2B, S2G
- IBM 3284 Printer Model 2
- IBM 3286 Printer Model 2
- IBM 3287 Printer Models 1, 1C, 2, and 2C
- IBM 3289 Printer Models 1 and 2

The following devices can be attached to the 3276[7] Control Unit Display Station Models 2, 3, and 4:

- IBM 3278 Display Station, Models 2, 3, and 4
- IBM 3279 Color Display Station, Models 2A, 2B, 2X, 3A[8], 3B[8], 3X[9], S3G[9], S2A, and S2B
- IBM 3287 Printer, Models 1, 1C, 2, and 2C
- IBM 3289 Printer, Models 1 and 2

The IBM 3275 Display Station, Model 2, is a stand-alone control unit and display station. You can attach the IBM 3284 Printer, Model 3, to the 3275. You can also attach the IBM 3286 Printer, Model 3, to the 3275 if RPQ MB4317 is installed. With the 3275 Dial Feature #3440, you can connect the 3275 to a processor over switched lines by using a Western Electric[10] or equivalent data set. VM/SP HPO does not support full screen read/write to the 3275 with the Dial feature.

System Generation Requirements for Remotely-Attached Terminals

When you generate VM/SP HPO, you must code appropriate CLUSTER, TERMINAL, and RDEVICE macros and assemble them as part of the DMKRIO (real I/O configuration) file to support 3270s for CP's use of virtual machine consoles. See Chapter 21, "Preparing the Real I/O Configuration File (DMKRIO)" for details of coding the DMKRIO file. Then, after DMKRIO assembles successfully, you must make a list of resource identification codes of all the remote 3270 lines and terminals.

---

7 The 3276 does not support extended color, highlighting, and programmed symbols for the 3278, 3279 and 3287.

8 3279 Models 3A and 3B cannot be attached to 3276 Model 2.

9 Not supported with the 3276 Model 2.

10 Trademark of Western Electric Co.

Give the list to the operations group at your installation. The members of that group need this information when they issue CP commands that control operation of remote 3270 lines and devices.

The RDEVICE macros must be coded in the same sequential order as the CLUSTER macros. (Refer to the example of a remote 3270 configuration, following.)

For a description of the CLUSTER, TERMINAL, and RDEVICE macros refer to Chapter 21, "Preparing the Real I/O Configuration File (DMKRIO)."

The CLUSTER Macro

Code one CLUSTER macro for:

- Each 3271 and 3274 control unit Model 1C, 21C, 31C
- Each 3276 control unit display station
- Each 3275 display station

A maximum of 256 CLUSTER macros is allowed. Each CLUSTER macro must have a unique label. The CLUSTER macro is described in Chapter 21, "Preparing the Real I/O Configuration File (DMKRIO)."

The TERMINAL Macro

Code one TERMINAL macro for each display station and printer attached to the clustered 3271.

Code one TERMINAL macro for each display station and printer that is attached to the clustered 3274 control unit. For a 3274 control unit TERMINAL macros for 3278s, 3279s, 3287s, 3289s on a Type A adapter must precede TERMINAL macros for 3277s, 3284s, 3286s, 3287s, and 3288s on a Type B adapter.

Code one TERMINAL macro for each display station and printer attached to a clustered 3276. Since the 3276 contains one integral display unit, code one TERMINAL macro for the 3276 as a 3278. Code each 3278 or 3279 attached to the 3276 as a 3278, and each 3287 attached to the 3276 as a 3284 or 3286.

Code one TERMINAL macro for the 3275 display station. If a 3284 or 3286 printer is attached to the 3275, code MODEL=3 on the TERMINAL macro. The TERMINAL macro is described in Chapter 21, "Preparing the Real I/O Configuration File (DMKRIO)."

The RDEVICE Macro

Code one RDEVICE macro for each binary synchronous line used for remote 3270s (one RDEVICE macro for each CLUSTER macro). A maximum of 256 RDEVICE macros for lines to support remote 3270s is allowed. The RDEVICE macro is described in Chapter 21, "Preparing the Real I/O Configuration File (DMKRIO)."

The Resource Identification Codes

After the real I/O configuration file (DMKRIO) assembles successfully, generate a list of the resource identification codes that correspond to each line address. Give the list to the operations group at your installation. The operator needs to know the resource identification code when he issues commands to control operation of remote 3270 lines and terminals.

The resource identification code is a four-digit hexadecimal code. The low-order two digits of the resource identification code are the resource address. The high-order two digits are the line code.

The resource address is generated by CP. The order in which TERMINAL macros appear in DMKRIO determines the resource addresses of the terminals defined. Each CLUSTER macro defines a 3270 control unit with a resource address of X'00' through X'FF'. The device defined by the first TERMINAL macro after the CLUSTER macro (in DMKRIO) has a resource address of X'01', the second has a resource address of X'02', up to the maximum of X'20' (since X'20' represents the decimal maximum of 32 devices on one control unit). This resource address is the two low-order digits of the resource identification code.

The line code is also generated by CP. Refer to the assembly listing for DMKRIO to determine the line code (the two high-order digits of the resource identification code). Locate the label DMKRIORN near the end of the DMKRIO assembly listing. This label identifies a list of all lines used by remote 3270s and by 3704/3705 Communications Controllers in EP mode. The two high-order digits are the line code and are assigned according to the order in which line addresses appear in the list. The first line address is assigned a line code 0 to complete its resource identification code, the second is assigned 1, and so on up to the last line. CP supports a maximum of 256 binary synchronous lines for use by remote 3270s. Thus, the maximum value of the high-order digit is F. Figure 30 shows you a sample DMKRIO assembly listing and the corresponding line codes. By looking at DMKRIORN in this example you can see that four lines have been generated. This simply means that the line codes for these four lines will be 00 – 03. If eight lines had been generated, the line codes would have been 00 – 07.

Sample of DMKRIO Assembly Listing			Line Code (in hexadecimal)
DMKRIORN	DC	F'4'	
	DC	AL2((RDV078-DMKRIODV)/8)	
	DC	XL2'078'	00
	DC	AL2((RDV07A-DMKRIODV)/8)	
	DC	XL2'07A'	01
	DC	AL2((RDV079-DMKRIODV)/8)	
	DC	XL2'079'	02
	DC	AL2((RDV07B-DMKRIODV)/8)	
	DC	XL2'07B'	03

Figure 30. Example of Determining Line Code for Remote 3270 Resource Identification Codes

Once you determine the resource identification codes for devices in your remote 3270 configuration, generate a list for operations. The list should include:

- Line address
- Line code
- Resource address
- Label of plug on control unit panel
- Resource Identification code
- Device type

Note: The plug panels of the 3271 control unit and 3274 control unit Model 1C have up to 32 ports where terminals and printers can be attached. The 3276 has up to 8 ports where the 3276 integrated display is attached and where up to 7 additional terminals or printers can be attached.

An Example of a Remote 3270 Configuration

The example below shows the contents of the DMKRIO to define:

- A clustered 3271 control unit with eight ports
- A standalone 3275 display station
- A second clustered 3271 control unit with eight ports

Macros are coded so that the 3271 clustered control unit can support eight display devices, or six display devices and two printers. To define such a configuration, you must code 2 CLUSTER, 16 TERMINAL, and 2 RDEVICE macros defining the 2 separate clusters. A 3275 stand-alone control unit, with one display and one printer, is also supported by the following macros.

To define it, you must code one CLUSTER, one TERMINAL, and one RDEVICE macro.

The real I/O configuration file for this example is:

```
DMKRIO   CSECT
CLUST078 CLUSTER  CUTYPE=3271,GPOLL=407F,LINE=078,DIAL=NO
         TERMINAL TERM=3277,SELECT=6040,FEATURE=OPRDR,MODEL=2
         TERMINAL TERM=3277,SELECT=60C1,FEATURE=OPRDR,MODEL=2
         TERMINAL TERM=3277,SELECT=60C2,FEATURE=OPRDR,MODEL=2
         TERMINAL TERM=3277,SELECT=60C3,FEATURE=OPRDR,MODEL=2
         TERMINAL TERM=3277,SELECT=60C4,FEATURE=OPRDR,MODEL=2
         TERMINAL TERM=3277,SELECT=60C5,FEATURE=OPRDR,MODEL=2
         TERMINAL TERM=3286,SELECT=60C6,MODEL=2
         TERMINAL TERM=3284,SELECT=60C7,MODEL=2
CLUST07A CLUSTER  CUTYPE=3275,GPOLL=407F,LINE=07A
         TERMINAL TERM=3275,SELECT=6040,FEATURE=OPRDR,MODEL=3
CLUST079 CLUSTER  CUTYPE=3271,GPOLL=407F,LINE=079
         TERMINAL TERM=3277,SELECT=6040,FEATURE=OPRDR,MODEL=2
         TERMINAL TERM=3277,SELECT=60C1,FEATURE=OPRDR,MODEL=2
         TERMINAL TERM=3277,SELECT=60C2,FEATURE=OPRDR,MODEL=2
         TERMINAL TERM=3277,SELECT=60C3,FEATURE=OPRDR,MODEL=2
         TERMINAL TERM=3277,SELECT=60C4,FEATURE=OPRDR,MODEL=2
         TERMINAL TERM=3277,SELECT=60C5,FEATURE=OPRDR,MODEL=2
         TERMINAL TERM=3277,SELECT=60C6,FEATURE=OPRDR,MODEL=2
         TERMINAL TERM=3277,SELECT=60C7,FEATURE=OPRDR,MODEL=2
         RDEVICE  ADDRESS=078,DEVTYPE=3705,ADAPTER=BSCA,     X
                  BASEADD=0B0,CLUSTER=CLUST078
         RDEVICE  ADDRESS=07A,DEVTYPE=3705,ADAPTER=BSCA,     X
                  BASEADD=0B0,CLUSTER=CLUST07A
         RDEVICE  ADDRESS=079,DEVTYPE=3705,ADAPTER=BSCA,     X
                  BASEADD=0B0,CLUSTER=CLUST079
```

In this configuration, if the 3271 cluster control unit is on line 078, there are six display devices and two printers supported. If the 3271 cluster control unit is on line 079, eight display devices and no printers are supported. Display devices can be interchanged among resource addresses allocated to display devices, and printers can be interchanged among resource addresses allocated to printers. A printer cannot be attached at an address defined for a display device, and vice versa.

Note: Cluster macros need not be coded in ascending numerical order. As in the example above, CLUST078 is coded, then CLUST07A, followed by CLUST079.

After the DMKRIO file assembles successfully, you should generate a list of resources for the operations group. Your list should be similar to the list shown in Figure 31. It corresponds to the preceding example.

Line Address	Line Code	Resource Address	Label of Plug in Control Unit Panel	Resource Identification Code	Device Type
078	00	00	--	0000	cluster
		01	0	0001	display
		02	1	0002	display
		03	2	0003	display
		04	3	0004	display
		05	4	0005	display
		06	5	0006	display
		07	6	0007	printer
		08	7	0008	printer
079	02	00	--	0200	cluster
		01	0	0201	display
		02	1	0202	display
		03	2	0203	display
		04	3	0204	display
		05	4	0205	display
		06	5	0206	display
		07	6	0207	display
		08	7	0208	display
07A	01	00	--	0100	cluster
		01	--	0101	display
		02	--	0102	printer

Note: The line code is determined by referring to Figure 30; which corresponds to this example.

Figure 31. Sample List of Resource Identification Codes for Operations

Local Attachments

Control units attached directly to the processor channels include:

- IBM 3272 Control Unit
- IBM 3274 Control Unit, Model 1B, 2B, 1D, 21D, and 31D

Display stations and printers attached directly to the control unit include:

- IBM 3277 Display Station
- IBM 3278 Display Station
- IBM 3279 Color Display Station
- IBM 3284 Printer
- IBM 3286 Printer
- IBM 3287 Printer
- IBM 3288 Printer
- IBM 3289 Printer (will not attach to a 3272)

Note: The printers listed above are supported for the PFnn copy function only.

The 3272 control unit can support:

- IBM 3277 Display Station
- IBM 3284 Printer
- IBM 3286 Printer
- IBM 3287 Printer
- IBM 3288 Printer

The 3274 control unit, Model 1B, 1D, 21B, 21D, 31D can support:

- IBM 3277 Display Station
- IBM 3278 Display Station
- IBM 3279 Color Display Station
- IBM 3284 Printer
- IBM 3286 Printer
- IBM 3287 Printer
- IBM 3288 Printer
- IBM 3289 Printer

Printers

The following printers are supported:

- IBM 3284 Printer, Model 2
- IBM 3286 Printer, Model 2
- IBM 3288 Printer, Model 2
- IBM 3287 Printer, Models 1 and 2
- IBM 3289 Printer, Models 1 and 2

System Generation Requirements for Locally-Supported Display Systems

System generation requirements for locally-supported terminals and control units are no different from the requirements for supported DASD or unit record devices. The channel, control unit, and devices are handled by their respective RCHANNEL, RCTLUNIT, and RDEVICE macros. See Chapter 21, "Preparing the Real I/O Configuration File (DMKRIO)."

Chapter 14. Planning for the 3704, 3705, and 3725 Communications Controllers

CP Support of the 3704, 3705, and 3725 Communications Controllers

CP supports the

- IBM 3704 Communications Controller, Models A1-A4
- IBM 3705-I Communications Controller, Models A1-D8
- IBM 3705-II Communications Controller, Models E1-H8
- IBM 3725 Communications Controller, Models 1 and 2

when attached to a processor. Three terminals are supported: 1050, 2741, and CPT-TWX 33/35. The 3767 terminal (operating as a 2741) is supported by lines in EP mode, and the 3101 and 3262 display terminals are supported as CPT-TWX 33/35.

The IBM 3704/3705 Communications Controllers can support:

- Up to 352 low speed start-stop lines
- Up to 60 medium speed synchronous lines
- Line speeds from 45.2 baud to 56.0K baud
- Modem capability within the 3704/3705
- Limited-distance "hard-wire" capability
- 16K to 256K internal storage
- Remote 3275, 3276, 3277, 3278, and 3279 terminals with optional 3284, 3286, 3277, 3287, 3288 and 3289 printers (EP mode only)
- Remote 2780 terminals (EP mode only)
- Emulator Program (EP) Version 3.0

CP's support of the 3704/3705 does not include:

- Remote 3704/3705 Communications Controllers

When planning for the installation of IBM 3704, 3705, and 3725 Communications Controllers, be sure that you are familiar with device characteristics, have the the appropriate publications and support package, and have a VM/SP HPO system that supports the 3704/3705/3725.

Note: In this publication, the term "3704/3705 control program" or "3704/3705/3725 control program" refers to the EP control program.

3704 and 3705 Support Package

Before you can generate a 3704/3705/3725 control program, you must have the following OS/VS Emulation Program Support Package. This is the only 3704/3705 support package that contains the CMS file required for generating and loading the 3704/3705/3725 control program under VM/SP HPO. The support package is:

- IBM 3704/3705 Emulation Support and System Support Package (EP/VS SCP) for OS/VS (order No. 5744-AN1). CP supports this package in emulation mode only.

This package contains the following basic material:

- A Program Directory
- *IBM 3704 and 3705 Communications Controllers, Emulation Program/VS Generation and Utilities, Guide and Reference Manual (for OS/VS and DOS/VS TCAM Users)*

  or

  IBM 3704 and 3705 Communications Controllers, Network Control Program/VS Generation and Utilities, Guide and Reference Manual (for OS/VS and DOS/VS VTAM Users)
- *IBM 3704 Control Panel Guide*
- *IBM 3705 Control Panel Guide*
- Machine-readable material:
  - Magnetic tape containing the macros and modules of the 3704/3705/3725 control program and the OS/VS system support programs.

Emulation Program (EP) with VM/SP HPO

The IBM 3704, 3705, and 3725 Communications Controllers are programmable units. The Emulation program can be generated to execute in 3704/3705 storage.

The minimum storage required by an EP control program is 16K.

There exist different versions of EP. CP can load EP (order number 5744-AN1) into a 3704 or 3705 communications controller. A special EP (order number 5735-XXB) was created for a 3705 so that ACF/NCP-SSP can provide loading and dumping facilities for this communications controller. This special EP can also be used for a 3725. CP provides no loading or dumping facilities for a 3725. These facilities are provided by ACF/NCP-SSP.

The Emulation program (EP) permits existing teleprocessing systems, including VM/SP HPO, that use the IBM 2701, 2702, or 2703 Transmission Control Units to execute without change on the 3704/3705.

The EP 3704/3705/3725 control program under VM/SP HPO:

- Emulates 2701, 2702, and 2703 operations
- Attaches to a System/370 byte multiplexer channel
- Supports up to 255 start-stop lines for 1050, 2741, and CPT-TWX (33/35) terminals
- Supports up to 50 medium-speed synchronous lines for 3270 and 2780 terminals
- Supports service programs and special CMS commands that allow you to easily generate the EP control program in a CMS virtual machine
- Supports the CP NETWORK command which allows you to load or dump the 3704/3705 and provides for automatic dumping and reloading if a fatal error occurs.

Planning Considerations

A 3704, 3705, or 3725 Communications Controller control program that runs under VM/SP HPO is normally generated after system generation is completed. When a 3704, 3705, or 3725 is to be generated, the following preparations must be made:

- An RDEVICE macro instruction for the 3704, 3705, or 3725 must be included in the real I/O configuration (DMKRIO) file. See Chapter 21, "Preparing the Real I/O Configuration File (DMKRIO)" for details of coding the RDEVICE macro for the 3704/3705/3725.
- 3704/3705 control programs to be used by VM/SP HPO must be stored on a CP-owned volume in the page format currently used for saved virtual machine systems (that is, those created by the SAVESYS command). Each 3704/3705 control program image to be saved must be defined by a NAMENCP macro instruction in the system name table (CP module DMKSNT), and saved with the SAVENCP command. This is not true for a 3725 communications controller or for a 3705 loaded by ACF/NCP-SSP.
- Enough space to contain the 3704/3705 control program image must be allocated on the CP-owned volume specified in the NAMENCP macro instruction. (Not applicable to 3725.)

Notes:

1. *The alternate system console for VM/SP HPO must not be on a telecommunication line on a real 3704/3705/3725, unless the 3704/3705/3725 is loaded by another operating system (OS/VS1, OS/VS2, or DOS/VS) before VM/SP HPO is loaded.*

2. *If you are using a 3705 with a VM/SP HPO system, whenever the system is shut down or abends, the logic used during system initialization for restart can cause a shared 3705 to ABEND.*

The *VM/SP HPO Installation Guide* contains a complete discussion on generating a 3704 or 3705 control program. It describes support provided with the Emulation Program (EP) and tells you how to generate the 3704/3705 control program, step by step.

Creating an Entry in the System Name Table

You must create an entry in the system name table (DMKSNT) for each unique 3704/3705 control program that you generate. If you can foresee generating several versions of the 3704/3705 control program, define extra entries in the system name table when you generate VM/SP HPO. In this way, you need not regenerate the VM/SP HPO system just to update the system name table. If you should have to regenerate the VM/SP HPO system just to add a new entry to the system name table, see the procedure in the *VM/SP HPO Installation Guide.*

The NAMENCP macro is described in Chapter 22, "Preparing the System Name Table File (DMKSNT)."

Reserving DASD space for the 3704/3705 Control Program Image

DASD space to contain the 3704/3705 control program image must be reserved on a CP-owned volume. The DASD space reserved should be sufficient to contain the number of pages specified in the SYSPGCT operand of the NAMENCP macro, plus one or more for system use.

If CPTYPE=EP, allow only one extra page.

These additional pages are used to store reference table information provided by the SAVENCP program.

Related Publications

The *Introduction to the IBM 3704 and 3705 Communications Controllers* describes the general functions of the 3704 and 3705. It is a prerequisite publication for generating a 3704/3705/3725 control program under VM/SP HPO.

If you are installing Version 4 of the Emulation Program/VS, the *IBM 3704 and 3705 Communications Controllers, Network Control Program/VS Generation and Utilities Guide and Reference Manual (for OS/VS and DOS/VS TCAM Users)* is a corequisite publication. If you are installing Version 4 of the Network Control Program/VS, the *IBM 3704 and 3705 Communications Controllers, Network Control Program/VS Generation and Utilities, Guide and Reference Manual (for OS/VS and DOS/VS VTAM Users)* is a corequisite publication. You must refer to one of these publications in order to code the 3704/3705/3725 control program generation macros. In this book this publication is called the *3704 and 3705 Generation and Utilities Guide.*

Chapter 15. Planning for SNA Console Communications Services

The Systems Network Architecture Console Communications Services (SNA CCS) provides a total data communication structure for transmitting information via a communications network. SNA communication products perform functions traditionally handled by the main processor, for example, management of communications lines, device dependent characteristics and control, and data formatting.

SNA CCS provides full console capabilities to terminal operators on SNA terminal devices. You can use SNA terminals as virtual machine consoles. The specific communication services and facilities used in exchanging information are transparent. If you are planning to use SNA CCS processing, you must consider the following topics.

Structure of the SNA Environment

Three major components contribute to SNA console support:

- SNA Console Communications Services (SNA CCS)
- Inter-User Communication Vehicle (IUCV)
- VTAM SNA Console Support (VSCS)

IUCV and SNA CCS are part of CP.

SNA virtual console support is provided through a virtual machine. The VTAM service machine (VSM) is the virtual machine that acts as an interface between SNA CCS and the SNA network.

The VTAM service machine is a machine that runs VSCS. VSCS works with the Advanced Communications Function/Virtual Telecommunications Access Method (ACF/VTAM) in doing its job.

Usually, ACF/VTAM is also in the VTAM service machine. VSCS, however, may be in a different virtual machine. ACF/VTAM manages the SNA network, while VSCS acts as the interface to CCS.

NCP and PEP Sharing

CP supports version 2.1 of the Network Control Program only. VTAM loads ACF/NCP. You must prevent CP from loading a back level of the NCP at initialization or restart. This can be accomplished in several ways. All methods depend on installation procedures.

One method is to initialize CP, but not enable lines until the VTAM service machine has been initialized. A similar technique could be used when the installation wishes to load the NCP before initializing the VSM.

The installation can initialize CP without enabling lines. Load the NCP using an installation-written CMS EXEC and then enable the lines. When the VSM is initialized, neither CP nor VTAM reloads.

Another method is to alter your system generation slightly by not specifying CPNAME= in the RDEVICE macro statement for the 370x device. This prevents automatic loading of the 370x at initialization via IPL. The VTAM service machine would then load the ACF/NCP.

In all cases, the 370x must be dedicated to the VSM. Loading and reloading of the 370x is controlled by the VSM.

The Partitioned Emulation Program (PEP) is shared by CP and the VTAM service machine. The 370x must be dedicated to the VSM. PEP is loaded by VTAM. Once the load is complete, EP lines are deactivated. The lines can then be enabled for use by CP. If the 370x is reloaded under control of the VSM, EP lines must be reactivated for CP users.

Tracing for SNA Console Communications Services

SNA CCS places a trace entry in the CP trace table for each inbound and outbound work transaction. The trace entry identifies the type of IUCV transmission, the SNA user, and the pertinent characteristics of the transaction. The transaction can be tracked throughout the system by use of the SNA CCS work element block, WEBLOK, the SNA CCS path ID, and the IUCV message ID. These fields can be matched with corresponding or similar fields in the IUCV trace elements in CP.

Normal Trace

SNA CCS creates trace table entries in the CP trace table, leaving an audit trail of its activities. This trace is invoked automatically. If you want to turn it off, set TRACE(9) to (0) in the LOCAL COPY control statement and reassemble the SNA modules.

Error Trace

In addition to the normal trace function described above, SNA CCS includes an error trace function. You can include the error trace independently of the normal trace. For error trace processing, SNA CCS places an entry in the CP trace table for logical errors and unexpected return codes from IUCV transmissions.

The error trace function is invoked automatically. If you don't want to use it, set the SNA CCS error trace bit off (X'40') in the CP trace flags, TRACFLG3, of the PSA (hex location 402).

Excluding SNA CCS Modules

If you do not want to use support provided by SNA CCS, you can eliminate the SNA CCS modules to conserve storage. You should delete the five SNA CCS modules (DMKVCP, DMKVCR, DMKVCT, DMKVCV, and DMKVCX) by excluding them from the load list. The size of the CP nucleus can be decreased further by eliminating the SNA routines in modules DMKQCN and DMKCPV. To eliminate the SNA routines in modules DMKQCN and DMKCPV, reassemble them with the LOCAL COPY control statement for SNA set off.

The format of the LOCAL COPY control statement to exclude SNA CCS processing is:

```
&SNAVCS SETB 0
```

Chapter 16. Planning for the 3800 Image Library

A 3800 image library that runs under VM/SP HPO is normally generated after the system generation is complete. When you intend to generate a 3800 image library, you must first:

- Include an RDEVICE macro instruction for the 3800 printer in the real I/O configuration (DMKRIO) file. See Chapter 21, "Preparing the Real I/O Configuration File (DMKRIO)" for details of coding the RDEVICE macro for a 3800.
- Store the 3800 image libraries to be used by VM/SP HPO on a CP-owned volume in the page format currently used for saved virtual machine systems (those created by the SAVESYS command). All 3800 image libraries are in the system name table (DMKSNT), and are saved with IMAGELIB or IMAGEMOD commands.
- Allocate enough space to contain the 3800 image library allocated on the CP-owned volume specified in the NAME3800 macro instruction.
- Decide which character sets, form control buffers (FCBs), and copy modifications will be used. These choices differ for 3800 Model 1 and Model 3 devices. Specify these elements through GENIMAGE control statements, as described in the *VM/SP HPO Operator's Guide.*

Creating and Updating a 3800 Named System

Establish a named system by means of the NAME3800 macro in DMKSNT for each system data set capable of activating an image library. The purpose of the named system is to contain 3800 character arrangement tables, copy modifications, graphic modifications, and FCBs. Before a file is printed on a 3800, the named system makes the name and data for that 3800 available to the file.

RDEVBLOK contains the active named system for a specific 3800. The named system can be changed by the START command. See the NAME3800 macro description in Chapter 22, "Preparing the System Name Table File (DMKSNT)."

Programs exist that allow you to dynamically change:

- Character arrangement tables
- Graphic modifications
- Copy modifications
- Available FCBs

With the system support discussed above and some special programs, you can make these changes during execution, without a system load. The special programs GENIMAGE, IMAGELIB, and IMAGEMOD are described in detail in the *VM/SP HPO Operator's Guide.*

Hardware Features Supported

The following hardware features are supported when the 3800 is used as a real spooling device:

- Automatic loading of character arrangement tables and graphic modifications
- Full support of the additional storage character generation feature
- Forms overlay feature (flashing)
- Copy modifications

The use of multiple character arrangement tables for printing use within one spool file (TRC support) is *not* supported. This feature, however, is supported through use of a virtual 3800.

Related Publications

Concepts of the IBM 3800 Printing Subsystem, GC20-1775, is intended for new users of printers who wish to take a quick look at the IBM 3800 Printing Subsystem, its basic concepts, and at how these concepts lead to new functions that may offer different options in planning and operations.

The *Reference Manual for the IBM 3800 Printing Subsystem*, GA26-1635, gives information on functions and features of the IBM 3800 Printing Subsystem relating to channel commands, sense bytes, and error detection, recovery, and recording. Specific information and examples are given for copy modification and control and graphic character modification.

The *IBM 3800 Printing Subsystem Programmer's Guide*, GC26-3846, gives planning and conversion information for the IBM 3800 Printing Subsystem and information on how to use the 3800.

The *Reference Manual for the IBM 3800 Printing Subsystem Model 3*, GA32-0050, gives information on functions and features specific to the IBM 3800 Printing Subsytem Model 3. Channel commands; sense bytes; and error detection, recovery, and recording are explained. Specific information and examples are given for copy modification and control and graphic character modification.

The *IBM Printing Subsystem Model 3 Programmer's Guide: Compatibility*, SH35-0051, gives planning and conversion information about the IBM 3800 Printing Subsystem Model 3 in compatibility mode.

The *VM/SP HPO System Programmer's Guide*, SC19-6224, discusses VM/SP HPO support of the 3800 Printing Subsystem.

Chapter 17. Planning for the 3850 Mass Storage System

The 3850 Mass Storage System (MSS) supplies large amounts of online data under system control. Up to 472 billion bytes of data space are available, allowing you to place significant amounts of tape and DASD shelf data under direct system control. Up to four virtual machines concurrently running OS/VS1, MVS, or SVS operating systems with MSS support can each control an interface to a common 3850 Mass Storage System.

Hardware Supported

Support for the 3850 is available on the following processors supported by VM/SP HPO:

> System/370 Models 155 II, 158 UP/AP/MP, 165 II, 168 UP/AP/MP, 303x, 3042 AP-2, 308x, 3090, 4341-3, 4381, 4381-2, and 4381-3 processors.

Major hardware components of MSS are:

- The 3851 Mass Storage Facility (MSF)
- The 3830 Model 3 Storage Control for System/370 Models 155 II, 158, 165 II, and 168 or the Integrated Storage Control for the System/370 Models 158 and 168
- The 3333 Disk Storage and Control (Models 1 or 11)
- 3330 Disk Storage Drives (Models 1, 2, or 11)
- 3350 Disk Storage Drives (Real Only)

Mass Storage Control (MSC) is a microprogrammed processor that provides operational control for components of Mass Storage System. It is housed in the 3851 Mass Storage Facility. MSC may have four System/370 channel interface positions, referred to as A, B, C, and D. A host system attaches to one of these through a control unit position of the byte multiplexer channel or block multiplexer channel operating in burst mode. The MSC channel interface is used for transfer of orders, commands, control information, and status messages between the host system and MSC. It does not carry user application data.

Up to four operating systems containing MSS support (OS/VS1, SVS, or MVS) may be connected to MSC. These operating systems may be running in a virtual machine under VM/SP HPO, or in a real processor, connected to the same MSC as VM/SP HPO. One of the four MSC interfaces is dedicated to each virtual machine. Each virtual machine using an MSC port reduces by one the number of other real processors that may be connected to the MSS.

The MSS uses the 3333 control unit and the 3330 Model 1, 2, or 11 for staging data and for holding tables it requires for its operation. These units connect to the Mass Storage Facility and to the processor through a staging adapter. Several models of the 3330 may be mixed on the staging adapter. 3330 disk drives can be one of the following:

- Real
- Staging
- Convertible

Real DASD drives are not available to the MSS for any activity. They are part of the system in that they have a data and control path through a staging adapter, but real drives are not logically connected to the MSS. Staging drives are used to hold data staged from mass storage volumes to be available for processing. Staging packs are divided into pages of storage. Each page consists of eight cylinders. The term virtual volume is used to refer to pages of space and the data staged to that space. Each virtual volume is assigned a virtual unit address. Staging drives are logically divided into staging drive groups to assist in the management of online space. Each staging drive must belong to one and only one staging drive group. There can be no more than two staging drive groups for each Staging Adapter. Each staging drive group can have a maximum of eight logical staging drives, a logical drive being the equivalent of one 3330 Model 1. One 3330 Model 11 counts as two logical staging drives.

Convertible drives can be real or staging drives, but not both at the same time. If the drive is to be real, the real path between the drive and the operating system must be available. When the drive is a staging drive, this real path must be offline.

Note: *Information describing MSS hardware can be found in* Introduction to the IBM 3850 Mass Storage System (MSS).

On a 3850 Mass Storage System, the Mass Storage Control can contain at most four channel interfaces to a single processor. The 3830 Model 3 Staging Adapter can have a maximum of four channel interfaces. The first channel interface on the 3830 Model 3 must be attached to a lower control unit position of the 3851 MSC. This control unit position does not conflict with the previously mentioned MSC port addresses. The remaining three channel interfaces of the 3830 may be attached to one or more host systems. Only the channels attached to the system being generated should be defined as primary or alternate channels.

For each of the three remaining (available) channel interface positions of a staging adapter, there are 64 possible device addresses. Thus, for each 3830 Model 3 control unit, or Integrated Storage Control with the staging adapter feature, there are 192 possible device addresses. Each device address corresponds to pages of staging space on the staging DASD. The staging space, which represents a volume, is allocated by MSC. Transfer of data between the staging space and the Mass Storage Facility, is also under control of MSC, which maintains the logical connection between a device address known to the host processor, the staging space allocated to the device, and the MSS volume mounted on the device.

When an MSS is connected to a VM/SP HPO system, the addresses known to CP are the MSC's channel interfaces and the device addresses to the channel interface positions on the Staging Adapter. MSC is supported in VM/SP HPO only as a dedicated device. For a virtual machine to access MSC, at least one of the MSC channel interfaces must be dedicated to the virtual machine.

In this publication, the device addresses corresponding to the channel interface positions on the staging adapter are referred to as 3330V device addresses. There are 64 3330V devices per channel interface position, or 192 3330Vs per staging adapter. There may be volumes mounted on all of these devices concurrently. These 3330V volumes represent 3330-1 volumes. With the proper programming support, they may be used for all purposes that a 3330-1 volume is used except VM/SP HPO system residence, paging, and spooling.

3330V devices may be used in three ways in VM/SP HPO:

- Mounted on the device and used as system volumes (excluding system residence, paging and spooling) under the control of CP
- Dedicated to a virtual machine as a 3330-1 and accessed from the virtual machine using standard 3330-1 support
- Dedicated to a virtual machine as a 3330V, in which case the virtual machine must contain MSS support.

A 3330V device address is not manually available to the VM/SP HPO system operator. Instead, it is an accumulation of pages of staging space on MSS staging DASD. Volumes are mounted on, and demounted from, 3330V devices only through orders passed to MSC. MSC is supported as a dedicated device under VM/SP HPO, and full MSC support is in OS/VS1 and MVS. Therefore, to mount and demount 3330V volumes for VM/SP HPO use, CP communicates with an OS/VS system to which an MSC channel interface is dedicated.

Any programming in a virtual machine that accesses a real 3330-1 can access a 3330V without modification. One or all CMS users may access CMS minidisks on MSS volumes. One MSS 3330V volume may contain minidisks for one or many CMS users. At the same time, virtual volumes may also be used as system residence packs for a VS system, and the VS system can be IPLed from the virtual volume.

The mounting and demounting of 3330V volumes used as VM/SP HPO system volumes is accomplished by CP communicating with an OS/VS system in a virtual machine. There is an MSS communication program named DMKMSS that is part of the VM/SP HPO system, but runs in problem program state in an OS/VS1 or MVS system. This DMKMSS program is the interface between the CP and MSC support in OS/VS. The steps to install DMKMSS in an OS/VS system are listed in the *VM/SP HPO Installation Guide.*

It is not necessary to generate a VS operating system specifically for the virtual machine environment. Any OS/VS1 or MVS system that supports MSS can utilize VM/SP HPO MSS support, and can act as host for the communicator program. There is, however, a requirement for MSS I/O devices in the VS system to match the definition of the virtual machine.

When OS/VS is IPLed, the system tests for any 3330Vs not online. When one is found, an order is issued to MSC for demount. In essence, the 3330V address is passed to MSC. The order tells MSC to demount any volumes currently mounted on that 3330V.

Although a 3330V may be offline to a virtual machine because none of VM/SP HPO's 3330Vs were allocated to the virtual machine at that virtual address, the 3330V may be a valid address to MSC. If the virtual machine issues a demount order to one of these 3330V devices, a volume in use by VM/SP HPO or another virtual machine MSC can be demounted. Therefore, the following rule must be used when defining (via IOGEN) 3330V devices in a VS system to run in a virtual machine to which an MSC interface is dedicated.

For each 3330V defined in the VS system there must be a corresponding 3330V defined to VM/SP HPO and allocated to the virtual machine.

For example, if you wish to dedicate real 3330Vs 240 through 27F to virtual CPUID 22222 as virtual devices 140 through 17F, then only 3330Vs 140-17F can be defined (via IOGEN) in the OS/VS system running in CPUID 22222.

Defining the MSS Communication Device

CP initiates an MSS mount or demount request by generating an attention interruption on a specified device. This device must be specified in the virtual machine directory as a unit record output device. For example:

```
SPOOL 017 2540 PUNCH
```

The same device address must be specified on the job control language used to start DMKMSS in VS. For example:

```
//MSSCOMM  DD UNIT=017
```

This device address must be constructed in VS at the same time as the IOGEN for the 3330Vs. The address chosen must not correspond to an actual device that VS will attempt to use for any other purpose. This is done by specifying the device as a DUMMY in the VS IOGEN. For example:

```
IODEVICE ADDRESS=017,UNIT=DUMMY,DEVTYPE=nnnnnnnn
```

The value of nnnnnnnn is any valid hexadecimal code. It is a VS requirement to provide a UNITNAME statement for this device. For example:

```
UNITNAME NAME=017,UNIT=017
```

The Mass Storage Control Tables

This topic is provided for installations that intend to run VS systems in a virtual machine, and access the MSS (under control of VS) from those systems. If you run only one VS virtual machine that has MSS support, and that virtual machine will access MSS only upon request from CP, then this section does not apply. You must, however, follow the guidelines in this topic if you have a virtual machine that has 3330Vs dedicated to it (that is, you intend to run more than one MSS virtual machine or to run VS MSS jobs in the MSS communication virtual machine).

MSC is driven from tables that reside on DASD. These tables are used, among other things, to define the MSS configuration. This configuration includes such items as addresses to be used for all components of the system, and available paths from all connected hosts to all these component devices. Thus, the MSC tables define the allowable paths from any host (as defined by that host's CPUID to a 3330V where the 3330V is defined in terms of the staging adapter address and the specific S/370 channel attachment to the staging adapter).

When a virtual machine is given access to MSS, one interface to MSC is dedicated to that virtual machine. To MSC, this is the same as having that interface connected to a native processor. The MSC tables must therefore be constructed so that MSC can process requests from the virtual machine. MSC must treat the requests as if they came from a native processor, controlling other components of MSS such that MSS activity, as seen by VM/SP HPO and the virtual machines, occurs on the correct 3330V device address.

Consider the example of a virtual machine that is given a virtual CPUID of 12345. This processor also has one of the MSC upper interfaces dedicated to it. Suppose that VM/SP HPO's 3330V 250 is dedicated to the virtual machine as virtual device address 150. When virtual CPUID 12345 issues an order to MSC, the 3330V placed in the order will be 150. When interruptions are generated for this 3330V, they will be sent from the Staging Adapter on the interface that corresponds to virtual CPUID 12345's 150. Since that device is known by VM/SP HPO as 250, the MSC tables must have been constructed such that the definition of 3330V 150 for virtual CPUID 12345 corresponds to the physical connection known to VM/SP HPO as 250.

Each 3330V in the MSC tables must map to a specific channel attachment on a specific Staging Adapter. In this case, the MSC table was constructed so that the definition for 3330V 150 on virtual CPUID 12345 corresponds to the physical connection from the real processor. This connection is through channel 2 to the same upper interface on the Staging Adapter. Thus, interruptions received from the virtual machine's 150 are received on VM/SP HPO's 250 as long as it is dedicated to the virtual machine corresponding to virtual CPUID 12345. Similarly, when the virtual machine issues an MSC order such as demount, the volume on VM/SP HPO's 250 is the volume demounted.

Two different virtual machines, having the same virtual device addresses can run concurrently under VM/SP HPO. If there are two virtual machines, each of which has defined a 3330V at the virtual machine's device address 150, then the MSC tables and the physical MSS configuration can be set so that each virtual machine can have a 3330V at address 150.

Example

One configuration has a native processor with two block multiplexer channels, Channel 1 and 2, and one Staging Adapter. Channel 1 is connected to the B interface of the Staging Adapter and Channel 2 is connected to the C interface of the Staging Adapter. The VM/SP HPO system has 3330Vs generated as 140 through 17F and 240 through 27F. Two virtual machines are defined as CPUID 11111 and CPUID 22222. Each of these machines can support an operating system in which the 3330Vs are generated at addresses 140 through 17F. The MSC tables for this configuration must show CPUID 11111 with its 3330Vs 140-17F mapped to the Staging Adapter interface B and CPUID 22222 with its 3330Vs 140-17F mapped to the Staging Adapter interface C.

Creating MSS Volumes

Before a pair of MSS data cartridges can be treated as a volume or accessed as VM/SP HPO system volumes, they must be initialized as the image of a 3330-1 disk pack. This initialization is accomplished by the use of an OS/VS Access Method Services command called CREATEV. CREATEV is one of several commands that are part of the MSS component of Access Method Services, which in turn is a standard component of OS/VS1 and OS/VS2. CREATEV can run either under VS on a native processor, or VS running in a virtual machine to which an MSC port has been dedicated. In either case, once CREATEV has completed, the volume is known to the MSS and may be referenced in MSC mount and demount orders.

Copying 3330-1 Volumes to 3330V Volumes

A full or partial 3330-1 volume may be copied to 3330V volumes. Once the MSS volumes have been initialized as described previously, with CREATEV, either of the following may be done:

- The access method services command CONVERTV may be executed from either a native processor or a VS virtual machine. This will make a bit by bit copy of the 3330-1 on the MSS 3330V.

- All or part of the 3330-1 volume and the 3330V volume can be allocated to a virtual machine by means of directory MDISK or DEDICATE statements or the operator ATTACH command. Standard CMS, OS, DOS, OS/VS and stand-alone utilities can then be used to copy data to the MSS volume.

Using 3330V Volumes for VS System Residence

A VS system can be loaded in a virtual machine from a 3330V volume because CP can make the virtual IPL device appear to be a 3330-1. The following steps describe one way this can be done:

1. Use the CREATEV command to create an MSS volume with a volume serial number of VOL001.

2. Define a directory entry for a virtual machine (VS2VM) with an MDISK statement, describing a minidisk spanning cylinders 1 through 401 on volume VOL001.

3. CP mounts VOL001 and allocates the minidisk when VS2VM logs on. The operator can then attach a 3330-1 containing a VS2 system to VS2VM.

4. Copy cylinders 0-400 of the 3330-1 to the minidisk within VS2VM.

5. IPL the virtual device address corresponding to the minidisk as a VS2 system residence device.

Part 2. Defining Your VM/SP HPO System

Part 2 describes macros and control statements you need to define your VM/SP HPO system. It contains:

- Introduction to System Definition
- Creating Your Directory
- Defining Your Own User Classes
- Preparing the Real I/O Configuration File (DMKRIO)
- Preparing the System Name Table File (DMKSNT)
- Preparing the CP System Control File (DMKSYS)
- Altering the Forms Control Buffer Load (DMKFCB)
- Altering the Universal Character Set or Font Offset Buffer
- Defining Saved Systems and Discontiguous Saved Segments

Chapter 18. Introduction to System Definition

System Definition Files

Before you start to generate a system on a real machine, you must create three files. These three files describe the VM/SP HPO system you plan to generate. At this time you may also choose to create two optional files.

Required Files

The three files that you *must* prepare are:

- Real I/O configuration file (DMKRIO), which defines the I/O configuration on the real machine
- CP system control file (DMKSYS), which defines the usage of CP-controlled DASD volumes
- VM/SP HPO directory file (normally a CMS file named USER DIRECT), which contains the directory entries that define the virtual machine configuration for each user.

Optional Files

The two other files that you may wish to create are:

- Forms control buffer load (DMKFCB), if you do not want to use the one supplied with the product tape
- System name table file (DMKSNT) if you plan to save systems[11]

To create these files, you can use:

- Files created with the System Product Editor
- Existing files changed with the System Product Editor
- Files from another VM/SP, VM/SP HPO, or VM/370 system on tape in dump format
- Card Input
- Other system files on tape in a card-to-card image

Use a CMS command such as READCARD, TAPE LOAD, MOVEFILE, or TAPPDS to bring files into the system.

---

11 If you generate the 3704/3705 control program, you must save it. If you don't, VM/SP HPO cannot load it.

System Integrity

System integrity allows only authorized users access to the operating system. The system is designed, utilized, and maintained to avoid the compromise of security controls. In other words, CP system integrity prevents illegal access to the system. Without the customer's knowledge and permission, no guest operating system or virtual machine should be able to:

- Circumvent or disable the Control Program main or secondary storage protection
- Access a Control Program (CP) password protected resource
- Use restricted passwords to access the system
- Obtain control in real supervisor state
- Obtain privilege class authority or directory capabilities greater than those it was assigned
- Circumvent the system integrity of any guest operating system such as MVS or VM/SP. (Operation of any CP facility implies system integrity for the guest.)

The following terms help describe the soundness of your system:

main storage protection: CP's isolation of one virtual machine from another by the use of hardware Dynamic Address Translation.

secondary storage protection: CP's isolation of minidisk and virtual disk extents via channel program translation.

password protected resource: CP's protection of resources via logon passwords and minidisk passwords.

directory capabilities: The selection of options to control functions normally reserved for specific assignment. Two examples are functions that permit bypassing normal system integrity controls and functions that are not normally granted to general users.

guest operating system: Another system control program that operates in a virtual machine under CP.

Note: VM system integrity applies to Class G users only.

MVS Guest Machine Environment

VM 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
- Preferred Machine Assist
- Single Processor Mode

A user or program on an MVS machine allowed to bypass system integrity controls can also bypass the system integrity controls built into VM/SP HPO. Under these conditions, the customer is solely resposible for any breach of security. The customer must make sure that the required MVS system integrity controls are installed and that authorized programs and users are properly controlled.

VM Control Program system integrity implies no protection of data between multiple users of a single CMS batch system.

Customer Responsibilities

Only the customer can answer for the security of the customer's data. For system integrity to be meaningful, proper use of security controls is essential.

Some areas where effective controls should be used are:

- Password protection
- Assignment of suitable privilege classes
- Assignment of directory options
- Set up and authorization of guest virtual machines

Specific actions and restrictions may vary, depending on system resources and conditions. The customer is responsible for selecting, applying, and implementing these actions and restrictions. Moreover, they must be adequate to ensure the customer's complete control of the system.

Note: IBM will accept APARs that describe exposures to the system integrity of VM/SP HPO. For instance, problems caused by a program running in a virtual machine not authorized by a mechanism under the customer's control can expose the system integrity.

Chapter 19. Creating Your Directory

The directory contains the entries of all potential virtual machines permitted to log on the VM/SP HPO system. Without the proper directory entry, a user cannot log on. User entries in the directory contain:

- User identification and password
- Virtual machine I/O configuration
- Associated virtual and real addresses
- Disk usage values
- Virtual processor storage size
- Other options

There must be at least one CONSOLE or SPOOL directory entry for each user, except those whose password is NOLOG. These options are discussed in the directory program control statement descriptions.

The directory usually resides on the CP system residence disk, and is pointed to by the VOL1 label. On a count-key-data DASD, this is at cylinder 0, track 0, record 3. On an FB-512 DASD, it is at absolute block 5.

The size of the directory was formerly limited to 10.8 K users. It is now limited only by the space available.

The directory program (module DMKDIR, invoked by the DIRECT command, or run standalone) processes your control statements and writes the directory on disk. You describe your installation's real configuration when you create the real I/O configuration file. Now, you describe the many virtual configurations for your installation with directory program control statements.

To create a directory:

1. Prepare directory program control statements.
2. Sort the directory by userid.
3. Format and allocate DASD space to contain the directory.
4. Change restricted passwords to installation-specific passwords.
5. Run the directory program in EDIT mode.
6. Correct errors.
7. Rerun the directory program.

At this time, prepare the directory program control statements and sort the directory by userid. Though not necessary, sorting the directory is recommended. Sorting the directory permits a search to go to the right page immediately instead of reading every page until the desired userid is found. During system generation, you must:

- Format and allocate DASD space for the directory
- Generate the directory

Considerations for Preparing the Directory Control Statements

General Considerations

First, prepare a directory control statement that defines the device on which the directory is to be written. This statement (DIRECTORY) must be the first control statement in the directory program. It is followed by the sets of statements describing your installation's virtual machines.

Next, prepare directory program control statements describing each virtual machine. The descriptions contain accounting data, options, and virtual machine configurations for each virtual machine that appears in the directory. Information about coding these control statements is found in "The Directory Program" on page 184.

If you are defining your own user classes for CP commands, refer to Chapter 20, "Defining Your Own User Classes" on page 237 and the *VM/SP HPO System Programmer's Guide* to help you prepare the appropriate statements.

Note: As the directory source file is processed, CP checks those statements that describe virtual devices for a subchannel protocol conflict. If a conflict is detected, CP sends an error message to the user. Only the effects of the T-DISK option of the MDISK statement and the CONSOLE, SPECIAL, and SPOOL statements are considered. The effects of the DEDICATE, LINK, and MDISK statements depend on the real device configuration at logon time.

CP does not check for overlapping extents. You must therefore ensure that minidisk extents defined in the directory do not overlap each other and (in the case of 3330, 3340, and 3350 disks) do not overlap the "alternate track" cylinders. If they do, file data damage is inevitable.

Using Directory Profiles

A directory profile is a set of statements used in defining users' virtual machines. Instead of coding the same statements over and over in many user entries, you can define one directory profile and include it in each user entry that needs the set of statements.

To define a directory profile, you must code a profile entry in the directory. A profile entry begins with a PROFILE control statement and may include any directory control statements except DIRECTORY, PROFILE, INCLUDE, USER, and MDISK. (See "PROFILE Control Statement" on page 191.)

To include a directory profile in a user's directory entry, code an INCLUDE control statement. (See "INCLUDE Control Statement" on page 195.) This has the same effect as coding all the statements in the directory profile.

Suppose that all the users in Department 46I need many of the same control statements in their directory entries. You code the following profile entry to define a directory profile called DEPT46I:

```
PROFILE DEPT46I
    OPTION REALTIMER
    ACCOUNT 46I05K 46IX0253
    ACIGROUP DEPT46I
    IPL CMS PARM AUTOCR
    CONSOLE 009 3210
    SPOOL   00C 2540 READER *
    SPOOL   00D 2540 PUNCH B
    SPOOL   00E 1403 6 PRINT25
    SPECIAL 0FF TIMER
    LINK    ANNSMITH    190 190 RR
    LINK    ANNSMITH    19A 19A RR
    LINK    ANNSMITH    49B 49B RR
    LINK    SYSTOOLS    395 399 RR
```

When you code the directory entries for each user in Department 46I, you don't have to repeat the thirteen control statements in the directory profile. Each user's directory entry can be quite short, for example:

```
USER JONDOEA XXXXXXX 02048K 4M G
    INCLUDE DEPT46I
    MDISK 191 3350 430 005 MADRD3 M

USER JONDOEB XXXXXXX 02048K 4M G
    INCLUDE DEPT46I
    OPTION ECMODE
    ACCOUNT 46I05K BLDG482
    MDISK 191 3350 430 005 MADRD3 M
```

By using a directory profile, you have saved time and storage space.

Notice that the OPTION and ACCOUNT statements in JONDOEB's directory entry contradict statements in the directory profile. Control statements in an individual user entry cause the corresponding statements in a directory profile to be ignored. JONDOEB has the ECMODE option but not the REALTIMER option.

Security Considerations

Access Control Groups

If your installation has the Resource Access Control Facility (RACF, #5740-XXH) and RACF/VM Support PRPQ (#5767-002), you can use the ACIGROUP control statement to determine which users are allowed to:

- Log on to the system
- Link to a given minidisk
- Transfer or spool data to a given userid
- Use the TAG command to send data to a given nodeid

When a user issues the LOGON or AUTOLOG command, a password verification routine checks to see if that user belongs to an authorized access control group. When a user issues the LINK, SPOOL, TRANSFER, or TAG command, an access verification routine does the same thing. If the user's directory entry does not contain an ACIGROUP statement with the right groupname, the command fails. The LOGON, AUTOLOG, and LINK commands fail even if the unauthorized user knows the logon or minidisk password.

You must modify RACF's password and access verification routines to indicate which access control groups are authorized. See the *VM/SP HPO System Programmer's Guide* to do this.

See "ACIGROUP Control Statement" on page 198 for instructions on coding the ACIGROUP directory control statement.

Passwords That Don't Work

For every virtual machine you create, you must define a password on the USER control statement. When the directory program is run under CMS, it checks all the passwords against the list of restricted passwords in a file called RPWLIST DATA. If any password is found in the RPWLIST DATA file, it is changed to NOLOG. The user will not be able to log on until you change NOLOG to a good password.

Note: The stand-alone directory program does not check for restricted passwords.

The list of restricted passwords supplied by IBM includes all the sample passwords published in IBM documents:

AUTOLOG1	BATCH	CE
CMSUSER	CPCMS	DIRM
ECMODE	GCS	IBMCE
IPCS	IVPASS	MAINT
MASTER	OPASS	OPERATNS
OPERATOR	PASSWORD	PRODBM
PROMAIL	RSCS	SFBATCH
SFCAL	SQLDBAPW	SQLUSER
SYSADMIN	SYSDUMP	

You can add your own passwords to this list, or delete passwords from it. You will probably want to restrict the use of such obvious passwords as your company's name and of any password that you think unauthorized persons may know.

To change the RPWLIST DATA file, use the System Product Editor. Enter one restricted password per record. Each record must have a fixed length of at least eight bytes and must be in this format:

Columns 1—8	Column 9	Columns 10—End of Record
PASSWORD	Blank	Comments

If you rename or erase the RPWLIST DATA file, passwords will not be checked.

See the *VM/SP HPO System Programmer's Guide* for more information.

Virtual Machines You Must Define

Virtual Machines for the Operator and System Analyst

You must define one or more virtual machines for the operator. You should define virtual machines for the system analyst or system programmer.

The operator's virtual machines control:

- The VM/SP HPO sessions
- Allocation of machine resources
- Spooling activity
- Online disk areas

Also define virtual machines for system analysts that

- Perform system analysis
- Modify certain VM/SP HPO functions

and virtual machines to update or operate:

- The CP system
- The CMS system
- The RSCS system, if you generate one
- The hardware
- Other operating systems that run in the virtual machine environment
- The Installation Verification Procedure

System Support Virtual Machines

At system generation, you should create two additional virtual machines beyond those needed by VM/SP HPO users. These machines include one each for hardware and software support. Consult IBM National Service Division (NSD) programming support when you determine configurations for these virtual machines.

Hardware Support: The hardware support is for:

- The processor, which must be supported in a dedicated environment. This is because there is no method available that allows concurrent support of the processor, real storage, and channels when executing problem programs.

- Input/output equipment, which can be supported using online test (OLT) under OLTSEP. The OLTSEP program can be executed in its own virtual machine.

Any offline testing capabilities of system devices can be used on inactive units while the system is operating.

To perform online hardware support, a virtual machine must be defined in the directory for the IBM service representative. The virtual machine should have enough virtual storage defined to execute OLTSEP. Normally, the service representative requires that the device being tested be dedicated to his virtual machine. The system operator can dedicate devices to a virtual machine by issuing the ATTACH command.

The virtual machine for hardware support should have the minimum configuration required to run online tests, and provide access to CMS with a read/write minidisk. Privilege class F should be assigned to allow the hardware diagnostics to be run, and error recording and retrieval facilities to be used.

The hardware service representative's virtual machine should also have access to CMS and to the error recording area of the system residence volume. An EREP program (CPEREP) runs under CMS allowing editing and printing of all recorded machine check and channel check errors. This directory entry is included in the directory provided with the starter system.

Software Support: The virtual machine for software support should have the sufficient system resources allocated to recreate (virtually) problems that occur on the real machine and to perform software service activities. The ECMODE option must be specified in the directory OPTION control statement for this machine. You should assign privilege classes to the user of this machine.

Sample Directory Entries

The following sample directory entries provide an installation with some of the entries necessary for operation and updating. The indentations are for readability and are not required by the directory program. LINK control statements are used whenever possible to minimize the number of changes to the directory whenever a minidisk extent is moved. A brief explanation of some of the virtual machine userids follows.

Warning: The passwords in these sample entries are restricted. You must substitute your own passwords. If you copy any sample password, it will be changed to NOLOG when the directory program is run. That virtual machine will not be able to log on.

For more information, see "Passwords That Don't Work" on page 176.

The System Operator's Virtual Machine (OPERATOR)

The userid for this directory entry must be the same as the userid on the SYSOPER operand of the SYSOPR system generation macro in the DMKSYS file. The USER control statement gives the operator all command privilege classes except class F. Actually, if other virtual machines are defined with command privilege classes appropriate for updating VM/SP HPO, the operator's virtual machine needs only class A command privileges. The MDISK control statement defines the 191 minidisk, which contains CMS files, EXEC procedures, and service programs to update VM/SP HPO.

```
USER OPERATOR OPERATOR 3M 16M ABCDEG
     ACCOUNT 2 OPERATOR
          CONSOLE 009 3215 T MAINT
          SPOOL 00C 2540 READER *
          SPOOL 00D 2540 PUNCH  A
          SPOOL 00E 1403 A
          LINK MAINT 190 190 RR
          LINK MAINT 19D 19D RR
          MDISK 191 3330 304 002 VMSRES MR ROPER WOPER MOPER
```

A Virtual Machine to Receive System Dumps (OPERATNS)

The userid for the following directory entry is the userid specified on the SYSDUMP operand of the SYSOPR macro when the VM/SP HPO system was generated. All abnormal termination dumps are sent to this virtual machine. This user normally is given command privilege classes A, B, C, and E. If the directory entry contains all disks normally attached to the system, described as full-volume minidisks, you can rewrite the directory, using the DIRECT command. The operations group can examine any disk while it is attached to the system, when these disks are defined as full-volume minidisks.

```
USER OPERATNS IPCS 512K 1M BCEG
     ACCOUNT 1 SYS PROG
          IPL CMS
          CONSOLE 009 3215
          SPOOL 00C 2540 READER D
          SPOOL 00D 2540 PUNCH  A
          SPOOL 00E 1403 A
          LINK MAINT 190 190 RR
          LINK MAINT 19D 19D RR
          LINK MAINT 19E 19E RR
          MDISK 191 3330 302 001 VMSRES MR RIPCS WIPCS MIPCS
          MDISK 193 3330 309 016 VMSRES MR RIPCS WIPCS MIPCS
```

Some installations may want to combine the functions of OPERATOR and OPERATNS into one virtual machine. This can be accomplished in the above examples by adding the last 8 control statements of the directory entry for OPERATNS to the directory entry for OPERATOR. The OPERATOR virtual machine can then perform all the system functions required to operate the VM/SP HPO system.

Other System Virtual Machines

In addition to virtual machines discussed to this point, there are virtual machines that:

- Support and update the VM/SP HPO system
- Test new releases of the system before placing them into production status
- Provide the hardware service representative with the ability to run diagnostics and extract recorded error data
- Provide other users with a remote file spooling capability

A Virtual Machine for Updating and Supporting VM/SP HPO (MAINT)

The following directory entry defines a virtual machine (MAINT) that can support and update the VM/SP HPO system. The MAINT virtual machine's command privilege classes include class E and class G, so it can issue the SAVESYS command to save CMS and other systems.

```
USER MAINT CPCMS 6M 16M ABCDEFG
 ACCOUNT 1 SYSPROG
 IPL 190
 CONSOLE 009 3215
 SPOOL 00C 2540 READER *
 SPOOL 00D 2540 PUNCH  A
 SPOOL 00E 1403 A
 MDISK 123 3330 000 404 VMSRES MW RSYSRES WSYSRES MSYSRES
 MDISK 124 3330 000 404 VMSEXT MW RSYSRES WSYSRES MSYSRES
 MDISK 125 3330 000 404 VMPK01 MW RSYSRES WSYSRES MSYSRES
 MDISK 126 3330 000 404 VMSTGE MW RSYSRES WSYSRES MSYSRES
 MDISK 127 3330 000 404 VMSTG2 MW RSYSRES WSYSRES MSYSRES
 MDISK 128 3330 000 404 PROFPK MW RSYSRES WSYSRES MSYSRES
 MDISK 130 3330 000 404 VMGCS1 MW RSYSRES WSYSRES MSYSRES
 MDISK 131 3330 000 404 SQLPK  MW RSYSRES WSYSRES MSYSRES
 MDISK 132 3330 000 404 SQLPK2 MW RSYSRES WSYSRES MSYSRES
 MDISK 190 3330 126 072 VMSRES MW ALL     WMAINT  MMAINT
 MDISK 191 3330 283 022 VMSRES MW RMAINT  WMAINT  MMAINT
 MDISK 193 3330 081 045 VMSRES MW ALL     WMAINT  MMAINT
 MDISK 194 3330 021 045 VMSRES MW RMAINT  WMAINT  MMAINT
 MDISK 196 3330 090 030 VMPK01 MW RMAINT  WMAINT  MMAINT
 MDISK 19D 3330 331 060 VMSRES MW ALL     WMAINT  MMAINT
*
```

```
 MDISK 201 3330 207 045 VMSEXT MW RMAINT  WMAINT  MMAINT
 MDISK 293 3330 335 024 VMPK01 MW RCMSAUX WCMSAUX MCMSAUX
 MDISK 294 3330 359 023 VMPK01 MW RCPAUX  WCPAUX  MCPAUX
 MDISK 295 3330 286 011 VMSEXT MW RUSRMOD WUSRMOD MUSRMOD
 MDISK 296 3330 051 035 VMPK01 MW RCPAUX  WCPAUX  MCPAUX
*
 MDISK 319 3330 195 012 VMSEXT MW ALL     WMAINT  MMAINT
 MDISK 393 3330 001 200 VMSTG2 WR RMAINT  WMAINT
 MDISK 394 3330 001 260 VMSTGE WR RMAINT  WMAINT
 MDISK 396 3330 261 070 VMSTGE WR RMAINT  WMAINT
*
 MDISK 496 3330 111 002 VMSEXT MW RIPCSX  WIPCSX  MIPCSX
 MDISK 497 3330 201 020 VMSTG2 WR RMAINT  WMAINT
*
 MDISK 595 3330 021 065 VMGCS1 MW RMAINT  WMAINT  MMAINT
 MDISK 596 3330 086 024 VMGCS1 MW RGCSAUX WGCSAUX MGCSAUX
 MDISK 59E 3330 122 020 VMGCS1 MW ALL     WMAINT  MMAINT
```

A summary of MAINT's minidisk use follows:

Disk	Contents
190	Current CMS system disk
191	Work area. This is MAINT's A-disk.
193	CMS base disk
194	CP and RSCS text retention
196	HPO text
19D	HELP files
201	EREP txtlib
293	CMS PTFs, updates, and updated text decks (object modules). The updated text decks may not be present in some installations.
294	CP and RSCS PTFs, updates, and updated text decks (object modules)
295	Local updates (modules)
296	HPO PTFs and updates
393	CMS source and macros
394	CP and RSCS source, macros, and copy files
396	HPO source, macros, and copy files
496	IPCS service
497	ICPS source
595	GCS object code
596	GCS service
597	GCS source
ccuu	CP system residence device, or a replica of it, for test purposes

A Hardware Service Virtual Machine (EREP)

The following directory entry defines the virtual machine (EREP) that can be used by the hardware service representative. This virtual machine usually has class F command privileges. For more information on the hardware service virtual machine, see the *VM/SP HPO OLTSEP and Error Recording Guide.*

```
USER EREP IBMCE 768K 2M FG
     ACCOUNT EREP IBMCE
          IPL CMS
          CONSOLE 01F 3215
          SPOOL 00C 2540 READER A
          SPOOL 00D 2540 PUNCH  B
          SPOOL 00E 1403
          LINK MAINT 190 190 RR
          LINK MAINT 19D 19D RR
          MDISK 191 3330 306 002 VMSRES WR READ WRITE
```

Note: EREP support, beginning with VM/SP Release 3.0, is separately orderable and is no longer included in the VM/SP distribution tapes.

The Directory Supplied with the Starter System

A control statement file for the directory program is supplied with the:

- 3330 Starter System
- 3350 Starter System
- 3375 Starter System
- 3380 Starter System

If the supplied control statement file meets your needs, you can execute the directory program using the supplied control statements. Otherwise, you can code your own control statements or edit the supplied control statements to produce the file you need for your installation. See the *VM/SP HPO System Definition Files* for supplied control statement files.

Warning: The passwords in the directory supplied with the starter system are restricted. You must substitute your own passwords. Any restricted password you do not replace will be changed to NOLOG when the directory program is run. That virtual machine will not be able to log on.

Allocating DASD Space for the Directory

Before you create a directory using the directory program, be sure you have enough DASD space allocated as directory space (DRCT). Use the CP Format/Allocate service program to format and allocate space to be used for the directory. The space must be allocated DRCT. To calculate the total space required, first calculate the total number of records used.

If your directory has any ACIGROUP control statements, use this formula:

$$NR = \frac{NU}{169} + \frac{(NU \times 4) + (NM \times 3) + (NI \times 2) + NOS}{170}$$

Otherwise, use this formula:

$$NR = \frac{NU}{169} + \frac{((NU + NM) \times 2) + NOS}{170}$$

where:

NR = Total number of records used
NU = Number of USER control statements
NM = Number of MDISK control statements (except for temporary disks)
NI = Number of IPL control statements with the PARM option
NOS = Number of other control statements

Then, calculate the number of cylinders (NC) required:

- For 2314, 2319: NC = NR/32
- For 2305, 3340: NC = NR/24
- For 3330: NC = NR/57
- For 3350: NC = NR/120
- For 3375: NC = NR/96
- For 3380: NC = NR/150
- For FB-512: the space required is equal to NR

Notes:

1. *You can reduce the amount of space needed for the directory by using the PROFILE and INCLUDE control statements.*

2. *You should initially format and allocate enough space for two directories. You can then build a new directory whenever needed, without overlapping the current one, and without formatting and allocating space each time a new directory is created. If you wish to reallocate the area in which the directory resides, you must reallocate the DASD space and then rerun the directory program. When a directory is written to a count-key-data DASD, space is allocated from the available cylinders on a cylinder-by-cylinder basis, and a minimum of two cylinders is used as DRCT space.*

When a directory is written to an FB-512 DASD, space allocation proceeds as follows:

- If there are already two DRCT extents (one in use and the other available for use) the new directory is written to the available extent. The available extent is flagged as in use, and the previously in use extent is flagged as available (the directories are swapped).

- If there is only one DRCT extent (it must be available), an attempt is made to divide it into two DRCT extents, allowing subsequent directories to be swapped dynamically. This is done as follows:

  1. If insufficient space is specified for the current directory, it is not written to the DASD volume.

  2. If sufficient space exists, the directory program attempts to divide it into two equally sized DRCT extents (one to hold the current directory and one to be available for future swapping of directories).

  3. If there is not enough space to create two equally sized DRCT extents, the current directory is built and the remaining space is reserved as available DRCT space.

If space for two directories is not initially allocated, each time you want to create a new directory, you must allocate space for the directory before you create it.

The Directory Program

You can run the directory program under CMS (using the DIRECT command) or stand-alone. The stand-alone version is provided in object deck form (a three card loader, followed by the DMKDIR text deck), and may be loaded directly from either a real or virtual card reader.

If you run the directory program under CMS, input records must be in a CMS file with a default fileid of USER DIRECT. The DIRECT command loads the directory creation module. If no filename is specified, the program looks for a file named USER DIRECT. Otherwise, it looks for a file with a filetype DIRECT.

If the file is not found, or if an error occurs during processing, the directory is not created and the old directory remains unaltered.

Normal completion writes the DASD address of the new directory in the VOL1 label. If it is updating the active system directory, it places the new directory in use by VM/SP HPO. You can print the new directory by issuing the CMS command PRINT USER DIRECT (or PRINT filename DIRECT).

The virtual machine executing the directory program must have write access to the volume to contain the new directory. If you create a directory that is to be written on the active system residence volume, your virtual machine's current directory entry must have write access to the volume containing the current directory.

Example: Assume that you have the following virtual machine for online directory modification.

```
USER DIRMAINT DIRM 1M 2M BG
 ACCOUNT 4 SYSADMIN
 OPTION REALTIMER
 IPL CMS
 CONSOLE 009 3215
 SPOOL 00C 2540 READER *
 SPOOL 00D 2540 PUNCH  A
 SPOOL 00E 1403 A
 SPECIAL OFF TIMER
 LINK MAINT 190 190 RR
 LINK MAINT 19E 19E RR
 MDISK 191 3330 399 005 VMIPFE MR RDIRM WDIRM MDIRM
 MDISK 193 3330 118 017 VMIPFE MR RDIRM WDIRM MDIRM
 MDISK 195 3330 282 017 VMSRES MR RDIRM WDIRM MDIRM
 MDISK 123 3330 000 404 VMSRES MW
*
```

Using the System Product Editor, you can create or modify a card-image file of the directory input. When you are ready to write a new directory, issue the command

```
DIRECT  filename
```

where filename is a CMS file (normally named USER) with filetype DIRECT containing the necessary directory program control statements. The DIRECT command puts this file into the form of a directory, and replaces the old directory with this new one.

Loading the DMKDIR object deck via the card reader is the same as issuing the DIRECT command in CMS, except that after IPL, the program asks you for the address of a card reader containing the Directory program control statements.

Once the directory is updated, not all directory changes for a user currently logged on to the system take effect until the user logs off the system and then logs back on.

When a new directory is written for a new system residence volume, the new directory does not take effect until the new system residence volume is loaded (via IPL). In order for all changes to become effective, the user has to log off and log on again.

Invoking the Directory Program (DMKDIR)

The directory program records the configuration of each user's virtual machine in the directory. Each virtual machine configuration includes counterparts of the components found in a real System/370: a virtual operator's console, virtual storage, and virtual I/O devices and control units.

The same version of the directory service program deck can be placed in the card reader and loaded directly, or run in a virtual machine under CMS.

The CMS file named DIRECT can be updated with the editor to include additional directory entries.

The CMS DIRECT Command

Use the CMS DIRECT command to process a file to see if it follows the required directory format. To actually change or swap the currently active directory, you must have both:

1. User class A, B, or C

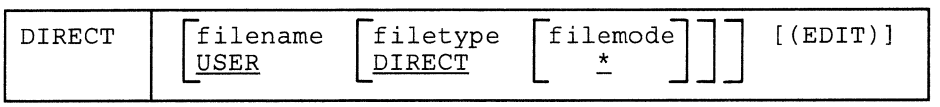
2. Write access to the system-owned (system residence or IPL device) volume that contains the current directory up to and including the directory cylinders, or to the volume that is to contain the new directory.

If you have the above qualifications and wish to verify that a CMS file can be used as a directory file, you must use the EDIT option. Otherwise, if there are no control statement errors, the file is put into active use.

To build a directory on a CP-owned volume using preallocated cylinders, a new directory should be built so as not to overlay an existing directory. You must, therefore, allow space for two directories, or allocate a new area for the directory each time it is created.

If you execute the directory program under VM/SP HPO, the newly created directory is dynamically swapped, and placed in use by VM/SP HPO (provided that you have class A, B, or C privilege and the directory you update is currently in use by the system, or the target directory pack is present in the system owned list). If you do not have the proper privilege class, the directory is updated on the directory volume but not dynamically swapped. In that case, the change will not go into effect until the next time the system is loaded (via IPL).

The format of the DIRECT command is:



where:

filename [filetype [filemode]]
is the identification of the file containing control statements for the directory program. For convenience, the filetype should be DIRECT, although other filetypes are permitted.

- If only filename is given, filetype defaults to DIRECT.
- If no filename and filetype are given, the program defaults to a file named USER DIRECT.
- If filemode is not specified, it defaults to *.

(EDIT)
specifies that the directory is to be examined, but not changed.

Under CMS, the DIRECT command loads the directory creation module. The first statement encountered must be a DIRECTORY statement. If the DIRECTORY statement is not found, the program stops. A syntax error in any statement generates an error message, and the directory is not updated. If no critical errors are encountered, the remaining statements are checked for syntax.

If the directory program ends abnormally, the old directory is not altered. Normal completion makes the directory available for use by VM/SP HPO. After the new directory is created, it can be printed by issuing the CMS command PRINT USER DIRECT or PRINT filename filetype.

The DIRECT command filename and filetype operands default to a CMS file identification of USER DIRECT. The filemode defaults to * if not specified. Defaults can be overridden on the command line. The EDIT option allows you to run the program without updating the directory on disk. This enables you to check the syntax of the directory statements without accessing the directory disk.

Invoking Directory as a Stand-alone Program

Stand-alone directory program operation in a virtual machine is the same as CMS operation, with this exception: after IPL, the program asks you for the virtual card reader address. If you enter a null line, the IPL device address is the default of 00C. The directory control statements must be in the same virtual reader as the directory program itself.

When running as a stand-alone program, DMKDIR searches for a console at address 009 or 01F. If there is no operational console at one of these addresses, the program enters a wait state until an interrupt occurs to identify the address of the console. If any non-console device is physically connected to address 009 or 01F, it must be detached or results are unpredictable.

Note: The stand-alone directory program does not check for restricted passwords.

Directory Control Statements

The following directory control statements are described in this section:

DIRECTORY
defines the device on which the directory is allocated.

PROFILE
defines a directory profile (a set of statements included in user directory entries).

USER
defines a virtual machine, creates and delimits a directory entry for one user.

INCLUDE
invokes a directory profile as part of a user entry.

CLASS
defines the user privilege classes assigned to one user.

ACCOUNT
defines an account number and distribution identification.

ACIGROUP
assigns a user to an access control group.

OPTION
selects specific user options.

IUCV
defines an authorization for establishment of a communication path with another virtual machine or a CP system service.

IPL
contains the name of the system to be loaded for the user when he logs on.

SCREEN
defines color and extended highlight options for the user terminal.

CONSOLE
specifies the virtual console.

MDISK
describes the cylinder extent on a direct access device to be owned by the user.

SPOOL
specifies the unit record device that is to be spooled.

DEDICATE
specifies that a real device is to be dedicated to this user.

LINK
makes a device that belongs to another user available to this virtual machine at logon time.

SPECIAL
specifies the I/O units available to the user that need not have a real I/O unit available.

Control statements should be in the formats illustrated in the following pages, with one or more blanks as operand delimiters. All operands are positional from left to right. If any operand is omitted, remaining operands in that statement must be omitted, with the exception of the OPTION statement. Its entries are self-defining and not positional.

Only columns 1 through 71 are inspected by the program. All data after the last possible operand on any card is ignored. Also, blank cards and cards having an asterisk (*) as the first operand are ignored.

If any input card is found to be in error, the program continues to validate all control statements before terminating. If the directory runs out of space, the program terminates immediately. After an abnormal termination, the old directory is not altered, and the new directory is not saved.

DIRECTORY Control Statement

The DIRECTORY control statement defines the device on which the directory is allocated. It must be the first statement.

The format of the DIRECTORY control statement is:

```
DIRectory  ccuu  devtype  volser  [alt-ccuu]
```

where:

ccuu
is the address of the device to contain the directory and is specified in three or four hexadecimal digits from 000 to 1FFF.

Note: If the program is being run in a virtual machine, the address is a virtual address. If the program is being run in a real machine (stand-alone), the address is a real address.

devtype
represents a supported device type suitable for the directory (2314, 2319, 2305, 3330, 3340, 3350, 3375, 3380, or FB-512). For a 3350 device in native mode, specify 3350 as the device type. For a 3350 used in 3330 compatibility mode, specify 3330. Specify a 3344 disk as a 3340, and a 3333 as a 3330.

Note: 3330V (virtual 3330) volumes associated with 3850 Mass Storage System cannot be specified as the residence device for the directory.

volser
is the volume serial number of the directory volume (one to six alphameric characters).

alt-ccuu
identifies an alternate device address for writing the directory if the primary ccuu is unavailable. It is specified in three or four hexadecimal digits from 000 to 1FFF. For multiprocessing systems, an alt-ccuu specification facilitates native execution of the directory program on either processor.

PROFILE Control Statement

The PROFILE control statement defines a directory profile and creates a profile entry in the directory. (For a description of directory profiles, see "Using Directory Profiles" on page 174.) You must prepare a separate PROFILE statement for each profile entry required.

Profile entries are optional. If you code them, they must appear immediately after the DIRECTORY control statement and before the first USER control statement. There is no limit to the number of PROFILE entries you may code.

The format of the PROFILE control statement is:

```
Profile profilename
```

where:

profilename
is the name of the directory profile. A profilename may consist of one to eight alphanumeric characters. Only one profilename may be specified on a PROFILE control statement.

USER Control Statement

The USER control statement defines a virtual machine and creates a user entry in the directory. It delimits the directory entry for one user. You must prepare a separate USER statement for each user entry required.

The format of the USER control statement is:

```
User userid pass [stor [mstor [cl [pri [le  [ld  [cd  [es  ]]]]]]]]
                                       ON    ON    ON    ON
                                       OFF   OFF   OFF   OFF
```

where:

userid
is a one- to eight-character user identification. Any alphameric characters may be used except SYSTEM. SYSTEM is the userid of the system VMBLOK, and should never be used for a virtual machine. There must be at least one CONSOLE or SPOOL directory entry for each user. These options are discussed in the directory program control statement descriptions.

If you plan to use any of the following CMS commands, the userid must conform to those characteristics valid for CMS filenames.

FILELIST	NAMEFIND	NAMES	NOTE
RDRLIST	RECEIVE	SENDFILE	TELL

These commands create and reference files that have the userid as the filename. Valid characters for CMS filenames are A-Z, 0-9, $, #, @, +, - (hyphen), : (colon), and _ (underscore).

Notes:

1. *The userid should not contain the characters "LOGOxxxx", where xxxx is a terminal address of the installation. During logon this character string is assigned to the terminal at address xxxx from the time the initial interrupt is received until the user is identified.*

2. *The userid should not contain the same characters as the "LUNAME " defined in VTAM. The "LUNAME", although unique to each VTAM service machine, may not be unique to VM/SP if more than one VTAM service machine is running under VM/SP HPO.*

3. *Do not specify SYSTEM or ALL as a userid. VM/SP HPO reserves SYSTEM and ALL as an identifier for its own use.*

4. *If the userid of AUTOLOG1 (a reserved system user identification) is used, during the VM/SP HPO IPL operation, the AUTOLOG1 virtual machine is automatically logged onto the system.*

In application, the AUTOLOG1 virtual machine could be the CMS batch virtual machine, or a virtual machine that, through the use of the directory's IPL statements, loads a CMS named system. Then the CMS system, using a PROFILE EXEC with AUTOLOG command statements within the EXEC file, initiates the logon of other virtual machines to the system.

5. *If the userid is 1-4 all numeric characters, unpredictable responses may occur for commands using both the userid and a 1-4 digit numeric spoolid as parameters (such as the class D query command).*

pass
is a one- to eight-character user-security password that must be entered by the user to gain access to the VM/SP HPO system and the virtual machine you are defining in these control statements.

Notes:

1. *Use the reserved password NOLOG for users who do not have a virtual machine configuration in the directory. Users whose password is NOLOG can use the real card reader for CMS batch processing, but they cannot log on.*

2. *Be sure that the password is not restricted. If you define one of the restricted passwords in the RPWLIST DATA file for a user, it will be changed to NOLOG when the directory program is run. That user will not be able to log on. (See "Passwords That Don't Work" on page 176.)*

stor
is one to eight decimal digits that define the virtual machine's storage size. It must be a multiple of 4K. All entries not on a 4K boundary are rounded up to the next 4K boundary. The last character must be K or M. The minimum size is 8K. The maximum size is 16M. The default is 128K.

mstor
is one to eight decimal digits that define the maximum virtual machine storage size that this user can define as his storage after logging on the system. It must be coded in multiples of 4K. All entries not on a 4K boundary are rounded to the next 4K boundary. The last digit must be K or M. The maximum size that can be specified is 16M. The minimum size is 8K. The default size is 1M.

cl
is a one- to eight-character field defining up to eight user classes assigned to the user. Each character represents a single privilege class. Valid characters are the letters A to Z and the digits 1 to 6. Characters may appear in any order, without duplication, and must not be separated by blanks.

An asterisk (*) placed in this field indicates that a CLASS control statement follows. Use the CLASS control statement to define more than eight privilege classes or as another way to define eight or fewer. If the field is not specified, or if it contains an asterisk and there is no CLASS control statement, the default class or classes assigned by the SYSFCN macro will be used.

Note: If privilege class F is assigned to a virtual machine, I/O error recording is not automatically done. This allows the class F user to set the kind of error recording he wants to perform.

pri

is a number from 1 to 99 used by the CP priority dispatcher. One is the highest priority; 64 is the default.

Note: The same priority value can be used for several users. Also, if the specification for this statement is not entered, line end (le), line delete (ld), character delete (cd), and escape (es) characters default to system-defined values.

The special VM/SP HPO logical editing symbols below may be set ON, OFF, or substituted with two hexadecimal characters or one graphic character of your choice.

Note: In addition to directory specification, you can change these logical editing symbols using the TERMINAL command. The default value for all symbols is ON. The exception to this rule is a virtual machine initiated by the CP AUTOLOG command. In this case all logical line editing is OFF.

le

is a one-character "line end" symbol or a two-character hexadecimal representation of the symbol. ON sets the system default value (#). OFF disallows "line end" symbol usage. For example:

"le" can be coded as + or 4D or ON or OFF.

ld

is a one-character "line delete" symbol or a two-character hexadecimal representation of the symbol. ON sets the system default value (<-). OFF disallows "line delete" usage.

cd

is a one-character "character delete" symbol or a two-character hexadecimal representation of the symbol. ON sets the system default value (@). OFF disallows "character delete" usage.

es

is a one-character "escape-character" symbol or a two-character hexadecimal representation of the symbol. ON sets the system default value (″). OFF disallows "escape character" symbol usage.

INCLUDE Control Statement

The INCLUDE control statement invokes a directory profile as part of the user entry. (For a description of directory profiles, see "Using Directory Profiles" on page 174.)

The INCLUDE control statement is optional. If you use it, you must place it immediately after the USER control statement. Only one INCLUDE control statement may be specified in a user directory entry.

The format of the INCLUDE control statement is:

```
INclude profilename
```

where:

profilename
is the name of the directory profile. A profilename may consist of one to eight alphanumeric characters. Only one profilename may be specified on an INCLUDE control statement.

CLASS Control Statement

The optional CLASS control statement defines up to 32 user classes assigned to a user. It must follow the USER control statement and precede the first device control statement (CONSOLE, MDISK, DEDICATE, LINK, or SPOOL). If the CLASS control statement is used, the USER control statement must have an asterisk (*) in its class field. The format of the CLASS control statement is:

```
Class classes
```

where:

classes
is a 1- to 32-character field defining the classes assigned to a user, one character per class. Valid characters are the letters A to Z and the digits 1 to 6. As long as no characters are duplicated or separated by blanks, they may appear in any order.

ACCOUNT Control Statement

The ACCOUNT control statement defines an account number and a distribution identification. The distribution identification has no internal system use. It is provided for customer use (for example, a code for distribution of printed output). The ACCOUNT statement is optional. However, if this statement is omitted, both the account number and the distribution code default to the userid. This statement (if coded) must precede the first device statement (CONSOLE, MDISK, DEDICATE, LINK, or SPOOL).

The format of the ACCOUNT control statement is:

```
Account  number  [distribution]
```

where:

number
is a one- to eight-character account number punched in the accounting data for this virtual machine. The USERID from the USER statement is also punched in the accounting data.

distribution
is a one- to eight-character distribution identification word that is printed or punched with the userid in the separator for spooled output for this user. This value is optional and defaults to the userid from the USER statement if omitted.

ACIGROUP Control Statement

The ACIGROUP statement is used to define a user as a member of an access control group. This statement is optional. If you use it, it must follow the USER control statement. It must precede the first device statement (CONSOLE, MDISK, DEDICATE, LINK, or SPOOL). Only one ACIGROUP statement may appear in a single user's directory entry.

The format of the ACIGROUP control statement is:

```
ACIgroup  groupname
```

where:

groupname
is the one- to eight-character name of an access control group. It must not be the same as any userid. Only one groupname may be specified.

OPTION Control Statement

The OPTION control statement selects specific options. This statement is optional and, if used, must precede the first device statement (CONSOLE, MDISK, DEDICATE, LINK, or SPOOL). Multiple OPTION statements can be inserted if the options selected exceed one statement record length.

The format of the OPTION control statement is:

```
Option  [Realtimer] [Ecmode] [Isam] [Virt=real] [PMA] [Acct]
        [Svcoff] [BMX] [CPUID bbbbbb] [AFFinity nn]
        [VMsave] [STFirst] [370E] [XMEM] [Maxconn nnnnn]
        [MIH] [MAXDev xxxx] [DIAG98] [SVCAccl]
```

where:

REALTIMER
provides a timer for the virtual machine that is updated during virtual processor run time and during virtual wait time. If the virtual machine does not have the REALTIMER option, its timer reflects only the virtual processor run time used. This option is required for virtual machines running systems or programs that go into a wait state expecting a timer interruption. This timing ability can also be obtained by issuing the CP command SET TIMER REAL.

ECMODE
allows the virtual machine to run in extended control mode. The ECMODE option must be specified for virtual machines using operating systems that:

- Operate in System/370 extended control mode (such as VM/SP HPO itself)
- Use the dynamic address translation facility (such as OS/VS1, OS/VS2, DOS/VS, DOS/VSE, VSE/AF, VM/370, VM/SP, and VM/SP HPO)
- Use control registers other than zero (such as OS GTF (General Trace Facility), which uses Monitor Call and requires control register eight)
- Depend on the System/370 extended channel masking feature
- Use preferred machine assist or preferred machine assist with control switch assist

The ECMODE option must also be specified for the virtual machine that is to perform system support or updating, and for an RSCS virtual machine. ECMODE is also required when using the clock comparator.

Note: A virtual machine defined without the ECMODE option in the directory is limited to six I/O channels, while a virtual machine with the ECMODE option may address up to 16 I/O channels. If a virtual machine with ECMODE executes in basic control mode,

the I/O masking for channels 6 and higher is simulated by the extended channel feature. If a virtual machine with the ECMODE option executes in extended control mode, the I/O masking for all 16 channels is handled via extended control register 2. This facility can also be obtained by issuing the CP command SET ECMODE ON.

ISAM
provides special channel command word translation routines that permit OS/PCP, MFT, and MVT ISAM programs (which dynamically modify their CCWs) to operate properly in a virtual machine. This is required only for virtual machines that use OS/PCP, MFT, or MVT ISAM access methods or OS/VS ISAM when executing either in a V=R partition or in a non-paging mode under OS/VS. This option is not needed for DOS, DOS/VS, DOS/VSE, VSE/AF, or OS/VS ISAM when run only in a V=V partition of OS/VS. This facility can also be obtained by issuing the CP command SET ISAM ON.

VIRT=REAL
is a performance option that allows you to place your virtual machine in lower storage, such that its virtual storage addresses correspond to real storage addresses (except page zero, which is relocated). The real page zero is controlled by the CP nucleus. No CCW translation is required. This option is required for a virtual machine to successfully execute self-modifying channel programs other than those generated by OS/VS TCAM (Level 5, generated or invoked with the VM/370 option) or OS ISAM. VIRT=REAL can be specified for any number of virtual machines, but only one virtual machine can use this facility at any given time. A named or shared system cannot be loaded (via IPL) in a virtual=real area. The device address must be specified in the IPL command. To generate a VM/SP HPO system with a virtual=real machine, see "Specifying a Virtual=Real Machine" in Chapter 10.

PMA
is an option that allows an MVS/SP virtual=real virtual machine to use preferred machine assist or preferred machine assist with control switch assist. This option, if specified, must appear *on the same card image* and *immediately* after the VIRT=REAL option.

By issuing the IPL command with the PMA option, the virtual=real virtual machine operator sets on preferred machine assist. The IPL command with the PMAV option sets on preferred machine assist with control switch assist. The following restrictions apply when either of these options is specified:

- You *must* use the appropriate release of MVS/SP for your processor: MVS/SP Release 1 Enhancement or a later release for the 3033, 308x, and 4381; MVS/SP Version 1 Release 5 Modification 3 for the 3090. Results are unpredictable if the operator loads an improper operating system.

- The virtual machine configuration *cannot* have:
  - A virtual disk that is not a full-extent virtual disk (for this reason, a temporary disk (TDISK) cannot be specified)
  - A dedicated device *or* full-extent virtual disk for which:
    - The virtual and real addresses are not the same *and* there is a real device control block (RDEVBLOK) for the said virtual address.
    - The virtual channel address does not match a channel defined in the DMKRIO file.
  - A nondedicated device for which:
    - There is a real device control block (RDEVBLOK) for the specified virtual address.
    - The virtual channel address does not match a channel defined in the DMKRIO file.
- To ready the virtual machine for preferred machine assist or preferred machine assist with control switch assist:
  - If the user logs onto the system and VM/SP HPO is in attached processor or multiprocessor mode, the user must set affinity to one processor. The AFFINITY option must be in the OPTION statement of the user's directory entry or the primary system operator must set affinity for the user's virtual machine (class A command SET AFFINITY userid nn).

    If VM/SP HPO is in uniprocessor mode when the user logs on, subsequent affinity settings are automatic. If, for instance, the system operator issues the CP class B command VARY ONLINE PROCESSOR nn, the preferred virtual machine's affinity is automatically set to the processor on which the machine was initialized.
  - If the option ECMODE does not appear in the OPTION statement in the user's directory entry, the user must issue the CP class G command SET ECMODE ON.
- If the preferred virtual machine exclusively owns the device used for IPL, (the device is undefined in the DMKRIO file), then the device must be a count-key-data DASD, else results are unpredictable.

  If the device used for IPL is a virtual device, it cannot be a device that violates certain virtual machine configuration restrictions (see the bullet above that describes these restrictions).

For more information on preferred machine assist and preferred machine assist with control switch assist, see "Preferred Machine Assist and Preferred Machine Assist with Control Switch Assist" on page 133.

ACCT
a user with the ACCT option in his directory can track another's use of virtual machine resources. For example, a user who sends a job to the CMS batch virtual machine can be charged for time used in the batch machine. Note that the ACCT option should be specified in the directory of the CMSBATCH virtual machine so user/job identifying information is printed on the forms separators of spooled output files.

SVCOFF
specifies that CP, instead of the virtual machine assist feature or the VM/370 Extended Control - Program Support handles all SVC interrupts for this virtual machine. A user whose directory entry contains this option can override it by issuing SET ASSIST SVC.

Note: All SVC 76 interrupts are handled by CP whether or not the SVCOFF option is specified.

BMX
specifies that all virtual machine I/O operations are to occur as block multiplexer channel operations rather than selector channel (the default) operations. In block multiplexer mode, the virtual channel is not busy until the initial SIO is complete (selector mode operates similarly). Block multiplexer allows the successful start of multiple SIOs to different devices on the same channel. However, virtual I/O operations on channel 0 are processed as byte multiplexer channel operations. Channels that have a channel-to-channel adapter are restricted to selector channel operation.

The channel mode setting for all channels except virtual channel zero can be changed by the CP DEFINE CHANNEL command.

CPUID bbbbbb
provides a unique processor identification (CPUID) to be stored in response to the STIDP instruction. It is necessary to associate a unique CPUID with each virtual machine attached to an MSC port. This is because solicited/unsolicited messages are directed to the host system in the virtual environment using the CPUID. CP does not check to ensure that all virtual machines using the SET CPUID command have specified unique processor serials. The hexadecimal field 'bbbbbb' is the processor identification number. The processor identification number (serial) is only a portion of the complete CPUID. The CPUID identification stored in response to a STIDP instruction is a string of 16 hexadecimal digits shown as follows:

```
aabbbbbbccccdddd
```

where:

aa
is the version code. These two digits are forced to X'FF' to identify that the virtual machine is running under VM/SP HPO.

bbbbbb
is up to six hexadecimal digits that indicate the processor identification number. This field is set by the directory OPTION statement values or modified by the SET CPUID command.

cccc
is the model number. This field contains a high order 0 digit, followed by the three digits of the model number (0-9). This field defaults to the model number of the real machine.

dddd
is the machine check extended logout. This field is forced to X'0000' since CP does not reflect machine checks to the virtual machine.

If the CPUID is not specified by the SET CPUID command or the OPTION control statement, the CPUID stored as a result of the STIDP instruction is the real CPUID. The first two digits are set to X'FF' and the last four digits are set to X'0000' (present CPUID logic). A processor serial of more than six digits on the SET CPUID command results in an error message.

A processor identification number (serial) of fewer than six hexadecimal digits results in zeros to the left of the number. A three-byte field in the VMBLOK (VMCPUID) contains the value set as a result of invoking this DIRECTORY option.

AFFINITY nn
is two decimal digits between 00 and 63, specifying that virtual machine execution is to be performed on a designated processor (nn). This attribute is applicable only in attached processor and multiprocessor environments. Any hexadecimal value from 00 to 3F is a valid main (IPL) or attached (non-IPL) processor address. However, the value selected must match the preset values established for your installation's IPL and non-IPL processors when the system was installed. If the AFFINITY option is not selected, the virtual machine is serviced by the first available processor from the dispatch queue. An affinity setting in the directory can be overridden by the class A CP command SET AFFINITY. If the system is running in attached processor or multiprocessor mode and an error forces recovery to uniprocessor mode, the affinity setting of virtual machines assigned to to the non-IPL processor is nullified. Virtual machine processing may continue on the IPL processor.

VMSAVE
specifies that the virtual machine contents are to be saved if VM/SP HPO is terminated or if VM/SP HPO terminates the indicated virtual machine. The contents of the registers and real storage of the virtual machine are saved on DASD space and made available to userid(s) specified by the NAMESYS macro instruction.

Notes:

1. *This option is effective only if the user has exactly one VMSAVE DASD area defined. The option is enabled only if that area does not contain a VMSAVE system. If more than one area is defined, or if a valid system is already stored in that area, the SET VMSAVE command must be used.*

2. *The VMSAVE option does not save vector status or vector registers.*

3. *To cancel the VMSAVE specification use the SET VMSAVE OFF command.*

STFIRST
specifies that the indicated virtual machine is authorized to use the SET STBYPASS command when virtual machine assist is active on the system for a virtual=virtual user.

Note: This is a restricted performance option that should be reserved only for virtual machine userids used to run production MVS, SVS, OS/VS1, or DOS/VS operating systems.

370E
specifies that the MVS/System Extensions or MVS/System Product support be enabled for the indicated virtual machine. See the table under "Using Performance Options" in Chapter 10 for a list of processors that support this option.

XMEM
enables MVS cross memory services assist for the MVS/System Product virtual machine. The virtual machine is able to use the System/370 Extended Facility enhancements and cross memory services implemented in Release 3 and subsequent releases of the MVS/System Product. Cross memory is initiated if it is present on one or both processors of an attached processor or multiprocessor system.

Note: If you enter the XMEM option in the directory for a virtual machine, you must also include the 370E option.

MAXCONN nnnnn
is the maximum number of Inter-User Communications Vehicle (IUCV) connections allowed for this virtual machine. If the MAXCONN option is omitted, the default is 4. The maximum is 65,535.

Note: The MAXCONN option is applicable only to virtual machines. For CP system code, a limit of 4,096 paths is established when the CP system is initiated.

MIH

When MIH is included as an option in the directory, the missing interrupt handler is turned on and CP simulates detected missing interrupts.

MAXDEV xxxx

is the maximum number of virtual devices a user may define. This may be any decimal value between 1 and the result of the calculation

```
(32767/VDEVBLOK size in doublewords) + 1
```

with the result of the division rounded down to the preceding whole number.

Note: 32767 is equal to X'7FFF'.

If MAXDEV is not coded, the default maximum number of devices that a user may attach is 410.

The maximum value for MAXDEV varies according to the size of the virtual device block. Since the current size of the virtual device block is 10 doublewords, the maximum value for MAXDEV is 3277. For changes in the VDEVBLOK size and resulting values for MAXDEV, see *VM/SP HPO Data Areas and Control Block Logic—CP.*

Note: For a virtual machine running in BC mode, the maximum value of MAXDEV is 1536.

DIAG98

authorizes the indicated virtual machine to use DIAGNOSE code X'98'. See the *VM/SP HPO System Programmer's Guide* for information about DIAGNOSE X'98'.

Note: A virtual machine should use only the real addresses returned by DIAGNOSE X'98' in its real channel programs. The virtual machine is responsible for any security violations it may cause by using any other real addresses.

SVCACCL

authorizes the use of the Supervisor Call Instruction Acceleration (SVCACCL) by the indicated virtual machine when IX/370 is running under CP. In addition to specifying the SVCAccl parameter of the OPTION control statement, you must enable the code by issuing the CP SET SVCAccl ON command. For more information on the CP SET command, see the *CP Command Reference for General Users.*

IUCV Control Statement

The IUCV control statement defines an authorization for establishment of a communication path with another virtual machine or a CP system service. A virtual machine can initiate communications with itself without directory authorization. The ability to initiate a CONNECT to a CP system service always requires specific authorization. CP system service communications must be restricted to service type virtual machines.

The format of the IUCV control statement is:

```
IUCV  [userid  ]  [PRIORITY] [MSGLIMIT limit]
      [*CCS    ]
      [*SIGNAL ]
      [ALLOW   ]
      [ANY     ]
```

where:

userid
is the one- to eight-character user identification of the virtual machine or CP system service that authorizes this virtual machine a communication path. If you are going to run VCNA, use *CCS as a userid.

***CCS**
is the CP system service name required for VCNA.

***SIGNAL**
is the CP system service name required to become a member of a Virtual Machine Group.

ALLOW
is a general authorization indicating that any other virtual machine may establish a communications path with this virtual machine. No further authorization is required in the virtual machine initiating the communication.

ANY
is a general authorization indicating that a communications path can be established between this virtual machine and any other virtual machine. ANY does not apply to CP system service names.

PRIORITY
indicates that a communications path with the specified virtual machine or CP system service is capable of handling priority as well as nonpriority communications if requested via the CONNECT function. If the PRIORITY option is omitted, no path authorized by this entry is able to handle priority messages.

MSGLIMIT limit
defines the maximum number of outstanding messages allowed on any path authorized by this entry. A lower limit can be specified via the CONNECT and ACCEPT functions. If omitted, the message limit is taken from the CONNECT or ACCEPT parameter list. Limits specified by CONNECT and ACCEPT can be different. If omitted from the parameter list, a default of 10 is used. The maximum allowed value for message limit is 255.

Notes:

1. *PRIORITY and MSGLIMIT are keywords and can be specified in any order.*

2. *When the CONNECT function is invoked, the directory entries are searched in a definite order:*

   - *The invoker's IUCV control statements are searched for an entry for the target's userid, and then*
   - *If the target is not a CP system service, the invoker's IUCV control statements are searched for an ANY entry.*
   - *The target's IUCV control statements are searched for the ALLOW entry.*

   The first entry found that is applicable is used to establish the priority status and message limit for the path. Connections to CP must be authorized by a directory entry.

3. *Connections invoked from CP system code do not need directory authorization. Priority status and message limit are taken from the CONNECT parameter list.*

4. *The IUCV control statement can be used as many times as needed for any user to define authorizations needed for that virtual machine.*

5. *Communications with Console Communication Services (CCS) for VCNA and VSCS require an IUCV control statement with *CCS as the userid.*

6. *Communication with the CP Message System Service does not require authorization by the virtual machine. If an IUCV control statement is specified, a userid of *MSG should be used. Although the PRIORITY and MSGLIMIT options may be specified on the control statement, the specifications will never be used since the virtual machine will only be receiving messages.*

7. *Communication with the CP Signal system service does not require authorization by the virtual machine. If an IUCV control statement is specified for the Signal system service, a userid of *SIGNAL should be used.*

IPL Control Statement

The IPL control statement contains the name of the system to be loaded for the user when he logs on. This statement is optional. If specified, it must precede the first device statement (CONSOLE, MDISK, DEDICATE, LINK, or SPOOL). This control statement may contain data to be passed to the system being loaded. The IPL statement can be overridden by the user at logon time by specifying "LOGON userid NOIPL".

Note: If the user is the primary system operator, an automatic IPL is performed when he logs on.

The format of the IPL statement is:

```
Ipl   iplsys [PARM data]
```

where:

iplsys
is a one- to eight-character system name or the one- to three-digit I/O virtual address of the device containing the system to be loaded.

PARM data
indicates the data to be passed to the system being loaded. PARM may be followed by up to 48 characters of data excluding all leading and trailing blank characters, following the keyword PARM but including all imbedded blanks. This data is placed in your virtual machines general purpose registers (4 Bytes per register), starting with the high order position of GPR0.

Note: Trailing blanks are not truncated, but passed.

Notes:

1. *Although the IPL command allows up to 64 characters on the PARM option, the directory restricts each statement to a single card image. Consequently, limit the number of characters on the IPL control statement in the directory.*

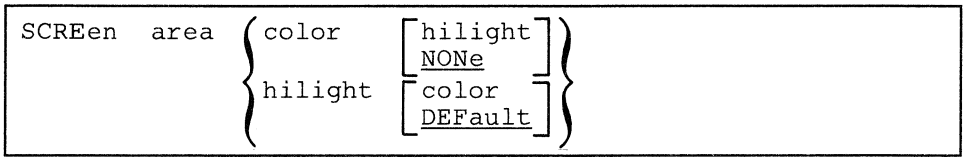
2. *The 'parmdata' on the IPL control statement is loaded into general registers 0-12 and is passed to the application specified by 'iplsys'.*

3. *You can specify the PMA or PMAV option on the IPL command, but not on the IPL control statement.*

SCREEN Control Statement

The SCREEN control statement defines the color and extended-highlight options for the user terminal. The SCREEN control statement is optional. If used, it must precede the first device statement (CONSOLE, MDISK, DEDICATE, LINK, or SPOOL).

The format of the SCREEN control statement is:



where:

area
specifies data to be highlighted or defined in color. Area may be:

ALL
entire screen. If this operand is specified, none of the other area specifications may be used.

INArea	Input area
STATarea	Status area
OUTarea	Output areas If this operand is specified, none of the other area specifications may be used (CPOut, VMOut, INRedisp).
CPOut	CP output
VMOut	Virtual machine output
INRedisp	Input redisplay.

color [**hilight** | **NONe**]
hilight [**color** | **DEFault**]
indicates color and extended-highlight attributes for the specified area. One color and/or one extended-highlight attribute must be entered for each area specified. The extended-highlight and color attributes are not positional. If both are specified, either may be first. Valid colors are:

BLUe	blue
RED	red
PINk	pink
GREen	green
TURquois	turquoise
YELlow	yellow
WHIte	white

Values for highlight are:

NONe	none
BLInk	blink
REVvideo	reverse video
UNDerlin	underline

Usage Notes:

1. Multiple SCREEN control statements are allowed. If you use multiple statements, they must be in a group with no other types of control statements between them.

2. Up to three areas can be defined on one SCREEN control statement.

3. A default value of NONE is applied for any unspecified extended-highlight attribute. DEFAULT is used for any unspecified color attribute. The DEFAULT color is monochrome (green and white).

4. If the ALL operand is used, it must be the first operand specified on the first SCREEN control statement for the user.

5. If the OUTAREA operand is used, CPOUT, VMOUT, or INREDISP may not be specified.

6. No SCREEN control statement operands may appear more than once within a user's directory entry.

Example:

The SCREEN control statement

```
SCREEN OUTAREA RED NONE INAREA GREEN BLINK STATAREA PINK
       UNDERLIN
```

results in the following assignments:

Area	Color	Highlight
Output areas (CP output, virtual machine output, and input redisplay)	Red	None
Input areas	Green	Blink
Status areas	Pink	Underline

CONSOLE Control Statement

The CONSOLE control statement specifies the virtual console.

The format of the CONSOLE control statement is:

```
Console   cuu devtype [class] [userid]
```

where:

cuu
is the virtual device address of one to three hexadecimal digits.

devtype
is the device type:

1052
3210
3215
3270

Specifying a device type of 3270 has the same effect as specifying a device type of 3215. To specify a 3270 virtual console, you must also issue the CP command TERMINAL CONMODE 3270. The devtype-3270 specification and TERMINAL CONMODE 3270 command apply only to local non-SNA display devices that have a 3270-compatible command set. These devices are:

- Model 138 console
- Model 148 console
- Model 158 console
- 3178
- 3277 (Model 2)
- 3278 (Models 2, 3, 4, 5 2A, 2C, and 3C)
- 3279 (Models 2A, 2B, 2C, 3A, and 3B)
- 3290
- 3036

Only one console can be specified. If a different console is sometimes required, use the CP DEFINE command to change the console address or add an alternate console.

For a 1052, 3210, 3215, or 3270 specification the system accepts any real console or terminal. 3270 is not supported for disconnected users; full screen channel programs will be command rejected when the user is disconnected.

class
is a one-character spooling class. A through Z and 0 through 9 are valid. The class governs printing of real spooled output. If the class operand is omitted, the default for console spooling is class T.

userid
is the one- to eight-character secondary userid whose console is to be used when the user disconnects.

If userid is specified, the class operand must also be specified.

Note: CP does not allow the user to mix the two types of subchannel protocols, shared and nonshared, on a single virtual control unit. At logon time, as the directory control statements for the user are processed, CP checks the devices represented by each CONSOLE, MDISK, DEDICATE, LINK, SPECIAL, and SPOOL statement for a possible conflict. For each directory control statement that violates the restriction, CP sends an error message to that user and does not create the virtual device. To avoid this problem, refer to "Matching Hardware to Shared or Nonshared Virtual Control Units" on page 396 for the listing of the protocols used by specific devices.

For more information about defining consoles, including a tutorial discussion, see *Virtual Machine Running Guest Operating Systems.*

MDISK Control Statement

The MDISK control statement describes the cylinder extent on a direct access device to be owned by the user. The DASD area assigned with this statement becomes the user's minidisk.

Note: The MDISK control statement cannot be used in a directory profile.

Warning: Neither CP nor the directory checks that minidisks defined with the MDISK statement do not overlap each other, and (for 3330, 3340, 3350, 3375, and 3380 disks) that they do not overlap the "alternate track" cylinders at the end of the real disk. If overlap occurs, file damage is inevitable.

The format of the MDISK control statement is:

```
Mdisk cuu devtype {cylr    cyls volser [mode [pr [pw [pm]]]]}
                  {T-DISK  cyls                              }
                  {blkr    blks                              }
```

where:

cuu
is the virtual device address of one to three hexadecimal digits.

devtype
is the device type:

2305
2311 Top (Top half of a 2314 or 2319)
2311 Bottom (Bottom half of a 2314 or 2319)
2314
2319
3330
3340
3350
3370
3375
3380
3380C
FB-512

For a 3350 device in native mode, specify 3350 as the device type. For a 3350 used in 3330 compatibility mode, specify 3330. Specify a 3344 disk as a 3340, and a 3333 as a 3330. For a 3330V system volume, specify 3330 as the device type.

If you specify a device type of 3380 for a minidisk in a 3880 Model 13 or Model 23 Storage Subsystem, all virtual I/O to that minidisk will bypass the subsystem cache. If you specify a device type of 3380C, the minidisk may access the cache.

{cylr / T-DISK / blkr}

is a three-digit decimal cylinder relocation factor that specifies the cylinder on a real disk that corresponds to cylinder 0 of the virtual disk. If T-DISK (temporary disk) is specified, temporary disk space is obtained at logon time from preallocated system disk space. This space must be initialized or formatted by the user when he logs on. It is a part of his virtual configuration until he logs off or detaches the disk. The data area is then returned for reallocation for another T-DISK area. To maintain security, this area should be erased before it is returned. blkr specifies the relocation factor in blocks for FB-512.

Note: It is not advisable to start a minidisk at real cylinder zero (unless the minidisk is to be used by OS ISAM, in which case it must begin at real cylinder zero). If you do assign a minidisk beginning at real cylinder zero, the user who owns it must realize that the minidisk label is the real label that both he and the VM/SP HPO system use to identify the disk. CP-owned volumes must not have minidisks beginning at real cylinder zero.

cyls

is a one- to three-digit decimal number specifying the number of cylinders.

blks

is a one- to six-digit decimal number specifying the number of FB-512 blocks.

Maximum Minidisk Sizes (cylinders or blocks)

Device Type	Model	CMS/VSAM	CMS 800-Byte Format	CMS 1K, 2K, or 4K Format
2314/2319	-	200 cyl	203 cyl	203 cyl
3310	-	126,016 cyl	Not supported	126,016 cyl
3330	1 or 2	404 cyl	246 cyl	404 cyl
3330	11	808 cyl	246 cyl	808 cyl
3333	1	404 cyl	246 cyl	404 cyl
3333	11	808 cyl	246 cyl	808 cyl
3340	35	349 cyl	348 cyl	348 cyl
3340	70	696 cyl	682 cyl	696 cyl
3350	Native mode	555 cyl	115 cyl	555 cyl
3370	A1 or B1	558,000 blk	Not supported	558,000 blk
3370	A2 or B2	712,752 blk	Not supported	71,752 blk
3375	-	959 cyl	182 cyl	959 cyl
3380	-	885 cyl	121 cyl	885 cyl
3380	AE4 or BE4	1770 cyl	121 cyl	1770 cyl

Note: The number of cylinders indicated for the 3344 is for each of the four logical 3340-70 devices.

If the device is a 2314 or 2319 and it is to be formatted by the Device Support Facility for 3375/3380, the minimum minidisk size is two cylinders. This is because, for these devices, the Device Support Facility reserves a cylinder at the end of every minidisk for alternate tracks. For other devices, the minimum size is one cylinder.

volser
is the volume serial number of the DASD volume (one to six alphameric characters).

mode
is the primary access mode requested for the device (read-only, write, or multiple-write), and the alternate access (read-only or write) desired (if any). An optional 'V' character, when appended to the mode request, specifies virtual reserve/release processing.

R
specifies that read-only (R/O) access is requested. The access is not given if any other user has the disk in write status.

RR
specifies that read-only access is requested, even if another user has the disk in write status.

W
specifies that write access is requested. The disk is not defined if any other user has the disk in read or write status.

WR
specifies that write access is requested if no other user has the disk in read or write status, but an alternate access of read-only is acceptable if others do have a link to the disk.

M
specifies that multiple access is requested. This means that a write link is to be given to the disk unless another user already has write access to it, in which case no link is to be given.

MR
specifies that a write link is to be given to the disk unless another user already has write access to it. In this case, a read link is given to the user and the "DEV xxxx FORCED R/O" message is issued.

MW
specifies that a write link is to be given to the disk in any case.

Warning: CMS does not protect a user from loss of data on a disk when multiple users have write access to the disk. More than one user writing to the same virtual device can result in a permanent loss of data. Users should not be linking with MW mode to obtain the M or MR function. (The M or MR access modes will allow only one write link to a disk.)

V

specifies that CP's virtual reserve/release support is to be used in the I/O operations for the specified device. The V is appended to the immediate right of the primary access mode specification (or the alternate access mode specification, if any). Thus, if the mode specified for a minidisk is MWV, the minidisk will function with write linkage using CP's virtual reserve/release function.

If a mode specification is omitted from the statement, it defaults to W.

pr

is the password that allows sharing in read mode (a one- to eight-character field).

pw

is the password that allows sharing in write mode (a one- to eight-character field).

pm

is the password that allows sharing in multiple-write mode (a one- to eight-character field).

Notes:

1. *A write password (pw) cannot be specified without a read password (pr). A multiple-write password (pm) cannot be specified without both a read password (pr) and a write password (pw).*

2. *If ALL is used for pr, pw, or pm, a user other than the owner of the minidisk is allowed to link to this minidisk without specifying a password.*

3. *When MSS support is used, the volume serial number may specify an MSS 3330V volume. In this case, the volume serial number must be six characters.*

4. *If the MSS communicator is initialized when the virtual machine logs on, and the system volume having a volume label of 'volser' is not mounted, CP attempts to find an available SYSVIRT 3330V and mount 'volser' on that device.*

5. *If virtual reserve/release processing is requested, minidisk users with read or write access are prevented from accessing a minidisk reserved by another virtual machine.*

6. *Do not specify T-DISK for the virtual=real user who will load MVS/SP when preferred machine assist or preferred machine assist with control switch assist is active. When that user issues the IPL command with the PMA or PMAV option, an error occurs because a virtual machine using preferred machine assist or preferred machine assist with control switch assist cannot use temporary disks.*

7. *Virtual disks used by the preferred virtual machine must be full-extent.*

8. *Protecting minidisks by specifying passwords on the MDISK statement provides additional security for your installation.*

9. *CP does not allow the user to mix the two types of subchannel protocols, shared and nonshared, on a single virtual control unit. At logon time, as the directory control statements for the user are processed, CP checks the devices represented by each CONSOLE, MDISK, DEDICATE, LINK, SPECIAL, and SPOOL statement for a possible conflict. This conflict occurs because the virtual control unit cannot support both shared and nonshared subchannel protocols at the same time. For each directory control statement that violates the restriction, CP sends an error message to that user and does not create the virtual device. To avoid this problem, refer to "Matching Hardware to Shared or Nonshared Virtual Control Units" on page 396 for the listing of the protocols used by specific devices.*

Examples:

```
MDISK 230 3330 5 10 WORK01 W ALL WRITE
```

is an MDISK statement for a minidisk with read/write access to 10 cylinders located on a real 3330 disk volume labeled WORK01, beginning at real cylinder 5. A user other than the owner of this minidisk can link to it in read status without specifying a read password, but must specify a password of 'WRITE' in order to gain write access to it.

```
MDISK 191 2314 50 15 CPDSK4 W RDPASS WRX2*
```

is an MDISK statement for a minidisk with read/write access to 15 cylinders located on a real 2314 labeled CPDSK4 starting at cylinder 50. A read password of RDPASS and a write password of WRX2* are provided. This allows the other users to access the minidisk through the directory LINK statement (see the description of the LINK statement in this section) or the LINK command.

```
MDISK 190 FB-512 75100 15748 FBACMS WR READ WRITE
```

is an MDISK statement for a minidisk with write access to 15748 FB-512 blocks on the real device labeled FBACMS. If the minidisk is already accessed to another user, read-only access is provided. The minidisk begins at relative block 75100 on FBACMS.

SPOOL Control Statement

The SPOOL control statement specifies the unit record device that is to be spooled. Multiple readers, punches, and printers may be specified, each on a separate SPOOL card.

The format of the SPOOL control statement is:

```
Spool  cuu devtype  [class] [ww ll] [2WCGM]  [CFS]  [DATCK]
                                    [4WCGM]  [BTS]  [NODATCK]
```

where:

cuu
is the virtual device address (one to three hexadecimal digits). The note that follows the description of ECMODE in the OPTION control statement describes a restriction on specifying the channel. For CMS, the following unit record addresses must be used:

PRINTER 00E
PUNCH 00D
READER 00C

devtype
is one of the following device types:

1403
2501
1443
3203
3211
2540R
2540P
3289
3525
3505
3800
3800-1
3800-3
4245
4248

Notes:

1. *2540P refers to the 2540 punch.*
2. *2540R refers to the 2540 reader.*
3. *If 3800 is specified, the 3800 Model 1 printer is assumed.*

class

is a one-character spooling class. The characters A through Z and 0 through 9 are valid. For spool input devices, such as a reader, an asterisk (*) may also be used.

For spool output devices, the class governs the punching or printing of the real spooled output. This operand is required for all output devices defined on the spool record. If this operand is omitted, the default class A is used.

For spool input devices, the class controls access to spool files by virtual card readers. The default class for readers is an asterisk (*), which means the reader can process any class of spool file.

For example

```
SPOOL 00E 1403 A
```

specifies a SPOOL record for a virtual 1403 at address 00E. The output class is A.

[ww ll]

specifies the physical characteristics of the paper to be loaded into the 3800 Model 1 or Model 3 printer. (ww) indicates the hexadecimal width code of the paper. The available form width codes are shown below.

Available Form	Width	Codes
X'01'	6-1/2 in.	(165 mm ISO)
X'02'	Reserved	(180 mm ISO)
X'03'	Reserved	
X'04'	8-1/2 in.	(215 mm ISO)
X'05'	Reserved	
X'06'	9-1/2 in.	(235 mm ISO)
X'07'	9-7/8 in.	(250 mm ISO)
X'08'	10-5/8 in.	(270 mm ISO)
X'09'	11 in.	(280 mm ISO)
X'0A'	12 in.	(305 mm ISO)
X'0B'	Reserved	(322 mm ISO)
X'0C'	Reserved	
X'0D'	13-5/8 in.	(340 mm ISO)
X'0E'	14-3/10 in.	(363 mm ISO)
X'0F'	14-7/8 in.	(378 mm ISO)

(ll)

indicates the decimal length of the paper. Specify (ll) as a decimal number using half-inches. If (ww ll) is not specified, 14-7/8 x 11 inches is assumed.

[2WCGM
4WCGM]

specifies the number of writable character generation modules (WCGM) assumed for the virtual 3800 printer. A WCGM is a 64-position portion of the 3800's character generation storage that holds the scan elements of one character set. A 3800 Model 1 can have either two or four WCGMs. A 3800 Model 3 can have only four WCGMs. If 2WCGM is specified, the virtual 3800 Model 1 printer is assumed to have two WCGMs. If 4WCGM is specified, four WCGMs are assumed. If neither is specified, 4WCGM is the default.

[CFS
BTS]

designates the stacker assumed for the virtual 3800 Model 1 or Model 3 printer. You may specify either CFS (Continuous Form Stacker) or BTS (Burster Trimmer Stacker). If neither is specified, CFS is assumed.

[DATCK
NODATCK]

specifies processing of certain virtual 3800 Model 1 or Model 3 data checks. If DATCK is specified, all data checks are reflected to the virtual machine (provided the 'BLOCK DATA CHECK' CCW has not been issued). If NODATCK is specified, only data checks that occur due to invalid translate table specifications or unmatched FCB codes are reflected to the virtual machine. This is the default condition.

Note: DATCK should be used only when necessary as it severely increases the overhead associated with simulation of WRITE and SKIP CCW's to the virtual 3800. In general, the reflection of data checks due to overprinting and invalid EBCDIC codes is not necessary.

Notes:

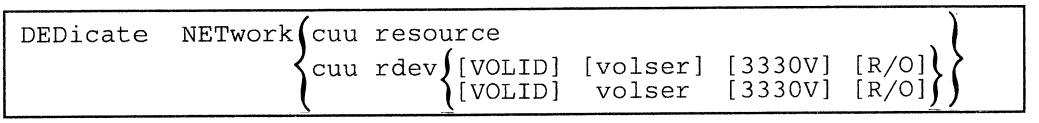
1. *All parameters of the SPOOL control statement are positional. They must be specified in the order shown.*

2. *At logon time, as the directory control statements for the user are processed, CP checks the devices represented by each CONSOLE, MDISK, DEDICATE, LINK, SPECIAL, and SPOOL statement for a possible conflict with the virtual control unit interface. This conflict occurs because the virtual control unit cannot support both shared and nonshared subchannel protocols at the same time. For each directory control statement that violates the restriction, CP sends an error message to that user and does not create the virtual device. To avoid this problem, refer to "Matching Hardware to Shared or Nonshared Virtual Control Units" on page 396 for the listing of the protocols used by specific devices.*

DEDICATE Control Statement

The DEDICATE control statement specifies that a real device is to be dedicated to this user. MSS 3330V (virtual 3330) volumes may be specified via the DEDICATE statement. If the device is a unit record device, input and output are not spooled by CP. A real device may be dedicated to only one user at a time. Should a device be specified as dedicated in more than one directory entry, only the first user to logon gains access to it.

The format of the DEDICATE control statement is:



where:

NETwork
is the keyword used if a remote 3270 Information Display System Printer (3284, 3286, 3287, 3288, or 3289) is to be automatically attached to a virtual machine at logon time. If this keyword is omitted, a local device is assumed.

Note: When a Network Printer is decicated, it must use a shared virtual control unit.

cuu
is the one- to three-character virtual device address.

resource
is the four-character resource id of a remote device as specified in DMKRIO. This operand *must* be specified if the NETwork keyword is specified, and is only valid if NETWORK is specified.

rdev
is the one- to four-character real device address.

VOLID
is the keyword that must be used if the volser is less than four characters long and rdev is not specified. It is optional when rdev is specified or volser is a length of four or more characters. In cases when rdev is not specified, the CP system will find an available rdev.

If the VOLID operand is used, the volume must be attached to the system when the user logs on. When he logs off, the operator can then detach the volume from the system.

volser
is the volume serial number of a disk pack mounted on some real disk storage device, or of an MSS volume to be dedicated to the virtual machine. The volser can be from one to six alphanumeric characters long.

3330V

specifies that all interruptions, including cylinder faults and attentions received on the rdev are to be passed to the virtual machine in its cuu.

R/O

specifies that the virtual device is to be in read-only status. If this operand is omitted, the status defaults to read/write.

Notes:

1. *When you dedicate a 2305 device, both the real and virtual device addresses must specify the first exposure on the 2305 (that is, device address 0 or 8). When you dedicate a 2305 or detach a dedicated 2305 from a user, all 8 exposures are processed.*

2. *Use caution in defining the hexadecimal addresses of virtual devices in DEDICATE statements, in order to avoid a usage conflict caused by control unit I/O interface protocol. Some devices use a shared subchannel protocol and others do not. See Appendix B, "Configuration Aid" on page 391 to find out whether a device uses a subchannel protocol. Devices should be grouped by control unit within a given channel according to their subchannel usage. CP does not permit you to group devices that use the shared subchannel protocol together with devices that do not use the shared protocol. The following is an example of a virtual machine's DEDICATE statements that are rejected at logon time.*

```
DEDICATE 10E 30E (30E is a real 3211)
DEDICATE 10F 30B (30B is a 2400 tape device)
```

The virtual addresses of both the 3211 and the tape device indicate the use of the same channel and control unit. By definition, the devices are virtual and will therefore share one virtual control unit (VCUBLOK) in CP. A real 3211 printer operates on a nonshared subchannel, and the real 2400 device is designed for shared subchannel operations. Both of these real devices are mapped to the same VCUBLOK. When the user logs on, the two DEDICATE statements cause an error message to be sent. The second virtual device (10F) is not created. When defining devices, you must therefore make sure the devices are defined (and separated) within their own control unit range and not shared with other devices.

Note that since there is no control unit in the real hardware for a system console, this restriction applies to any system console such as the 3036, 3278, and 3158.

Examples:

```
DEDICATE 0B8 0B0
```

is a DEDICATE statement for a device at real address 0B0. Its virtual address is 0B8.

```
DEDICATE 250 MYPACK
```

is a DEDICATE statement that defines, for this virtual machine, virtual address 250 as the real device where DASD volume MYPACK is mounted.

This restriction also applies to the CONSOLE, MDISK, SPECIAL, SPOOL, and LINK statements. The effects of the DEDICATE, LINK, and MDISK statements depend on the real device configuration at logon time.

Remote 3270 Information Display System Printers can also be attached by the NETWORK ATTACH command. For more details see the VM/SP HPO Operator's Guide.

A network attached printer will become an unshared virtual device. For more information, see Appendix D, "VM/SP HPO Restrictions."

3. *When the real device is a 3330V, the action CP takes in processing the DEDICATE statement at logon time depends on the combination of operands specified. Following are the allowable combinations and the control program action for each:*

```
DED cuu rdev
```

The real device must have the VIRTUAL feature (not SYSVIRT). The real device will be dedicated to the virtual machine as virtual device cuu, which is a 3330-1. All cylinder fault activity on the rdev will be processed by CP, transparently to the virtual machine.

```
DED cuu rdev 3330V
```

The real device must again be a VIRTUAL 3330V. All cylinder faults and unsolicited interrupts received by CP on the rdev will be passed to the virtual machine.

```
DED cuu VOLID volser
```

When processing this statement, the control program will allocate an available SYSVIRT 3330V and dedicate that real device to the virtual machine as virtual device cuu. The MSS volume having volser will be mounted on the real device, and the virtual device will be a 3330-1. This form of DEDICATE is used to dedicate volumes to non-MSS operating systems, such as CMS, since the control program chooses the real device address and no cylinder fault interrupts are passed to the virtual machine.

```
DED cuu rdev volser
```

The difference between this example and the one immediately preceding is that in this case the real device address is preselected and must have the VIRTUAL feature. This format allows the installation to control which real devices are dedicated to virtual machines, rather than having the control program choose a device address when the statement is processed.

```
DED cuu rdev volser 3330V
```

This format is the same as the preceding, except that the virtual device becomes a 3330V, such that CP does not intercept any cylinder fault interrupts or the associated attention interrupts.

4. *There are considerations that must be made when dedicating real 3330Vs to a virtual machine that also has a dedicated MSC port and is running an OS/VS operating system with MSS support. (See Appendix D, "VM/SP HPO Restrictions.")*

5. *When dedicating a real CTC, the CTC should be on a separate real channel from all other virtual devices because of a possible lock-out problem.*

6. *A dedicated 328x printer is a shared device. It is defined by CP as a 3210, which is a non-shared device. As a result, there can be no other shared devices on the subchannel.*

7. *For a virtual machine using preferred machine assist or preferred machine assist with control switch assist, the DEDICATE control statement should not violate restrictions that apply to the LINK, ATTACH, and DEFINE commands. If the DEDICATE control statement does violate those restrictions, an error occurs when the user attempts to execute the IPL command with the PMA or PMAV option. For more information, see "Preferred Machine Assist and Preferred Machine Assist with Control Switch Assist" on page 133.*

8. *At logon time, as the directory control statements for the user are processed, CP checks the devices represented by each CONSOLE, MDISK, DEDICATE, LINK, SPECIAL, and SPOOL statement for a possible conflict with the VCU interface. This conflict occurs because the virtual control unit cannot support both shared and nonshared subchannel protocols at the same time. For each directory control statement that violates the restriction, CP sends an error message to that user and does not create the virtual device. To avoid this problem, refer to "Matching Hardware to Shared or Nonshared Virtual Control Units" on page 396 for the listing of the protocols used by specific devices.*

LINK Control Statement

The LINK control statement makes a device that belongs to another user (userid) available to this virtual machine at logon time. If you want to make one volume available to several virtual machines:

- Define the volume for one of the virtual machines with an MDISK statement.
- Define a link to that volume, with the LINK statement for all other virtual machines that use the volume.

Then, if you later must move or change that volume, you need only update the one MDISK statement, the LINK statements need not be updated.

The format of the LINK control statement is:

```
Link   userid  cuu1 [cuu2 [mode]]
```

where:

userid
is the one- to eight-character user identification of the user to be linked to.

cuu1
is the virtual device address of the device owned by userid to be linked to (three hexadecimal digits). This is the virtual device address, assigned by userid, of the disk you wish to link to.

cuu2
is the virtual device address for the virtual machine being defined. If not specified, "cuu2" defaults to the same address as the linked-to device (three hexadecimal digits). If your virtual machine has the ECMODE option, any address up to X'FFF' is valid; otherwise, any address up to X'5FF' is valid.

mode
is the primary access mode requested for the device (read-only, write, or multiple-write), and the alternate access (read-only or write) desired, if any. They are:

R
specifies that read-only (R/O) access is requested. The link is not given if any other user has the disk in write status.

RR
specifies that read-only access is requested, even if another user has the disk in write status.

W
specifies that write access is requested. The disk is not defined if any other user has the disk in read or write status.

WR
specifies that write access is requested if no other user has the disk the disk in read or write status, but that an alternate access of read-only is acceptable if others do have a link to the disk.

M
specifies that multiple access is requested. This means that a write link is to be given to the disk unless another user already has write access to it, in which case no link is to be given.

MR
specifies that a write link is to be given to the disk unless another user already has write access to it. In this case, a read link is to be given to the user, and the "DEV xxxx FORCED R/O" message is issued.

MW
specifies that a write link is to be given to the disk in any case.

Note: If the mode is not specified, the default is R.

Warning: CMS supports multiple read-only access to disks in full. CMS does not support write access to disks by multiple users. CMS does not protect a user from loss of data on a disk when multiple users have write access to the disk. More than one user writing to the same virtual device can result in a permanent loss of data. CMS disks should never have more than one existing write link at a time.

A disk accessed in write mode by one CMS user is available to other CMS users in read-only mode, but those files altered by the write-mode user cannot be read by the other users.

It is the responsibility of the operating system executing in each virtual machine to keep data from being destroyed or altered on shared disks.

CMS disks should never be linked in write mode to more than one user. If two or more CMS users have write access to the same disk, all data on the disk may be destroyed.

No more than two levels of indirect directory links are permitted. For example, if userA owns a virtual device that was obtained via a directory MDISK statement:

```
MDISK 100 3330 5 10 VMDISK W READ WRITE
```

Then userB may have a directory LINK statement to obtain this device at logon time:

```
LINK userA 100 200 RR
```

Any number of users may have directory LINK statements to either userA's or userB's device. However, if userC has a directory LINK to userB's device (which was obtained by a LINK to userA's device):

```
LINK userB 200 300 RR
```

then no user can obtain a LINK (either through a directory LINK statement or the LINK command) to this device through userC because more than two levels of indirect directory links are in use.

Notes:

1. *For a virtual machine using preferred machine assist or preferred machine assist with control switch assist, the LINK control statement should not violate restrictions that apply to the LINK command. If the LINK control statement does violate those restrictions, an error occurs when the user attempts to execute the IPL command with the PMA or PMAV option. For more information, see "Preferred Machine Assist and Preferred Machine Assist with Control Switch Assist" on page 133.*

2. *At logon time, as the directory control statements for the user are processed, CP checks the devices represented by each CONSOLE, MDISK, DEDICATE, LINK, SPECIAL, and SPOOL statement for a possible conflict with the VCU interface. This conflict occurs because the virtual control unit cannot support both shared and nonshared subchannel protocols at the same time. For each directory control statement that violates the restriction, CP sends an error message to that user and does not create the virtual device. To avoid this problem, refer to "Matching Hardware to Shared or Nonshared Virtual Control Units" on page 396 for the listing of the protocols used by specific devices.*

SPECIAL Control Statement

The SPECIAL control statement specifies the I/O units available to the user that need not have a real I/O unit available. Special devices are program simulated devices that may or may not be connected to real or virtual devices after the user has logged off.

The format of the SPECIAL control statement is:

```
SPEcial  cuu  devtype  [IBM]
                        [TELE]
```

where:

cuu
is a one- to three-character virtual device address.

devtype
is the device type:

2701
2702
2703
3088
3138 (virtual 3138 console)
3148 (virtual 3148 console)
3158 (virtual 3158 console)
3270 (virtual 3270 only)
CTCA (channel-to-channel adapter)
TIMER (pseudo-timer device)

The above virtual devices created by the SPECIAL statement require a nonshared virtual control unit, except the 3270 device type.

[IBM]
[TELE]
valid only if devtype is 2701, 2702, or 2703

For example, a virtual machine executing a multiple-access system that supports four IBM Type 1 adapter lines, would have four SPECIAL entries, one for each of those addresses. This provides a virtual 270x line to allow a user to dial this multiple-access system rather than logging on as a separate virtual machine.

Note: At logon time, as the directory control statements for the user are processed, CP checks the devices represented by each CONSOLE, MDISK, DEDICATE, LINK, SPECIAL, and SPOOL statement for a possible conflict with the VCU interface. This conflict occurs because the virtual control unit cannot support both shared and nonshared subchannel protocols at the same time. For each directory control statement that violates the restriction, CP sends an error message to that user and does not create the virtual device. To avoid this problem, refer to "Matching Hardware to Shared or Nonshared Virtual Control Units" on page 396 for the listing of the protocols used by specific devices.

Directory Entries for CMS/DOS

The VSE system and private libraries are accessed in read-only mode under CMS/DOS. If more than one CMS virtual machine is using the CMS/DOS environment, you should update the directory entries so the VSE system residence volume and the VSE private libraries are shared by all CMS/DOS users.

The directory entry for one CMS virtual machines should contain MDISK statements defining VSE volumes. Directory entries for other CMS/DOS users should contain LINK statements.

For example, assume the VSE/AF system libraries are on cylinders 0-149 of a 3330 volume labeled DOSRES. Also, assume the VSE/AF private libraries are on cylinders 0-99 of a 2314 volume labeled DOSPRI. Then one CMS machine (for example, DOSUSER1) would have the MDISK statements in its directory entry.

```
USER DOSUSER1 password 320K 2M G
     .
     .
     .
   MDISK 331 3330 0 150 DOSRES R rpass
   MDISK 231 2314 0 100 DOSPRI R rpass
```

All other CMS/DOS users would have links to these disks. For example:

```
LINK DOSUSER1 331 331 R rpass
LINK DOSUSER1 231 231 R rpass
```

Note: *Refer to the* VM/SP HPO System Definition Files *for a list of the sample directories supplied with the product tape.*

Chapter 20. Defining Your Own User Classes

Note: *When you install Release 4.0 or any later release, you must take special steps if you have made any changes to CP commands. Consult the chapter "How to Add a Command" in the* VM/SP HPO System Programmer's Guide *if you have:*

- *Defined new CP commands*
- *Defined new subcommands for existing CP commands*
- *Changed the abbreviations for CP commands*
- *Redefined user privilege classes*

CP commands are a set of interactive console functions that an operator uses to control the VM/SP HPO system. Every CP command belongs to at least one of the eight IBM-defined user classes. Each class includes tasks performed by a different type of user.

IBM User Class	User Type	Primary User
A	O	Primary System Operator
B	R	System Resource Operator
C	P	System Programmer
D	S	Spooling Operator
E	A	System Analyst
F	C	Service Representative
G	G	General User
Any	None	Any user

The figure "CP Privilege Class Descriptions" in the *VM/SP HPO Operator's Guide* shows which functions belong to each IBM-defined class.

Each user is assigned one or more classes in the VM directory, according to his needs. He may issue only those commands and DIAGNOSE codes that belong to his assigned user classes, plus a few commands available to all classes, called "class ANY" commands. Commands and codes belonging to other classes are not available. For example, if someone who is assigned only the General User class tries to use a command from the System Analyst class, the command will not work. In this way, you can protect your installation against misuse by restricting certain functions to authorized users.

Some commands have different sets of operands for different classes. One such command is the QUERY command. A variety of QUERY operands such as READER, PRINTER, and TIME are available for general users. All users need information about spooled reader or printer files; all users need to know the time of day. Some operands, however, are available only to those who need them. For example, only the primary system operator and the system analyst need to know the priority of another userid, so only they may use the PRIORITY operand.

To distinguish among different versions of a command, at least one one-character User Type is associated with each operand of each command. These types correspond to the same functional groups as the IBM-defined classes; but no matter how often you change the classes, the types remain constant. Class ANY commands, which are available to all users, cannot be reclassified and have no type.

If the eight-class structure supplied by the system is not flexible enough for your installation, the user class restructure (UCR) option permits you to:

- Subdivide user classes. You can define up to 32 different user classes, depending on how restrictive you want to be. You may have many users, each one performing a few very specialized tasks. Give each user access only to the commands necessary for his work.

- Combine user classes. The users of your system may need to perform a wide variety of tasks. For example, your programmers may also need to handle the spooling and analyst functions. Give them access to the commands for all three functions.

- Create new user classes including selected commands from several of the IBM-defined classes.

Planning Your New Class Structure

Before you decide how to set up your new user class structure, analyze your present structure.

1. Determine how many different groups of users your installation has and the tasks each group has to perform.

2. Consider what it would be efficient for them to do as well as what they are now permitted to do.

3. Draw up a worksheet for a representative user of each group. List the commands and DIAGNOSE codes necessary for such a user's work. You will find a complete list of commands in the *VM/SP HPO CP Command Reference for General Users* or in the *VM/SP HPO Operator's Guide*. DIAGNOSE codes are listed in the *VM/SP HPO System Programmer's Guide*.

   The worksheet in Figure 32 lists the needs of a sample group of users: inexperienced application programmers. Use this worksheet as a guide.

   Remember that some commands have several sets of operands, identified by type. Remember, too, that some CP commands are almost the same as directory control statements. Don't make these commands unavailable to users who need them by putting them in a restricted class. For example, any user whose directory entry contains IPL or LINK control statements should be allowed to issue the IPL and LINK commands.

Command or Code	IBM User Class	User Type	New User Class
QUERY	C	P	I
QUERY	G	G	G
INDICATE	E	A	I
ATTN	G	G	G
ADSTOP	G	G	G
BEGIN	G	G	G
SET	G	G	G
DISPLAY	G	G	G
IPL	G	G	G
LINK	G	G	G
LOGOFF	ANY	-	ANY
LOGON	ANY	-	ANY
TRACE	G	G	G

Figure 32. Sample Worksheet for Planning a New Class Structure

4. Look at the commands and codes you have listed for the various users. What patterns do they fall into?

   - Are certain commands and codes usually listed together; that is, do people who need one usually need them all? If so, these commands and DIAGNOSE codes should probably be in the same class.

   - Have some useful commands and codes been unavailable to certain users because they could not be given access to all the commands in one of the IBM-defined classes? Have you been allowing users access to commands they should not use in order to allow them to use a few commands and DIAGNOSE codes they need to do their work? If your installation has either of these problems, put the commands and codes whose use should be restricted in different classes from the commands and codes to which many users can be given safe access.

5. Note any changes you plan to make to command and code classes on the worksheet. In the sample worksheet in Figure 1, the planner proposes to create a new class, I, containing certain commands from the IBM-defined classes C and E that inexperienced application programmers are to be permitted to use.

6. When you have decided which commands and DIAGNOSE codes belong together, count your new classes to make sure that there are no more than 32. Make sure that you have assigned each one a unique identifying character, a letter from A to Z or a digit from 1 to 6. You are now ready to set up your new class structure.

Changing the Class Structure

Note: Before attempting to change the IBM-defined class structure, wait until VM/SP HPO Release 4.2 has been completely installed. Trying to install Release 4.2 with your own user classes can lead to needless difficulties in the migration process.

After you have installed Release 4.2 and planned the structure of your new classes, you must take the following steps to override the IBM-defined classes:

1. Allocate OVRD space on a CP-owned volume to allow room for the directory's internal override file.

2. Create a class-override file containing your changes to the IBM-defined classes.

3. Run the override program.

4. Update your directory to assign your new classes to users.

5. IPL the system.

6. Update the HELP files if you have changed the commands available to general users.

7. Consult the *VM/SP HPO System Programmer's Guide* to see how to record your changes.

To see these steps explained in detail, consult the *VM/SP HPO System Programmer's Guide* .

The Override Program

The program that overrides the IBM-defined classes, DMKOVR, is invoked as a CMS application program by the OVERRIDE command. This program processes the control statements in the class-override file. It defines the changes you wish to make to the class structure, and makes them available to the VM/SP HPO directory by converting them to an internal format in the directory pages. If there is already an internal override file in the directory, the class-override file replaces it. A pointer to the address of the internal override file is stored in the directory volume label. If there are no overrides, the pointer is zero.

OVERRIDE Command

The OVERRIDE command invokes DMKOVR to implement your changes to the class structure. Its format is:

```
OVERRIDE fn [ft [fm]]  [(EDIT]
                       [(FREE]
```

where:

fn
is the filename of the class-override file.

ft
is the filetype of the class-override file. The default is OVERRIDE.

fm
is the filemode of the class-override file. The default is *.

EDIT
specifies that the file is to be examined but not changed.

FREE
specifies that the pointer to the internal override file in the VM directory label is to be set to zero. Use FREE if you wish to return to the IBM-defined class structure.

Specifying neither EDIT nor FREE converts the class-override file to internal directory format.

Note: The new overrides will not take effect until the next time you IPL the system.

The Class-Override File

Create a class-override file only if you wish to change the existing user privilege class structure. The class-override file may have any name you choose, and must contain:

- A DESTINATION control statement identifying the the virtual representation of the CP-owned volume that is to contain the directory's internal override file.
- OVERRIDE control statements defining the changes to the classes of commands and DIAGNOSE codes that you wish to make.

DESTINATION Control Statement

The DESTINATION control statement identifies the virtual representation of the CP-owned volume that will contain the internal override file. This must be the same volume that contains the directory. The DESTINATION control statement must be the first statement in the class-override file. The format of the DESTINATION control statement is:

```
DESTination cuu devtype volser
```

where:

cuu
is the virtual address of the device containing the internal override file, specified in hex digits.

devtype
indicates the type of device containing the internal override file (2314, 2319, 2305, 3330, 3340, 3350, 3375, 3380, or FB-512).

volser
is the volume serial number of the CP-owned volume (one to six alphameric characters).

OVERRIDE Control Statement

The OVERRIDE control statement changes the user privilege class to which a command or DIAGNOSE code belongs. One OVERRIDE control statement is required for each command or DIAGNOSE code being assigned to a new class. The format of the OVERRIDE control statement is:

```
command [Type=c] Class={classes}
                       {*      }
```

where:

command
is the name of the command or DIAGNOSE code. It must be the first parameter in the control statement.

Type=c
indicates the User Type associated with a command. Specify the User Type only if the command belongs to more than one function group (that is, if it has more than one set of permitted operands). Diagnose codes have no User Type; therefore, if you specify a User Type for a DIAGNOSE code, you will get an error message.

You may specify only one User Type per OVERRIDE control statement. If you wish to assign more than one version of a command to new classes, you must enter an OVERRIDE control statement for each User Type.

If you specify the User Type for a command that has only one User Type, the Type field will be ignored.

Class=
indicates the new class or classes to which you are assigning the command or DIAGNOSE code. This parameter is always required.

classes
is a string of 1 to 32 class identification characters: any of the letters from A to Z or the digits from 1 to 6. Each character represents a single privilege class. The class identification characters may appear in any order, without duplication, and must not be separated by blanks.

*****
indicates that the command or DIAGNOSE code is available to all users, whatever their user privilege class.

Directory Control Statements for Defining User Classes

If you define your own user class structure, your new classes must be reflected in the VM/SP directory. If you wanted to implement the changes planned on the sample worksheet in Figure 32 on page 232, you would have to update your directory so that:

- Inexperienced application programmers are assigned both Class G and Class I.
- Users formerly assigned Class C or Class E are now assigned Class I as well.

When you define a user class structure, your VM/SP directory is a little different from the directory of a system with IBM-defined classes only, as described in Chapter 19, "Creating Your Directory." As with IBM-defined classes, the USER control statement defines a virtual machine and creates a VM/SP HPO directory entry for each user; but it varies slightly in that the class operand may assign non-IBM-defined classes or contain an asterisk to show that the CLASS control statement follows. The format of the USER control statement is shown in "USER Control Statement" on page 198.

You may also need to add the CLASS control statement to some entries. The CLASS control statement assigns classes to each user. It may be used to assign eight or fewer classes to one user, and must be used to assign more than eight classes to one user. The format of this control statement is shown in "CLASS Control Statement" on page 202.

Chapter 21. Preparing the Real I/O Configuration File (DMKRIO)

The real I/O configuration file consists of macros that describe the I/O devices, control units, and channels attached to the real System/370. CP uses this information to schedule I/O operations and to allocate resources. Therefore, the real I/O macro entries must represent the real hardware configuration accurately. Generally, there must be one real I/O macro entry for each hardware unit in your configuration.

Make note of the following before coding the real I/O configuration file:

- You can include entries for more devices than your installation has so that devices can be added without performing another system generation; but, bear in mind that the control blocks generated (RDEVBLOK, RCUBLOK, and RCHBLOK) occupy space in real storage.

- If you are creating a real I/O configuration file for a VM/SP HPO system that will use preferred machine assist or preferred machine assist with control switch assist support, leave some channels undefined. Leaving channels undefined allows the preferred virtual machine to define and exclusively own those channels. VM/SP HPO or other virtual machines cannot use those channels. If you anticipate times when VM/SP HPO will not use preferred machine assist or preferred machine assist with control switch assist support, you may wish to create a second real I/O configuration file that defines all channels. After generating VM/SP HPO with this second file, VM/SP HPO and its virtual machines can use channels that otherwise the preferred virtual machine would use. (See "Preferred Machine Assist and Preferred Machine Assist with Control Switch Assist" on page 133.)

- If you are generating a multiprocessor system, a COPY OPTIONS statement is required in DMKRIO prior to assembly. Refer to Chapter 12, "Attached Processors and Multiprocessor Systems."

- For the 308x Processor Complex, in addition to preparing the real I/O configuration file you must prepare the input/output configuration program source file and execute the Input/Output Configuration Program to define the I/O configuration to the processor. See "Considerations for Coding the Input/Output Configuration Source File" in this chapter for more information.

When preparing the RDEVICE and RCTLUNIT entries, refer to Appendix B, "Configuration Aid" on page 391 to assist you in configuring control units and devices. Following the descriptions of the CLUSTER, TERMINAL, RDEVICE, RCTLUNIT, RCHANNEL, and RIOGEN macros, there is an example showing how these macros are coded for one particular real configuration.

The macros, in their proper sequence, are:

Macro Name	Units Referred To
{CLUSTER / TERMINAL}	Remote Display Stations
.	
.	
.	
RDEVICE	I/O Devices
RCTLUNIT	Control Units
RCHANNEL	Channels
RIOGEN	System Console

The file should be created in the order shown:

```
DMKRIO CSECT
       CLUSTER macro
       TERMINAL macro
            .
            .
       RDEVICE macros
            .
            .
       RCTLUNIT macros
            .
            .
       RCHANNEL macros
            .
            .
       RIOGEN macro
       END
```

Note: There must be a CLUSTER macro for each 3270 control unit for remote 3270s. Each CLUSTER macro must be followed immediately by the TERMINAL macros representing each display station and printer on that control unit. The CLUSTER and TERMINAL macro groups must precede all other real I/O configuration macros. See special requirements for TERMINAL macros for devices attached to the 3274 Model 1C under "Coding the Real I/O Configuration Macros for Remote 3270s" on page 243.

All groups of CLUSTER and TERMINAL macros must appear first, followed by all RDEVICE macros, all RCTLUNIT macros, all RCHANNEL macros, and finally by the RIOGEN macro. In addition, the first statement in the file must be the DMKRIO CSECT statement (as shown) and the last statement must be the assembler END statement.

Coding the Real I/O Configuration Macros for Remote 3270s

Two types of remote 3270 configurations are supported: a cluster control unit with multiple terminals and printers attached and standalone display stations. The clustered configurations attach to either a 3271, 3274 Model 1C, or 3276 control unit. The standalone station is a 3275 display station which contains its own built-in control unit. All remote configurations are attached via binary synchronous communication lines.

To define remote 3270 stations you must code CLUSTER, TERMINAL, and RDEVICE macros. Code one RDEVICE macro for each binary synchronous line that supports a remote 3270 configuration. Code one CLUSTER macro to define the 3270 control unit for each of those lines and code one or more TERMINAL macros, as needed, to define the devices in the remote 3270 configuration.

The CLUSTER macro defines the control unit (3271, 3274 Model 1C, 3275, or 3276) for the remote 3270 configuration. Each CLUSTER macro must have a unique label. This label is coded on the RDEVICE macro that defines the corresponding binary synchronous line and logically links the line and the cluster. The address of the line (defined by the ADDRESS=ccuu operand of the RDEVICE macro) is coded in the LINE=ccuu operand of the CLUSTER macro.

Follow each CLUSTER macro with the TERMINAL macros that define the terminals for the remote 3270 control unit. For the 3271 and 3276 directly following the CLUSTER macro, code a TERMINAL macro for each terminal address to which a terminal can be attached (regardless of whether or not the intermediate addresses are unused). For example, if terminals are attached to the third, fourth, and eighth addresses, you code eight TERMINAL macros. The first macro represents the first (lowest) address, the last represents the eighth (highest) address.

For the 3274 Model 1C that has only 3278s, 3279s, 3290s (attached via Terminal Adapter Types A1, A2, or A3), 3287s, or 3289s attached, follow the same procedure as for the 3271 and 3276 in coding the TERMINAL macros. If the 3274 Model 1C has 3277s, 3284s, 3286s, 3287s (attached via Terminal Adapter Types B1, B2, B3, or B4), or 3288s attached, directly following the CLUSTER macro, first code TERMINAL macros for all 3278s, 3279s, 3287s, 3290s (attached via Terminal Adapter Types A1, A2, or A3), and 3289s. These devices must occupy the first 8, low-order addresses, and each following block of 8 addresses until all of these devices are attached. As before, a TERMINAL macro must be coded for all unused addresses in each block of 8 addresses that are required. Immediately following the last TERMINAL macro in the block of 8, 16, or 24, code a TERMINAL macro for each 3277, 3284, 3286, 3287s (attached via Terminal Adapter Types B1, B2, B3, or B4), and 3288 that can be attached. These devices will occupy the higher-order addresses on the controller. Again, a TERMINAL macro must be coded for each unused address to which a terminal can be attached up to the last address occupied.

For the 3275, directly following the CLUSTER macro, code a single TERMINAL macro specifying TERM=3275. If the 3275 has a 3284 or 3286 Model 3 Printer attached, specify MODEL=3 to define the printer; otherwise, the printer is ignored.

After all CLUSTER-TERMINAL groups of macros have been coded, code the other real I/O configuration macros. You must code an RDEVICE macro for each binary synchronous line that supports remote 3270 stations. Specify the label of the corresponding CLUSTER macro on the RDEVICE macro (CLUSTER=label).

CLUSTER Macro

Use the CLUSTER macro to define a control unit associated with a remote 3270. Each CLUSTER macro represents a display control unit (a 3271, 3274 Model 1C or 51C, or 3276) on a leased BSC line, or a standalone 3275 on either a switched or leased BSC line. One CLUSTER macro must be specified for each 3271, 3274 Model 1C, 3275, and 3276.

Note: Each CLUSTER macro must immediately precede the TERMINAL macros defining the devices attached at each remote 3270 station. The groups of CLUSTER and TERMINAL macros must precede all other macros in the DMKRIO file.

The format of the CLUSTER macro is:

Name	Operation	Operands
label	CLUSTER	CUTYPE={3271 / 3274 / 3275 / 3276} ,GPOLL=cudv ,LINE=ccuu ,DIAL={YES / NO}

where:

label
is a name of the CLUSTER macro. It must be specified. The label may be any valid assembler language symbol. The label establishes a unique symbolic name for this cluster control unit or stand alone station.

CUTYPE = {3271 / 3274 / 3275 / 3276}
is the station control unit. It is either 3271, 3274 Model 1C or 51C, 3275, or 3276.

Note: VM does not support multi-drop control units on a single line defined by a CLUSTER/TERMINAL group.

GPOLL = cudv
are the general polling characters that represent the general polling technique to be used for this station. When general polling is used, the first device ready to send data over the line is allowed to do so. The characters, cudv, are the 4-digit hexadecimal general polling characters assigned to the station control unit. The hexadecimal equivalent of the EBCDIC transmission code is in the form cudv, where:

cu
are the polling characters for the control unit.

dv
are the characters for any available input device.

The general polling characters for the remote 3270 device (dv) are always X'7F' and the general polling characters for the control unit are defined when the control unit is installed. Use Figure 33 to determine what you should code as the general polling characters for the control unit. GPOLL is ignored if CUTYPE = 3275 and DIAL = YES are specified.

Note: The 3274 and 3276 terminal control unit address switches are set by the user to match polling and selection address characters shown in Figure 33.

LINE = ccuu
is the line interface address. It is the address specified on the RDEVICE macro associated with this CLUSTER macro.

DIAL = { YES / NO }
specifies whether the 3275 has the Dial feature. DIAL = NO must be specified if CUTYPE = 3271.

Examples

The following CLUSTER macro describes a 3271 control unit with a control unit address of 2 and a line address of 078.

```
CLUST001 CLUSTER CUTYPE=3271,GPOLL=C27F,LINE=078,DIAL=NO
```

The following CLUSTER macro describes a 3275 display station (without the Dial feature) that has a control unit address of 0 and a line address of 080.

```
CLUST020 CLUSTER CUTYPE=3275,GPOLL=407F,LINE=080,DIAL=NO
```

In the real I/O configuration file (DMKRIO), the CLUSTER macro must immediately precede TERMINAL macros that define stations attached to that cluster or standalone station.

TERMINAL Macro

Use the TERMINAL macro to define:

- A display station or printer that is attached to the remote 3270 display system or
- A terminal address that is available to attach an additional remote 3270.

Each terminal address attached to a cluster must be represented by a TERMINAL macro. Only one TERMINAL macro is specified for a stand-alone 3275 display station.

Code one TERMINAL macro for each display device and each 5K printer attached to a cluster control unit (3271, 3274 Model 1C, or 3276). You must code a TERMINAL macro for every terminal address to which a terminal can be attached, even if a terminal address is unused. When you code a TERMINAL macro for an unused terminal address, specify a valid TERM= operand and the correct selection or addressing characters.

An adapter card position must be present in the control unit for any terminal address generated, whether a terminal is physically attached or not. Failure to meet this requirement will result in timeouts with remote device type 3274 and 3276 control units.

For a 3274 Model 1C or 51C Control Unit that has 3277s, 3284s, 3286s, 3287s (attached via Terminal Adapter Types B1, B2, B3, or B4), or 3288s attached, code a TERMINAL macro for all 3278s, 3279s, 3287s, 3289s, and 3290s in groups of 8 until all 3278s, 3279s, 3287s, 3289s, and 3290s have been included. You must code a TERMINAL macro for every terminal address in each group of 8. Following these macros, code a TERMINAL macro for each 3277, 3284, 3286, 3287, or 3288. Again, you must code a TERMINAL macro for every terminal address to which a terminal can be attached.

Code only one TERMINAL macro to define the display station, and optionally a printer, attached to a stand-alone station (3275). Since a 3276 is a cluster controller and not a stand-alone, code each 3276 with a TERMINAL macro. Code TERM=3275 to define the 3275 display station and, optionally, code MODEL=3 to define a 3284 or 3286 printer attached to the 3275.

The format of the TERMINAL macro is:

Name	Operation	Operands
	TERMINAL	TERM={3275 / 3276 / 3277 / 3278 / 3279 / 3284 / 3286 / 3287 / 3288 / 3289 / 3290} ,SELECT=cudv [,MODEL=1 / ,MODEL=2 / ,MODEL=3 / ,MODEL=4 / ,MODEL=5] [,FEATURE=OPRDR]

Note: All TERMINAL macros defining devices attached to a remote 3270 station must follow the CLUSTER macro that defines the control unit for that station. Groups of CLUSTER and TERMINAL macros must precede all other macros in the DMKRIO file.

where:

TERM = {**3275** / **3276** / **3277** / **3278** / **3279** / **3284** / **3286** / **3287** / **3288** / **3289** / **3290**}

is the device type of the remote 3270 station attached to the clustered or stand-alone 3270 control unit. If TERM = 3276, 3278, or 3279, MODEL= *must* be specified.

SELECT = cudv

are the 4-digit hexadecimal selection or addressing characters assigned to this device, where:

cu

are the characters for the control unit.

dv

are the characters for the device.

Use Figure 33 to determine the selection and addressing characters for this device. The SELECT operand is ignored if DIAL = YES is specified for the 3275 in the CLUSTER macro.

Note: If a printer is attached to the 3275, it has the same address as the 3275 display station.

Use this column for: • Device selection • Specific poll • General poll • Fixed return addresses		Use this column for: • 3270 control unit selection addresses	
If the Control Unit or Device Number is:	**The EBCDIC Code (in hexadecimal) is:**	**If the Control Unit Number is:**	**The EBCDIC Code (in hexadecimal) is:**
0	40	0	60
1	C1	1	61
2	C2	2	E2
3	C3	3	E3
4	C4	4	E4
5	C5	5	E5
6	C6	6	E6
7	C7	7	E7
8	C8	8	E8
9	C9	9	E9
10	4A	10	6A
11	4B	11	6B
12	4C	12	6C
13	4D	13	6D
14	4E	14	6E
15	4F	15	6F
16	50	16	F0
17	D1	17	F1
18	D2	18	F2
19	D3	19	F3
20	D4	20	F4
21	D5	21	F5
22	D6	22	F6
23	D7	23	F7
24	D8	24	F8
25	D9	25	F9
26	5A	26	7A
27	5B	27	7B
28	5C	28	7C
29	5D	29	7D
30	5E	30	7E
31	5F	31	7F

Figure 33. Remote 3270 Control Unit and Device Addressing

[MODEL=1
MODEL=2
MODEL=3
MODEL=4
MODEL=5]

is the model number of the terminal or printer. The default is model 2.

Note: If TERM=3276, 3278, or 3279, MODEL= must be specified, and should be equal to the actual model of the real device. If the model specification does not match the real device, unpredictable results may occur.

If TERM=3290, model number must not be specified.

The following is a list of terminals and their model numbers:

3275	Model 2
3276	Model 2, 3, or 4
3277	Model 2
3278	Model 2, 3, 4, or 5
3279	Model 2 or 3
3284	Model 2 or 3
3286	Model 2 or 3
3287	Model 1 or 2
3288	Model 2
3289	Model 1 or 2

Note: If TERM= 3276 3278, or 3279 the model number 2, 3, 4, or 5 must be specified.

The following printers can be attached to a 3271, cluster control unit:

- IBM 3284 Printer, Model 2
- IBM 3286 Printer, Model 2
- IBM 3287 Printer, Models 1 and 2
- IBM 3288 Printer, Model 2

The following printers can be attached to a remote 3274 Model 1C cluster control unit:

- IBM 3284 Printer Model 2
- IBM 3286 Printer Model 2
- IBM 3287 Printer Models 1 and 2
- IBM 3288 Printer Model 2
- IBM 3289 Printer Models 1 and 2

The following printers can be attached to a 3276 cluster control unit:

- IBM 3287 Printer Models 1 and 2
- IBM 3289 Printer Models 1 and 2

The following printers can be attached to a standalone 3275 station:

- IBM 3284 Printer, Model 3
- IBM 3286 Printer, Model 3 (via RPQ MB4317)

FEATURE=OPRDR
specifies the optional feature, operator identification card reader, available on the 3277 Display Station, Model 2, or the magnetic slot reader on a 3276; 3278 Display Station, Models 2, 2A, 3, 4, and 5; or 3279 Color Display Station, Models 2A, 2B, 2X, 3A, 3B, 3X, S2A, S2B, S3G.

Examples

Example 1: This TERMINAL macro describes a 3277 with a selection address of 2, and a control unit address of 2.

```
TERMINAL TERM=3277,SELECT=E2C2,FEATURE=OPRDR
```

Example 2: This TERMINAL macro describes a 3286 with a selection address of 3 and a control unit address of 3.

```
TERMINAL TERM=3286,SELECT=E3C3
```

Example 3: This TERMINAL macro describes a 3284 with a selection address of 4 and a control unit address of 4.

```
TERMINAL TERM=3284,SELECT=E4C4,MODEL=2
```

Example 4: This TERMINAL macro describes a 3275 Display Station with a 3284 Printer, Model 3, attached and a control unit address of 0.

```
TERMINAL TERM=3275,SELECT=6040,MODEL=3
```

If no printer is attached to the 3275, code:

```
TERMINAL TERM=3275,SELECT=6040
```

Figure 34 shows some examples of valid polling characters. For more information on polling sequences, see *IBM 3270 Information Display System Component Description.*

3271, 3274, and 3276 Addressing			3275 Addressing		
General Poll for Control Unit 5	Control Unit Address	EBCDIC C5 C5	General Poll for Control Unit 5	Control Unit Address	EBCDIC C5 C5
	Device Address	7F 7F		Device Address	7F 7F
Specific Poll Device 4 on Control Unit 5	Control Unit Address	C5 C5	Specific Poll for Control Unit 5	Control Unit Address	C5 C5
	Device Address	C4 C4		Device Address	40 40
Select Device 4 on Control Unit 5	Control Unit Address	E5 E5	Select Control Unit 5	Control Unit Address	E5 E5
	Device Address	C4 C4		Device Address	40 40

Figure 34. Examples of Remote 3270 Addressing

RDEVICE Macro

Use the RDEVICE macro instruction to generate a real device block (RDEVBLOK). You must code an RDEVICE macro for each real I/O device in your I/O configuration. The maximum number of real devices that can be included on the real VM/SP HPO system is 4096.

RDEVICE macro instructions describe each device, or group of devices, attached to your processor. These can be in any order, but they must be contiguous, and must precede all RCTLUNIT and RCHANNEL macros in the real I/O configuration file (DMKRIO). Also, RDEVICE macro instructions must follow all groups of CLUSTER and TERMINAL macros, if any. The first RDEVICE macro generates the label DMKRIODV, which indicates the start of real device blocks to CP.

The name field may not be specified for the RDEVICE macro instruction. If a name is specified it is ignored. The RDEVICE macro generates a name by appending the device address to the characters RDV. For example, the name RDV0234 is generated for the device address 234.

If your system includes any of the following devices, turn to the pages shown for information about that device and special instructions on coding the RDEVICE macro for it.

Remote 3270 Display Devices

- Chapter 13, "Planning for 3270 Information Display System Terminals" on page 143
- "Coding the Real I/O Configuration Macros for Remote 3270s" on page 243
- "Coding the RDEVICE Macro for Remote 3270 Information Display Devices" on page 266

3704, 3705, and 3725 Communications Controllers

- Chapter 14, "Planning for the 3704, 3705, and 3725 Communications Controllers" on page 153
- "Coding the RDEVICE Macro for the 3704/3705/3725 Communications Controller" on page 268

3800 Printer

- Chapter 16, "Planning for the 3800 Image Library" on page 163
- "Coding the RDEVICE Macro for the 3800 Printing Subsystem" on page 273

3880 Storage Subsystem

- "3880 Storage Subsystem" on page 96
- "Coding the RDEVICE Macro for the 3880 Storage Subsystem" on page 274

The format of the RDEVICE macro is:

Name	Operation	Operands
	RDEVICE	ADDRESS={ccuu / (ccuu,nnn)} ,DEVTYPE=type[,MODEL=model] [,FEATURE=(feature[,feature]...)] [,CLASS= (cl[,cl]...)] where cl = DASD, TAPE, TERM, GRAF, URI, URO [,ADAPTER={BSCA / IBM1 / SDLC / TELE2 / TYPE1 / TYPE2 / TYPE3 / TYPE4 / TYPE5}] [,CPTYPE={EP}] . [,ALTCU=ccuu] [,SETADDR=sadnum] [,CPNAME=cpname] [,BASEADD=ccuu] [,CLUSTER=label] [,IMAGE=imagelib] [,CHARS=ffff] [,FCB=lpi] [,DPMSIZE=n]

where:

ADDRESS = {ccuu / (ccuu,nnn)}
is the real I/O device address (or addresses).

ccuu
is three or four hexadecimal digits from 000 to 1FFF, without leading blanks. The one or two high-order digits are the address of the channel to which the device is attached. The two low-order digits represent the control unit and device address.

nnn
is the number of RDEVBLOK entries to be generated. It may be any number from 001 to 256. For example, if ADDRESS=(100,5) is specified, RDEVBLOKs with device addresses 100, 101, 102, 103, and 104 are generated. If nnn is omitted, a value of 1 is assumed for all devices except the 2305, which has a default value of 8. For a 2305, the last character of cuu should be 0 or 8 and the maximum value of nnn is 16.

If DEVTYPE=3066, 3138, 3148, or 3158, or if DEVTYPE=3278 and Model=2A, nnn can only be 1. This is because only one system display console can be specified for each RDEVICE macro.

DEVTYPE = type
is the type of device.

The device type can be CTCA, 1017, 1018, 1052, 1053, 1403, 1442P, 1442R, 1443, 2150, 2250, 2260, 2265, 2301, 2303, 2305, 2311, 2314, 2319, 2321, 2401, 2402, 2403, 2404, 2415, 2420, 2495, 2501, 2520P, 2520R, 2540P, 2540R, 2671, 2701, 2702, 2703, 2955, 3036, 3066, 3088, 3138, 3148, 3158, 3203, 3210, 3211, 3215, 3230, 3235, 3262, 3263, 3277, 3278, 3279, 3284, 3286, 3287, 3288, 3289, 3290, 3310, 3330, 3333, 3340, 3350, 3370, 3375, 3380, 3410, 3411, 3420, 3430, 3505, 3525, 3704, 3705, 3725, 3800, 3851, 3880, 4245, 4248, 4250, 7743, FB-512, HFGD, ICA.

Specify the DEVTYPE for the 3880 Model 11 Storage Subsystem as 3350P, and for the 3880 Model 21, as 3351P.

Coding Considerations

Additional information about the support of HFGD (High Function Graphic Device) devices can be found in the *Graphics Access Method/System Product General Information Manual*, GC33-0125.

For TWX terminals, 3101 display terminals or 3232 keyboard terminals, specify 270x as the device type and ADAPTER = TELE2.

Remote terminals such as a 2741 or a 3767 must be coded as a 2701, 2702, 2703, 3704, 3705, or 3725.

For a 3350 device in native mode, specify 3350 as the device type. For a 3350 being used in 3330 compatibility mode, specify 3330.

Specify a 3344 disk as a 3340, and a 3333 as a 3330.

Specify a 3250 device as a 2250.

An MSS 3330V device address must be defined as DEVTYPE = 3330 with one of the two FEATURE = operands allowed. Refer to the explanation of the FEATURE operand below.

For 3287 printers attached via a 3272 Control Unit Model 2, specify DEVTYPE = 3284 or 3286.

For a 3289 Model 4 to be attached to a 4331 Display Printer Adapter as a system printer, specify DEVTYPE = 3289E, or DEVTYPE = 3289, MODEL = 4.

For a logically attached 3290 Information Panel, you may specify either:

1. DEVTYPE = 3290.

   In this case, do not specify the MODEL = operand. For 3278 compatibility mode, VM assumes that this is a 3278-2. Note that for many applications, you should define one logical terminal of the 3290 as a 3278-2 before you log on to the system.

or

2. DEVTYPE=3278.

   In this case, you must specify the MODEL= operand.

To support the 4248-1 printer in 3211 compatibility mode, specify 3211 as the device type.

Specify CTCA for Channel-to-Channel Adapter. Since a CTCA may tie up a channel, you should generate only one per channel. If other devices are to be attached to the same channel as a CTCA, they should be non-critical devices, such as readers or printers.

The system console must be specified in both the RDEVICE and RCTLUNIT macros. Specify the system console in both macros as follows:

System/370 Model	System Console
158 II	3210 or 3215
158	3158 (if in display mode) 3215 (if in printer-keyboard mode and has the 3213 Printer Model 1)
165 II, 168	3066
303x, 3042	3036
308x	3278 Model 2A (if in display mode)
3090	3278 Model 2A
4341, 4341-2	3278 Model 2A (if in display mode) 3215 (if in printer-keyboard mode) 3279 Model 2C
4381-1, 4381-2, 4381-3	3278 Model 2A 3279 Model 2C

Addresses 0F0 through 0FF are reserved for the attachment of the support processor subsystem devices, for the SIGM and SIGP instructions that are used by the instruction processing function and support processor to communicate with each other, and as spare addresses 0F0 to 0FF, which have the following assignments:

- 0F2 - 3278 Model 2A primary console
- 0F3 - 3278 Model 2A additional console or 3287 Printer
- 0F4 - 3278 Model 2A additional console or 3287 Printer
- 0F5 - 3278 Model 2A additional console or 3287 Printer
- 0F6 - SIGM instruction
- 0F7 - SIGP instruction
- 0F0, 0F1, 0F8 through 0FF - spares

Device types 2540R and 2540P refer to the same IBM 2540 Card Read Punch (as do 1442P and 1442R, and 2520P and 2520R). Each logical device must be specified in a separate RDEVICE macro.

In addition, any other device that can be attached to a real processor can be specified in the RDEVICE macro by its device type. For unsupported devices that do not have a device type listed under the DEVTYPE operand, you should code the subclass on the CLASS operand. Then unsupported devices can be dedicated to a virtual machine, and CP can log any error recordings. CP does not use unsupported devices for its own operations.

If a device specified in the RDEVICE macro is not supported by VM/SP HPO, the following MNOTE message (warning level) is generated:

```
UNSUPPORTED DEVICE TYPE
```

The device is generated as an unsupported device. An unsupported device can be used only if it is dedicated to a virtual machine. It is dedicated to a virtual machine if a DEDICATE control statement is coded in the directory entry for the virtual machine, or if it is attached to it by the CP ATTACH command.

Notes:

1. *If you code a 2702 device type the SETADDR value must be specified.*
2. *If you code a 3278 or 3279 device type the MODEL= operand must be specified.*

MODEL=model
is the model letter and number for a particular device. It must be coded for 2305, 3330, DASD, 3278, and 3279 display devices, and for 3203 and 3262 printers. If not specified, the model number defaults to zero except for the 3203, for which it defaults to 4. If a model number is not coded for 3704 or 3705 devices, or the 3262 printer, an MNOTE is generated.

Model is a value that can be:

Value	Device
1 or 2	2305
4 or 5	3203
1, 5, or 11	3262
1, 2, or 11	3330
1 or 11	3333
A1 - H8	3704, 3705-I, or 3705-II
1 or 2	3725
1, 2, 3, 4, 5, or 6	2415
5 or 7	2420
1, 2, or 3	3410 or 3411
3, 4, 5, 6, 7, or 8	3420
1	3272 or 3274, Model 1B

Value	Device
2, 2A, 3, 4, or 5	3278
2, 2C, or 3	3279
4	3289
2	3178 (See Note 4)
1	4248

Notes:

1. *The 3277 Model 1 is a 480-character display screen and is supported by VM/SP HPO only as a dedicated device.*
2. *If a model number is included for devices that do not require model numbers, system generation is terminated with an error message. FB-512 devices do not accept model numbers.*
3. *If DEVTYPE=3278 or 3279, MODEL=* must *be specified.*
4. *MODEL need not be coded for 3178, as it will default to 2.*
5. *If DEVTYPE=3290, MODEL should not be specified. For 3278 compatibility mode, VM assumes that the device is a 3278-2.*

FEATURE=(feature[,feature]...)
are the device's optional features. Features can be written in any order. They are:

Feature	Explanation
7-TRACK	7-track head on a tape drive
CONV	Conversion feature on a 7-track tape drive
DUALDENS	Dual density on a tape drive
OPRDR	Operator identification card reader on a 3277 Model 2, or magnetic slot reader on a 3278 or 3279
SYSVIRT	A 3330V (DEVTYPE=3330) device that may be used by VM/SP HPO for mounting MSS system volumes
TRANS	Translation feature on a 7-track tape drive
UNVCHSET	Universal character set printer
VIRTUAL	A 3330V (DEVTYPE=3330) device that may be dedicated to a virtual machine

Feature	Explanation
2CHANSW	Two-channel switch feature for tape or DASD drive
4CHANSW	Four-channel switch feature for tape or DASD drive
4WCGMS	A 3800 (DEVTYPE=3800) device with four Writeable Character Generation Modules
FH	3350 Fixed-head Feature (3340 optional)

Note: For a 3330V device, either FEATURE=VIRTUAL or FEATURE=SYSVIRT must be specified.

Coding Considerations

To allow CMS to correctly verify tape mode set operations, the correct feature code for a tape device must be specified.

Note: The dual density selected by the DUALDENS feature is dependent on the tape drive and model specified in the DEVTYPE and MODEL operands.

If the local 3277, 3278, or 3279 display device is equipped with the optional operator identification card reader or magnetic reader attachment, then the virtual machine operator can gain access to the system (log on) only if he inserts a magnetically encoded card. Use the FEATURE=OPRDR operand of the RDEVICE macro to specify the display device with a card reader. If you do not want access authorization by a card reader, do not code FEATURE=OPRDR in the RDEVICE macro. FEATURE=OPRDR is invalid if DEVTYPE=3158.

Note: The 7-TRACK, CONV, DUALDENS, and TRANS features are not allowed for the 34340.

It is allowable but not necessary to designate FEATURE=(2CHANSW/4CHANSW) on the RDEVICE macro. DMKCPI dynamically determines if the hardware has a two- or four-channel switch feature by issuing a release CCW to the tape or count-key-data DASD volumes. For FB-512 devices, the RDFEAT bit in the appropriate FB-512 RDCBLOK is checked. If the hardware supports the two- or four-channel switch feature, the FTRRSRL bit is turned on in the RDEVFTR field.
FEATURE=(2CHANSW/4CHANSW) on the RDEVICE macro is ignored since determination is made by the initialization routine. For compatibility reasons, the FEATURE=(2CHANSW/4CHANSW) operand is allowed, but when specified, causes the following MNOTE to be issued:

```
2CHANSW FEATURE IGNORED
4CHANSW
```

CLASS = { (cl[,cl]...) | DASD | TAPE | TERM | GRAF | URI | URO }

is the device class. It is either the output spooling class or a special subclass for unsupported devices.

Note: * *is a valid output class for printers. It indicates that a printer will accept any output class.*

Output Spooling Classes

The spooling classes (cl,cl...) list up to four output spooling classes separated by commas. This form of the CLASS operand can be specified only for a 1403, 1443, 3203, 3211, 3289 Model 4, or 3262 printer, or 2520P, 2540P or 3525 card punch. The spooling class, cl, is one alphameric character. If you specify more than one class, you must separate them by commas. If no class is specified, class A is assumed for printers and punches.

CLASS is used by the CP START command and may be changed by this command. For a complete description of the START command, and for more information about spooling classes, see the *VM/SP HPO Operator's Guide.*

Subclass for Unsupported Devices

Specify a device subclass for unsupported device types only. CP uses the subclass when it translates virtual CCW strings directed to unsupported devices. This form of the CLASS operand is valid only if the device type specified on the DEVTYPE operand does not appear in the list of valid device types.

Subclasses are:

DASD	Direct Access Storage Devices
TAPE	Tape devices
TERM	Terminals
GRAF	Display mode terminals
URI	Unit record input devices
URO	Unit record output devices

You must determine the correct subclass to specify for any device type that does not appear in the list of valid device types under the DEVTYPE operand. Do not code a subclass for any device type that appears in that list. For example, a 1287 Optical Reader is an unsupported device. It does not appear in the list of supported devices and is not listed as a device type for the DEVTYPE operand of the

RDEVICE macro. However, you can define a 1287 and use it if you dedicate it to a virtual machine. You must decide the correct subclass. For example,

```
RDEVICE ADDRESS=010,DEVTYPE=1287,CLASS=URI
```

defines a 1287 Optical Reader at address 010. The 1287 belongs to the unit record input (URI) subclass.

Notes:

1. *If you use this form of the CLASS operand, and the unsupported device does not function properly, try dedicating the device to a virtual=real machine and inhibiting CCW translation (by issuing SET NOTRANS ON). A maximum of 32 sense bytes can be in the RDEVBLOK created for an unsupported device.*

2. *The CLASS operand is invalid if you are specifying service record file devices.*

ADAPTER = { **BSCA** | **IBM1** | **SDLC** | **TELE2** | **TYPE1** | **TYPE2** | **TYPE3** | **TYPE4** | **TYPE5** }

is the terminal control or transmission adapter used to connect a telecommunication I/O device to its control unit. This operand is required if a DEVTYPE of 2701, 2702, 2703, 3704, 3705, 3725, or ICA is specified. It is ignored if specified for any other device type.

BSCA
specifies an IBM Binary Synchronous Terminal Adapter Type II for a 2701, or an IBM Binary Synchronous Terminal Control Type II for a 2703, 3704, 3705, or 3725. BSCA must be specified for remote 3270 terminals and printers.

IBM1
specifies that an IBM Terminal Adapter Type I attaches a 1050 or 2741 to a 2701, or that an IBM Terminal Control Type I attaches a 1050 or 2741 to a 2702 or 2703, or that a Line Interface Base Type I attaches a 1050 or 2741 to a 3704, 3705, or 3725.

SDLC
specifies that a 4331 Communications Adapter operates its teleprocessing lines in Synchronous Data Link Control (SDLC) mode. ADAPTER=SDLC is valid only when you specify DEVTYPE=ICA.

TELE2
specifies that a 3101 display terminal or a CPT-TWX (Models 33/35) Terminal attaches to:

- A Telegraph Terminal Adapter Type II in a 2701
- A Telegraph Terminal Control Type II in a 2702 or 2703
- To a Line Interface Base Type I in a 3704, 3705, or 3725

TYPEn
specifies the channel adapter accessed by a 3704, 3705, or 3275 Communications Controller:

TYPE1	for a 3704
TYPE1 or TYPE4	for a 3705
TYPE5	for a 3725

In identifying the channel adapter, TYPE1, TYPE4, or TYPE5 must be specified for the Emulation Program (EP). In identifying the line adapter, IBM1, TELE2, or BSCA can be specified only in relation to another RDEVICE macro containing ADAPTER=TYPE1 (if DEVTYPE=3704), ADAPTER=TYPE4 (if DEVTYPE=3705), or ADAPTER=TYPE5 (if DEVTYPE=3725).

SETADDR=sadnum
is the set address (SAD) command issued for a telecommunication line attached to a 2702, 3704, or 3705 control unit. This operand is required if the device is a 2702.

Sadnum Value	Command
0	SADZERO
1	SADONE
2	SADTWO
3	SADTHREE
4	(no SAD command is issued)

CPTYPE={EP}
is the 3704/3705/3725 control program to be run in a 3704, 3705, or 3725 Communications Controller. EP specifies the 2701, 2702, or 2703 Emulation Program.

The adapter types valid when CPTYPE=EP are as follows:

TYPE1	Yes
TYPE2	No
TYPE3	No
TYPE4	Yes
TYPE5	Yes

ALTCU=ccuu
specifies an alternate control unit address to be used if paths through the primary control unit are unavailable. ccuu is a three- or four-digit hexadecimal address from 000 to 1FFF. Only one ALTCU can be specified.

The ALTCU ccuu must specify an address with a low order of 0 or 8. Otherwise, the following MNOTE is issued:

```
INVALID ALTCU ADDRESS
```

The ALTCU operand is valid only for tape and DASD volumes. An MNOTE is issued if an invalid device type is specified.

```
"ALTCU" IS INVALID FOR DEVICE TYPE "devtype"
```

The ALTCU operand should be specified when you have the String Switch feature to support two control unit paths to a device.

In an MP system, the alternate control unit address may be the same as the primary control unit address. If there are two physical control units with the same address, they cannot be attached to both processors at the same time. In order to attach a primary control unit and an alternate control unit to both processors of an MP configured system, you must specify unique control unit addresses.

The ALTCU ccuu address should specify the low address associated with the alternate real control unit. When the FEATURE=xxx-Device operand indicates that the control unit supports more than sixteen devices and the devices on the second or following group of sixteen devices are defined by separate RDEVICE macros, the ALTCU ccuu should identify the logical RCUBLOK. CP constructs one RCUBLOK for each set of sixteen devices supported by the real control unit.

Given an alternate control unit configuration where each of two control units support thirty-two devices, the following two macro definitions are acceptable:

```
RDEVICE ADDRESS=(0300,32),ALTCU=1400
RCTLUNIT ADDRESS=300,FEATURE=32-DEVICE
RCTLUNIT ADDRESS=1400,FEATURE=32-DEVICE
RCHANNEL ADDRESS=03
RCHANNEL ADDRESS=14
RDEVICE ADDRESS=(300,16),ALTCU=1400
RDEVICE ADDRESS=(1410,16),ALTCU=310
RCTLUNIT ADDRESS=300,FEATURE=32-DEVICE
RCTLUNIT ADDRESS=1400,FEATURE=32-DEVICE
RCHANNEL ADDRESS=03
RCHANNEL ADDRESS=14
```

CPNAME=cpname
is the one- to eight-character name of a 3704/3705 control program that is to be automatically loaded in the 3704 or 3705 at IPL time. If an automatic load is not desired, omit this operand. Do not code this operand for a 3725 Communications Controller.

Note: If you do not code the CPNAME operand on the RDEVICE macro for the 3704/3705 base address, CP will mark the device "not operational" at IPL time. You may not be able to use the cluster on that 3704/3705.

BASEADD = ccuu
is the native address (load address) of the 3704/3705 that controls the physical line(s). This operand is required for correct operation of VM/SP HPO recovery management for emulation lines based on a 3704/3705. This operand is valid only if ADAPTER = IBM1 (or = TELE2 or = BSCA). It must be specified in order to use the 370x EP Line Trace Facility.

CLUSTER = label
is the label of the CLUSTER macro that defines the clustered or stand-alone remote control unit attached to this line. This operand is valid only if ADAPTER = BSCA is specified.

IMAGE = imagelib
is the image library to be used by the 3800 printer device after a cold start if none is specified on the START command. If this operand is omitted, the default is IMAG3800.

CHARS = ffff
is one to four characters that represent the character arrangement table for the 3800 printer device to be used after a cold start if none is specified on the START command. If this operand is omitted, the default is GF10.

DPMSIZE = n
is the maximum size of the delayed purge queue for the 3800 printer device to be used after a cold start if none is specified on the START command. If this operand is omitted, the default is 1. (The maximum allowed is 9.)

FCB = lpi
is the FCB to be used for the page separator (6, 8, or 12) for the 3800 printer device after a cold start if none is specified on the START command. If this operand is omitted, the default is 6.

Examples

The following examples illustrate the use of the RDEVICE macro instructions to describe a 1403 printer with the Universal Character Set (UCS) feature, four 9-track, 800 bpi tape drives, and eight CPT-TWX lines on a 2702.

```
RDEVICE ADDRESS=00E,DEVTYPE=1403,FEATURE=UNVCHSET,
        CLASS=(A,C)
RDEVICE ADDRESS=(0C0,4),DEVTYPE=2401
RDEVICE ADDRESS=(030,8),DEVTYPE=2702,ADAPTER=TELE2,
        SETADDR=2
```

Unit Record Error Messages

The RDEVICE macro instruction generates an entry in a table of printers or a table of punches or a table of readers for spooling when DEVTYPE = 1403, 3203, 3211, 3262, 3263, 3289 Model 4, 3289E, 3800, 1443, 2540P, 3525, 2540R, 3505, 4245, 4248, or 2501 is specified. Each table has a maximum of 32 entries. One of the following messages results if more than 32 readers, printers, or punches are specified.

```
MORE THAN 32 READERS
MORE THAN 32 PRINTERS
MORE THAN 32 PUNCHES
```

If any of these messages prints, it indicates that the RDEVBLOK is generated, but no entry is made in the printer or punch table. The device cannot be used for CP spooling.

Control Unit Error Messages

The RCTLUNIT macro generates an RCUBLOK containing an index to each of sixteen possible devices. When ALTCU is specified, both the primary and alternate RCUBLOKS contain an index to the same RDEVBLOK. The following MNOTE is issued when an RDEVICE macro specifying the ALTCU operand is defined and an RDEVICE macro is also defined for a device with the alternate control unit address:

```
CONTROL UNIT TABLE for raddr1 IN USE by raddr2
```

For example:

```
RDEVICE ADDRESS=140,ALTCU=150
RDEVICE ADDRESS=150,
RCTLUNIT ADDRESS=140
RCTLUNIT ADDRESS=150
RCHANNEL ADDRESS=1
```

Device 140 is defined with a primary control unit address of 140 and an alternate control unit address of 150. The ALTCU = 150 specification indicates that the 150 RCUBLOK will contain an index to the 140 RDEVBLOK. In this example, a RDEVICE macro also appears for device 150. A conflict arises since the RCUBLOK index for control unit 150 cannot index to both RDEVBLOK 140 and RDEVBLOK 150. In the above example, the user must remove the 150 RDEVICE macro to resolve the conflict.

Coding the RDEVICE Macro for Remote 3270 Information Display Devices

Code one RDEVICE macro for each binary synchronous line used for remote 3270s (one RDEVICE macro for each CLUSTER macro). A maximum of 256 RDEVICE macros for lines to support remote 3270s is allowed.

The format of the RDEVICE macro for the binary synchronous line and transmission control unit for remote 3270s is included here to help you code the macro correctly. The format of an RDEVICE macro for a communication line that supports remote 3270s is:

Name	Operation	Operands
	RDEVICE	ADDRESS=ccuu, DEVTYPE= { 2701 | 2703 | ICA | 3704 | 3705 | 3725 } ADAPTER=BSCA [,BASEADD=ccuu] ,CLUSTER=label

where:

ADDRESS = ccuu
is the real I/O address of the binary synchronous line to be used by remote 3270s.

The address, ccuu, is three or four hexadecimal digits from 000 to 1FFF.

DEVTYPE = { 2701 | 2703 | ICA | 3704 | 3705 | 3725 }
is the device type of the transmission control unit that controls the line.

Note: The lines attached to the 3704/3705/3725 must be controlled by the Emulation Program (EP).

ADAPTER = BSCA
is the terminal adapter for the transmission control unit. BSCA represents the:

- IBM Binary Synchronous Terminal Adapter Type II for a 2701
- IBM Binary Synchronous Terminal Control Type II for a 2703
- IBM 3704, 3705, and 3725 Communications Controllers

BASEADD = ccuu
is the native subchannel address (load address) of the 3704/3705/3725 that controls the physical line or lines. This operand is required for correct operation of VM/SP HPO recovery management for emulator lines on a 3704/3705/3725. Specify this operand only for 3704/3705/3725 Communications Controllers.

CLUSTER = label
is the label of a CLUSTER macro that defines the cluster or stand-alone station attached to this line.

Examples

The following examples are RDEVICE macros describing a nonswitched point-to-point communication line connected to a 2701, 2703, and 3705.

Example 1:

```
RDEVICE ADDRESS=078,DEVTYPE=2701,ADAPTER=BSCA,                X
        CLUSTER=CLUST001
```

The cluster station that is connected to this line is defined by the CLUSTER macro labeled CLUST001. The line at address 078 is controlled by a 2701 transmission control unit.

Example 2:

```
RDEVICE ADDRESS=080,DEVTYPE=2703,ADAPTER=BSCA,                X
        CLUSTER=CLUST020
```

The line at address 080 is controlled by a 2703 transmission control unit and the corresponding CLUSTER macro is labeled CLUST020.

Example 3:

```
RDEVICE ADDRESS=0B8,DEVTYPE=3705,ADAPTER=BSCA,                X
        BASEADD=0B0,CLUSTER=CLUST030
```

The line at address 0B8 is controlled by a 3705 Communications Controller and the corresponding CLUSTER macro is labeled CLUST030.

Coding the RDEVICE Macro for the 3704/3705/3725 Communications Controller

The 3704/3705/3725 Communications Controllers have varied uses. Consequently, the control program generation for the 3704/3705/3725 is complex.

When using multiple 3705s, you must generate each unit independently so that each has the controller attribute. Otherwise, VM/SP HPO considers the 3705s to be 270x lines and, therefore, unusable to guest virtual machines.

EP-Only Control Programs

If the 3704/3705/3725 is to be run in emulation mode:

- Use the (ccuu,nnn) form of the ADDRESS operand to generate multiple RDEVBLOKs.
- Specify the appropriate name for CPNAME. CPNAME is not valid for a 3725 Communications Controller.

To generate additional emulator lines for the same 3704/3705/3725, use the following coding guidelines on subsequent RDEVICE macros:

- Omit the CPTYPE, CPNAME, and MODEL operands.
- Specify the ADAPTER as IBM1, TELE2, or BSCA, as appropriate.

For ADAPTER=IBM1 (or TELE2), the SETADDR operand must also be specified, exactly as if the device were a 2702 or 2703.

Note: If you use the (ccuu,nnn) form of the ADDRESS operand to generate multiple RDEVBLOKs and specify the CPNAME and ADAPTER=TYPE1 operands on the RDEVICE macro, the additional RDEVBLOKs are generated as ADAPTER=IBM1 and SETADDR=4.

The format of the RDEVICE macro for a 3704/3705/3725 is included here to help you code the macro correctly. The format of the RDEVICE macro for any IBM 3704/3705/3725 is:

Name	Operation	Operands
	RDEVICE	ADDRESS={ccuu | (ccuu,nnn)}, DEVTYPE={3704 | 3705 | 3725}, ADAPTER={TYPE1 | TYPE2 | TYPE3 | TYPE4 | TYPE5 | IBM1 | TELE2 | BSCA} [,MODEL=ab] [,SETADDR={0 | 1 | 2 | 3 | 4}] [,CPTYPE=EP] [,CPNAME=ncpname] [,BASEADD=ccuu]

where:

ADDRESS = {ccuu | (ccuu,nnn)}
is the real device address (ccuu) of the 3704/3705/3725. Use the (ccuu,nnn) form to generate multiple (nnn) real device blocks (RDEVBLOKs) when CPTYPE = EP is specified.

DEVTYPE = {3704 | 3705 | 3725}
is the device type. You should specify 3704, 3705, or 3725 instead of 2701, 2702, or 2703 when CPTYPE = EP.

ADAPTER = {TYPE1 | TYPE2 | TYPE3 | TYPE4 | TYPE5 | IBM1 | TELE2 | BSCA}
identifies either the channel adapter accessed by the specified real address (TYPE1, TYPE2, TYPE3, TYPE4, or TYPE5), or a line adapter if this is an emulator line group (IBM1, TELE2, or BSCA). Only TYPE1 is valid for a device type of 3704. Only TYPE5 is valid for a device type of 3725. For DEVTYPE = 3705, TYPE1 or TYPE4 may be

coded. In identifying the line adapter, IBM1, TELE2, or BSCA can be specified only in relation to another RDEVICE macro that has ADAPTER=TYPE1, TYPE2, TYPE3, or TYPE4.

MODEL=ab
is the 3704/3705 model letter and number, respectively, or the 3725 model number.

The model number determines the size of the 3704/3705 storage. See Figure 35 for a list of the model numbers and the corresponding storage sizes. You should enter both a letter and a number for the 3704 or 3705. If you enter only a single numeric character, an MNOTE is issued and the number is treated as model data for a 3704 or 3705-I, depending on the value of DEVTYPE.

For the 3275, the model number is 1 or 2.

SETADDR= $\left\{\begin{matrix}0\\1\\2\\3\\4\end{matrix}\right\}$
indicates which set address (SAD) command must be issued prior to enabling the emulator lines.

CPTYPE=EP
indicates which type of 3704/3705/3725 control program is run in this 3704/3705/3725.

The adapter types valid when CPTYPE=EP are as follows:

TYPE1	Yes
TYPE2	No
TYPE3	No
TYPE4	Yes
TYPE5	Yes

CPNAME=ncpname
is the one- to eight-character name of the control program to be automatically loaded in the 3704/3705 at system IPL time. If an automatic load is not desired, omit this operand. CPNAME is not valid for a 3725.

IBM 3704 Communications Controller		IBM 3705 Communications Controller		
Model	Storage		Model	Storage
A1	16K	3705-I	A1, B1, C1, D1	16K
A2	32K		A2, B2, C2, D2	48K
A3	48K		B3, C3, D3	80K
A4	64K		B4, C4, D4	112K
			C5, D5	144K
			C6, D6	176K
			D7	208K
			D8	240K
		3705-II	E1, F1, G1, H1	32K
			E2, F2, G2, H2	64K
			E3, F3, G3, H3	96K
			E4, F4, G4, H4	128K
			E5, F5, G5, H5	160K
			E6, F6, G6, H6	192K
			E7, F7, G7, H7	224K
			E8, F8, G8, H8	256K

Figure 35. IBM 3704/3705 Models

Note: Failure to code the CPNAME operand on the RDEVICE macro for the 3704/3705 base address causes CP to mark the device "not operational" at IPL time. The cluster on that 3704/3705 is therefore unusable.

BASEADD=ccuu
is the native address (load address) of the 3704/3705/3725 that controls the physical line(s). This operand is required for correct operation of VM/SP HPO recovery management for emulator lines based on a 3704/3705/3725. It must be specified in order to use the 370x EP line trace.

This operand is valid only if ADAPTER=IBM1, TELE2, or BSCA.

Summary of RDEVICE Macro Coding Considerations for the 3704/3705/3725

For each physical 3704/3705/3725, there should be only one RDEVICE macro which specifies the ADAPTER=TYPEn, MODEL, CPTYPE, and CPNAME operands. This RDEVICE macro defines the base address of the 3704/3705/3725 (that is, the real address used to perform the load and dump operations). If the physical device is a 3705 with two channel adapters installed, there may be a second RDEVICE macro that specifies the ADAPTER=TYPEn, MODEL, and CPTYPE operands. There must *never* be a second use of the CPNAME operand. Even if CPTYPE=EP is specified, the 3704/3705/3725 base address cannot be used as a telecommunication line. Its function is only to load and dump the 3704/3705/3725. The device type and class are different from those of all other lines generated.

Whenever there is more than one subchannel address (CPTYPE=EP), include in the DMKRIO file all RCTLUNIT macros required to specify real addresses which the EP control program may use.

If you have a 3704/3705/3725. and a 2701/2702/2703 on the same VM/SP HPO system, the virtual addresses for the 3704/3705/3725 must not be the same as any of the real 2701/2702/2703 addresses.

Examples

Examples of RDEVICE macro specifications follow. For convenience, the continuation character in position 72 is not shown.

Example 1:

```
RDEVICE ADDRESS=(020,16),
        DEVTYPE=3704,
        MODEL=A2,
        ADAPTER=TYPE1,
        CPNAME=VMEP01
```

This describes a 32K 3704 at address X'020', with 15 emulator lines addresses X'021' to X'02F' and with the default parameter of ADAPTER=IBM1 and SETADDR=4. The 3704 is to be loaded with the Emulation Program 'VMEP01'.

Example 1a:

```
RDEVICE ADDRESS=(030,16),
        DEVTYPE=3704,
        ADAPTER=IBM1,
        SETADDR=2,
        BASEADD=020
```

This describes an additional 16 emulator lines on the same 3704 specified by Example 1.

3704/3705/3725 Error Messages

The RDEVICE macro instruction generates an entry in a table of programmable communications controllers when DEVTYPE=3704, 3705, or 3725 is specified. This table has a maximum of 256 entries. The following message results if more than 256 3704, 3705, or 3725 devices are specified:

```
MORE THAN 256 TP CONCENTRATORS
```

This message indicates that the RDEVBLOK is generated, but no entry is made in the Programmable Communications Controller table.

Coding the RDEVICE Macro for the 3800 Printing Subsystem

The format of the RDEVICE macro for a 3800 is included here to help you code the macro correctly. This format is:

Name	Operation	Operands
	RDEVICE	ADDRESS=ccuu, DEVTYPE=3800, [FEATURE=4WCGMS,] [IMAGE=imagelib,] [CHARS=ffff,] [FCB=lpi,] [DPMSIZE=n,] [MODEL=n,]

where:

ADDRESS = ccuu
is the real device address (ccuu) of the 3800.

DEVTYPE = 3800
is the device type.

FEATURE = 4WCGMS
is a 3800 device with four-writeable character generation modules.

IMAGE = imagelib
is the image library to be used by the 3800 printer device after a cold start if none is specified on the START command.

CHARS = ffff
is the character arrangement table for the 3800 printer device to be used after a cold start if none is specified on the START command.

FCB = lpi
is the FCB to be used for the page separator for the 3800 printer device after a cold start if none is specified on the START command.

DPMSIZE = n
is the maximum size of the delayed queue for the 3800 printer device.

MODEL = n
is the model number of the 3800 printer device: 1 or 3.

Coding the RDEVICE Macro for the 3880 Storage Subsystem

Code one RDEVICE macro for each string of DASD in your 3880 Storage Subsystem.

The format of the RDEVICE macro for a 3880 Storage Subsytem is included here to help you code the macro correctly.

This format is:

Name	Operation	Operands
	RDEVICE	ADDRESS=(ccuu,nn), DEVTYPE={3350P 3351P 3380 } [,ALTCU=ccuu]

where:

ADDRESS=(ccuu,nn)
is the real I/O address of the first storage device attached to the paging storage director.

ccuu
is three or four hexadecimal digits from 000 to 1FFF. The one or two high-order digits are the address of the channel to which the storage director controlling this string of DASDs is attached. The two low-order digits represent the control unit and device address.

The digit indicating the control unit in the address of storage devices attached to a 3880 Model 11 Storage Subsystem must be an even number. The digit indicating the device address must be between 0 and 7. The last digit of the address of storage devices attached to a 3880 Model 21 Storage Subsystem must be between 0 and 3.

If you specify an address whose last digit is not in the range 0-7 for the Model 11, or whose last digit is not in the range 0-3 for the Model 21, you will receive this error message:

```
MNOTE: 'INVALID DEVICE BASE ADDRESS SPECIFIED'
```

nn
is the number of storage devices in the string of DASD: one to eight for the Model 11, one to four for the Model 21, one to sixteen for the Model 13 and Model 23. Since the Model 21, 13, and 23 can attach two strings of DASD, the Model 21 can attach a total of eight storage devices; and the Model 13 and Model 23 can attach a total of thirty-two storage devices.

If you try to attach more than four storage devices to a Model 21 storage director, you will receive this error message:

```
MNOTE: 'MORE THAN 4 DEVICES SPECIFIED'
```

DEVTYPE = {3350P / 3351P / 3380}

is the device type: 3350P for the Model 11, 3351P for the Model 21, 3380 for the Model 13 or Model 23.

[ALTCU = ccuu]

creates an alternate channel path through the second storage director to the storage devices attached to a 3880 Model 21, 13 , or 23 Storage Subsystem. When ALTCU = ccuu is coded, each storage director can communicate with the storage devices attached to the other storage director, as well as with its own storage devices.

When you code RDEVICE macros for the Model 21, Model 13, or Model 23, you must include the ALTCU = ccuu operand if you are generating the second storage director. Otherwise, the ALTCU = ccuu operand is not required; but IBM strongly recommends coding it in the RDEVICE macro for the Model 21, 13, and 23. It is invalid for the Model 11.

ccuu

is three or four hexadecimal digits from 000 to 1FFF. The one or two high-order digits are the address of the channel providing the alternate path. The two low-order digits represent the control unit and device address of the first device in the string.

Since the DASDs must have the same address from the point of view of both storage directors, the device portion of the ALTCU operand must be the same as the device porion of the ADDR operand.

Examples of the RDEVICE Macro for the 3880 Storage Subsystem

Model 11:

```
RDEVICE ADDRESS=(240,8),DEVTYPE=3350P
```

This RDEVICE macro will generate a real device block for a 3880 Model 11 Storage Subsystem with eight devices attached at the following addresses:

Base Address	Paging Addresses
240	248, 250, 258
241	249, 251, 259
242	24A, 252, 25A
243	24B, 253, 25B
244	24C, 254, 25C
245	24D, 255, 25D
246	24E, 256, 25E
247	24F, 257, 25F

Model 21:

```
RDEVICE ADDRESS=(240,4),DEVTYPE=3351P,ALTCU=340
RDEVICE ADDRESS=(350,4),DEVTYPE=3351P,ALTCU=250
```

This pair of RDEVICE macros will generate real device blocks for a 3880 Model 21 Storage Subsystem with eight devices attached at the following addresses:

Base Address	Paging Addresses
240	244, 248, 24C
241	245, 249, 24D
242	246, 24A, 24E
243	247, 24B, 24F
350	354, 358, 35C
351	355, 359, 35D
352	356, 35A, 35E
353	357, 35B, 35F

The primary path to the first group of storage devices is through Channel 2, the alternate path through Channel 3.

The primary path to the second group of storage devices is through Channel 3, the alternate path through Channel 2.

Model 13 and Model 23:

```
RDEVICE ADDRESS=(2A0,16),DEVTYPE=3380,ALTCU=3A0
RDEVICE ADDRESS=(3B0,16),DEVTYPE=3380,ALTCU=2B0
```

This pair of RDEVICE macros will generate real device blocks for a 3880 Model 13 or 23 Storage Subsystem with one string of sixteen devices attached at 2A0 and a second string attached at 3B0.

The primary path to the first group of storage devices is through Channel 2, the alternate path through Channel 3.

The primary path to the second group of storage devices is through Channel 3, the alternate path through Channel 2.

RCTLUNIT Macro

Use the RCTLUNIT macro to generate a real control unit block (RCUBLOK). One RCTLUNIT macro must be specified for each real control unit. The maximum number of real control units is 511, provided that you have enough real storage to hold the real control unit blocks (RCUBLOKs). Control units generally fall into two categories: those supporting eight or fewer devices, and those supporting more than eight devices.

A control unit that supports eight or fewer devices must be assigned an address that is a hexadecimal multiple of eight. That is, its last digit must be 8 or 0. All devices with an address equal to the control unit's address (the base address) or any of the next seven sequential addresses are mapped to this control unit. For example, devices with addresses of 018 through 01F are mapped to a control unit with address 018.

On a multiplexer channel, several device addresses may fall within the address range of one RCTLUNIT macro. When this occurs, only one RCTLUNIT macro may be coded, even though more than one real control unit is present. This case is an exception to the general rule that one RCTLUNIT macro must be specified for each real control unit. For example, a system console at address 009, a 2540 reader at address 00C and a 2540 punch at address 00D would be defined in a single RCTLUNIT macro with a control unit address of 008, even though the card reader and punch have different real control units. In this case, any valid control unit type can be coded. The only exception to this is that control units that operate on a shared subchannel must be specified by separate RCTLUNIT macros.

For control units supporting a range of more than eight device addresses, use the FEATURE operand. The base address must be divisible by sixteen. All devices from the base address up to the number of devices specified by the FEATURE= operand are mapped to the specified control unit. When a control unit supports more than eight devices, the RCTLUNIT macro must specify FEATURE=xxx-DEVICE, where xxx is the number of addressable devices that can be attached to this control unit. The number of devices specified must be divisible by sixteen and rounded to the next higher increment of sixteen if not divisible. The maximum number of devices that can be attached to a control unit is 256.

For example, if you have a 3830 control unit with the 64-device feature installed, you must specify FEATURE=64-DEVICE for it, even if fewer than sixty-four 3330s are installed.

CP requires that all devices on one physical control unit be specified on a single RCTLUNIT macro. The microcode in the 3830-2 that supports 3350 DASD allows address skipping (in blocks of eight addresses) on the same physical control unit. For example, this address scheme

Device Addresses 150-157 and 160-167 on first 3830-2
Device Addresses 158-15F and 168-16F on second 3830-2

is *not supported* by CP. All addresses on a given physical control unit must be specified with a single RCTLUNIT macro using the FEATURE=xxx-DEVICE operand, where appropriate, for a contiguous range of addresses.

A device that attaches directly to the channel without a separate control unit must still have a RCTLUNIT macro coded for it. For example, if a 3210 is defined with an RDEVICE macro, it must have a corresponding RCTLUNIT macro.

All models of the 3880 Storage Subsystem contain two separately addressable storage directors. You must therefore code two RCTLUNIT macros, one for each storage director.

The optional ALTCH operand of the RCTLUNIT macro allows specification of up to three alternate channel interfaces to a single director module. CP supports a maximum of four channel paths to a single director module.

RCTLUNIT macro instructions describing the control units for your installation's system may be in any order, but they must be contiguous and follow all of the RDEVICE macro instructions in module DMKRIO. The first RCTLUNIT macro instruction also generates the label DMKRIOCU, which indicates the start of the real control unit blocks to CP.

The name field need not be specified for the RCTLUNIT macro instruction. If a name is specified it is ignored. The macro generates a name by appending the control unit address to the characters RCU. For example, if the control unit address is 230, RCU0230 is generated.

The format of the RCTLUNIT macro is:

Name	Operation	Operands
	RCTLUNIT	ADDRESS=ccuu ,CUTYPE=type [,ALTCH=(nn,nn,nn)][,FEATURE=xxx-DEVICE] [,UCW= {SHR}] {UNS}

where:

ADDRESS=ccuu
is the real address of the control unit. ccuu consists of three or four hexadecimal digits, without leading blanks. The one or two high-order digits are the channel address of this control unit, and may be any hexadecimal number from 0 to 1F. The two low-order digits must be the lowest address of the control unit. If the xxx-DEVICE feature is supported, the low-order digit must be 0. Otherwise, it must be either 0 or 8.

Note: If your installation has a 2701, 2702, or 2703, and a 3704, 3705, or 3725 executing in emulation mode, you must be sure that their addresses are not the same.

CUTYPE = type
is the device type of the control unit. One of the following device type numbers can be specified: 1052, 1442, 2150, 2250, 2314, 2319, 2403, 2404, 2415, 2495, 2501, 2520, 2701, 2702, 2703, 2803, 2804, 2820, 2821, 2822, 2826, 2835, 2840, 2841, 2844, 2845, 2848, 2955, 3036, 3066, 3088, 3138, 3148, 3158, 3210, 3215, 3272, 3274, 3411, 3505, 3525, 3704, 3705, 3725, 3803, 3811, 3830, 3880, 3851, 4245, 4248, 7443, CTCA (Channel-to-Channel Adapter), FTA (File Tape Adapter), HFCU (High Function Control Unit), ICA (Integrated Channel Adapter), IFA (Integrated File Adapter), ISC (Integrated Storage Control), or SVPC (Service Processor).

Only one CTCA may be defined at a time.

Any other control unit that can be attached to a real processor may be specified in a RCTLUNIT macro instruction by its device type.

Notes:

1. *Specify an Integrated Printer Adapter (IPA) as a 2821.*

2. *Specify a 3274 Model 1B as a 3272.*

3. *If CUTYPE = CTCA, the low order digit of ADDRESS = must be 0.*

4. *If you are using a 3289 Model 4 Printer as a system printer (DEVTYPE = 3289E) attached to a 4331 Display Printer Adapter, specify CUTYPE = SVPC in the corresponding RCTLUNIT macro. Be careful not to specify a control unit type of 3272 or 3274. This will result in locking out other devices on the adapter, such as 3278s or a second printer, while a printer is operating.*

Even though some devices attach directly to the channel without a separate control unit, a RCTLUNIT macro instruction must be included for them. For example, if you want to define a 3215, you must code an RDEVICE and RCTLUNIT macro for the 3215. Even though the 3215 does not require a control unit, it requires a RCTLUNIT macro. If several devices have addresses that are within the same control unit address, only one RCTLUNIT macro can be specified. Which control unit you specify is not relevant.

ALTCH = (nn,nn,nn)
specifies the alternate channel(s) to be used with the control unit address if the primary channel path is unavailable or offline. nn represents the one- or two-digit channel address the alternate channel paths, and may be any number from 0 to 1F. You can specify up to three alternate channels for AP or UP systems, but only one alternate channel for multiprocessor systems. Specification of more than one alternate channel path for MP generated systems produces the following MNOTE:

```
INVALID ALTCH SPECIFICATION
```

There can be no splitting of control units when using alternate channels. All devices on one physical control unit must be defined as having alternate channel(s) or no alternate channel(s).

FEATURE = xxx-DEVICE
is the optional control unit feature. The feature, xxx-DEVICE, indicates that the control unit is controlling more than eight devices. The prefix, xxx, can be 16, 32, 48, 64, 80, 96, 112, 128, 144, 160, 176, 192, 208, 224, 240, or 256. Appendix B, "Configuration Aid" lists the maximum number of devices that may be specified for each control unit. This feature may be specified for a 2403, 2702, 2703, 2803, 2835, 3272, 3345, 3704, 3705, 3725, 3803, 3830, 3880, FTA (File Tape Adapter), HFCU (High Function Control Unit), ICA (Integrated Channel Adapter), IFA (Integrated File Adapter), or ISC (Integrated Storage Control).

The Integrated File Adapter's 9821 feature, when used with 3344's, is too large for the RCTLUNIT macro to handle. The range of addresses 160-1F7 is invalid.

A DASD controller, such as a 3830, can be ordered with a hardware feature to support 16, 32, or 64 devices. The RCTLUNIT macro *must* specify FEATURE = XXX-DEVICE to match the hardware feature of the controller, regardless of the actual number of devices attached to it. The ADDRESS operand is also very critical. For 32 devices, ADDRESS must be c00, c20, c40, etc. (where c is the channel number): always an even multiple of X'20' (decimal 32). Likewise, for 64 devices, the only correct addresses are c00, c40, c80, and cC0. This information is important when an upgrade of installed DASD controllers occurs. Be sure to change the ADDRESS value as well as the FEATURE specification.

For any unsupported control unit, FEATURE = 16-DEVICE is valid and is the maximum you can specify. Unsupported control units are any that do not appear in Chapter 2, "Configurations."

Note: To use a 3880 Model 11 or Model 21 Storage Subsystem, or a 3880 Model 13 or Model 23 Storage Subsystem with two strings of DASD, you must include the Feature = 32-DEVICE operand in the RCTLUNIT macro. To use a 3880 Model 13 or Model 23 Storage Subsystem with one string of DASD, you must include the Feature = 16-DEVICE operand in the RCTLUNIT macro. This restriction does not apply to other 3880s.

Warning: The starter system does not provide support for configurations over 16 devices when you are defining those needed to do the system generation. Therefore, if your installation includes control units that share more than 16 devices switchable to another processor, the channel interface enable switch on the other processor should be in the disable position while you perform the system generation.

UCW = {SHR / UNS}

is the attribute of the UCW. This can be specified only for 3272 or 3274 control units.

SHR indicates a shared UCW. This is the default.

UNS indicates an unshared UCW.

Examples

The following examples illustrate the use of the RCTLUNIT macro instruction to describe the control units for: a 3215 console printer-keyboard with address 01F, a 3880, and a 3705 with lines 040 through 04B.

```
RCTLUNIT ADDRESS=018,CUTYPE=3215
RCTLUNIT ADDRESS=230,CUTYPE=3880,FEATURE=32-DEVICE
RCTLUNIT ADDRESS=040,CUTYPE=3705,FEATURE=16-DEVICE
```

Channel Error Message

The RCHBLOK contains an index to each of thirty-two possible RCUBLOKS. When the ALTCH operand is specified on the RCTLUNIT macro, both the primary and alternate RCHBLOKS contain an index to the same RCUBLOK. The following message is issued when an RCTLUNIT macro is defined, specifying the ATLCH operand, and an RCTLUNIT macro is defined, which creates an RCUBLOK for the alternate channel address.

```
CHANNEL TABLE FOR RCUxxx IN USE BY RCUyyy
```

For example:

```
RDEVICE ADDRESS=250
RDEVICE ADDRESS=350
RCTLUNIT ADDRESS=250,ALTCH=(3)
RCTLUNIT ADDRESS=350
RCHANNEL ADDRESS=2
RCHANNEL ADDRESS=3
```

The ALTCH specification indicates that the RCHBLOK for channel three should index to the 250 RCUBLOK. The RCTLUNIT macro for 350 causes a conflict since the RCHBLOK cannot index to both the 250 and 350 RCUBLOKS. In the above configuration, the user must remove the RCTLUNIT macro and RDEVICE macro for 350.

RCHANNEL Macro

Use the RCHANNEL macro to generate a real channel block (RCHBLOK). An RCHANNEL macro instruction must be coded to define each real channel in the I/O configuration.

The standard maximum number of channels that can be attached to a processor is 16 for a uniprocessor and 24 for a dyadic processor. The extended channel support option for the 3090 Processor Complex increases these limits to 48 channels for a dyadic processor.

The RCHANNEL macro instructions describing channels for your installation may be in any order, but they must be contiguous and follow all of the RCTLUNIT macro instructions in module DMKRIO. The first RCHANNEL macro instruction generates the label DMKRIOCH, which indicates the start of the real channel blocks to CP.

No name need be specified for the RCHANNEL macro instruction. If a name is specified, it is ignored. The RCHANNEL macro generates a name by appending the channel address to the characters RCHAN. For example, if the channel address is 2, the name RCHAN02 is generated.

The format of the RCHANNEL macro is:

Name	Operation	Operands
	RCHANNEL	ADDRESS=cc ,CHTYPE={SELECTOR MULTIPLEXER BLKMPXR FTA}

where:

ADDRESS = cc
is the real address of the channel. It is a hexadecimal number from 0 to 1F.

CHTYPE = {SELECTOR | MULTIPLEXER | BLKMPXR | FTA}
is the type of channel.

SELECTOR
indicates a selector channel.

MULTIPLEXER
indicates a byte-multiplexer channel.

BLKMPXR
indicates a block multiplexer channel.

FTA
indicates a file tape adapter on a 43XX.

Examples

The following examples illustrate the use of the RCHANNEL macro instruction to describe a multiplexer channel whose address is 0, a selector channel whose address is 1, and a block multiplexer channel whose address is 2.

```
RCHANNEL   ADDRESS=0,CHTYPE=MULTIPLEXER
RCHANNEL   ADDRESS=1,CHTYPE=SELECTOR
RCHANNEL   ADDRESS=2,CHTYPE=BLKMPXR
```

If any errors are detected, the real channel block is not generated. This results in undefined symbols in the real control unit blocks for this channel.

RIOGEN Macro

Use the RIOGEN macro instruction to generate the channel index table and unit record and console tables. RIOGEN must appear as the last macro instruction before the END statement in the DMKRIO file.

The name field must not be specified for the RIOGEN macro.

The format of the RIOGEN macro is:

Name	Operation	Operands
	RIOGEN	CONS=ccuu [,ALTCONS=(ccuu[,ccuu,ccuu...])] [,SRF=(ccuu[,ccuu,ccuu...])]

where:

CONS = ccuu
is the address of the primary system console. The address is a hexadecimal device address that was previously specified in an RDEVICE macro entry. This device must be either a 3036, 3066, 3210, 3215, 7412, 3277, 3278, or 3279 (local attachment), or a 3278 Model 2A, 1052 (via a 2150 freestanding console), a system console for the 3158 (in printer-keyboard mode with the 3213 Printer Model 1 required), or a System Console for the 3138 or 3148 (in printer keyboard mode with a 3286 printer required, or in display mode).

[,ALTCONS = (ccuu = [,ccuu,ccuu...])]
is the address or a list of addresses of alternate consoles. These addresses are hexadecimal device addresses that were previously specified in an RDEVICE macro instruction. There is no limit on the number of alternate consoles that may be specified. These devices, which should be physically located as close as possible to the primary system console, may be any device supported as a VM/SP HPO logon device (except for those remote terminals connected via 3704/3705 Communications Controllers). If the primary system console is not operational at system initialization, an attempt is made to access the first alternate console. If the first alternate console is not operational, an attempt is made to start the next alternate console. If an operational console is found, the console will be used as the system operator's console. If no operational alternate console is found (or if no alternate console was specified), CP enters a disabled wait state with a wait state code of X'005' in the instruction address register (IAR).

Coding Considerations

The alternate console must not be a telecommunications line on a real IBM 3704/3705/3725 Communications Controller unless the 3704/3705/3725 was previously loaded by some other operating system with a 270X Emulator Program.

If the alternate console is an IBM 2741 Communication Terminal, or 3767 Communication Terminal (operating as a 2741), it must use the EBCDIC transmission code. If the alternate console is a local 3277, it must be a Model 2.

[,SRF = (ccuu[,ccuu,ccuu...])]
is the address or a list of addresses of SRF (service record file) devices used for the 303x processors. ccuu is the hexadecimal device address that was previously specified in an RDEVICE macro statement. The device type of the SRF is 7443.

In a 3033AP or 3033MP system, there are two 3036 consoles. Each of these consoles has two SRF devices; therefore, you should specify multiple SRF devices at system generation. The SRF addresses you specify in the RIOGEN macro statement should be the same as the addresses of the SRF devices attached to the service support consoles (see note 3). The RIOGEN macro statement produces an MNOTE warning message if you specify more than 32 SRF devices.

Notes:

1. *In 3033AP or 3033MP environments with I/O configured asymmetrically to one processor, to access the SRF devices in both 3036 consoles, a channel path must be available from the I/O processor to both SRF devices.*

2. *If an SRF device is found to be inaccessible during initialization of the error recording cylinders, an error message is sent to the system operator. Processing continues, but the frames from that SRF device are not placed on the error recording cylinders.*

3. *Only one of the two SRF devices of a 3036 console is accessible at any given time by CP. Therefore, if both SRF devices of a 3036 are specified on the RIOGEN macro, message DMKIOH559W will be issued for* one *of these SRF devices during initialization of the error recording cylinders. Since both SRF devices of a 3036 console contain identical frame data, only one SRF per 3036 needs to be successfully accessed during error recording initialization.*

Examples

The following examples define a primary system console (01F) with an alternate console (050), a system console (009) with no alternate console, and a system console (01F) with two alternate consoles (050 and 060):

```
RIOGEN CONS=01F,ALTCONS=050
RIOGEN CONS=009
RIOGEN CONS=01F,ALTCONS=(050,060)
```

Example of Coding the Real I/O Configuration File (DMKRIO)

In this example, macro instructions are coded to support the following real devices:

1 2540 Card Reader/Punch

2 1403 Printers with the Universal Character Set feature

1 3211 Printer with the Universal Character Set feature

2 3215 Console Printer-Keyboards

1 2955 Data Adapter Unit

1 3278 Console, Model 2A

8 3279 Color Display Stations

1 3705 Communications Controller (with an IBM1, TELE2 and BSCA adapter)

1 2305 Fixed Head Storage with 8 addresses

2 3330 Disk Storage devices (One unit has eight modules and the other has ten. The unit with ten modules has eight of them switchable between two channels.)

1 3350 Direct Access Storage with 8 addresses (string switched)

1 3380 Direct Access Storage with 16 addresses

1 3420 Magnetic Tape Unit, Model 8

2 3420 Magnetic Tape Units, Model 7

1 Multiplexer channel

1 Selector channel

3 Block multiplexer channels

1 Channel-to-Channel Adapter

2 Channel interfaces on the 3851 MSC

96 3330V Direct Access Storage devices, 48 of which can be dedicated to one or more virtual machines and 48 of which are to be used for system volumes

4 3330-1 device addresses that are not real spindles, but rather allow the processor to have direct access to the MSC tables through the 3830-3 Staging Adapter

1 4245 Printer

1 4248 Printer

Figure 36 shows the real configuration.

Figure 37 shows the real I/O configuration file that supports this example.

Considerations for Coding the Input/Output Configuration Source File

If you are generating VM/SP HPO on a 308x or 3090 Processor complex, you must define additional I/O information to the processor controller. To define the I/O configuration to the 308x or 3090 processor, create an input/output configuration source file and write it to the processor controller using the Input/Output Configuration Program. The IOCP source file you create should contain entries that define all real I/O devices.

When preparing the IOCP source file, be sure you code the IOCP macro definitions in the following order:

- Channel paths
- Control units
- I/O devices

It is not necessary to group all macro definitions of the same type together; but you must insure that devices and control units are defined under the appropriate channel macros. If not, IOCP terminates with an error message.

Do not intermix IOCP macro instructions and VM/SP HPO I/O macro instructions in the same file. You must maintain two distinct files, the real I/O configuration file and the corresponding IOCP source file. See the *Input/Output Configuration Program User's Guide and Reference* for IOCP source file coding conventions.

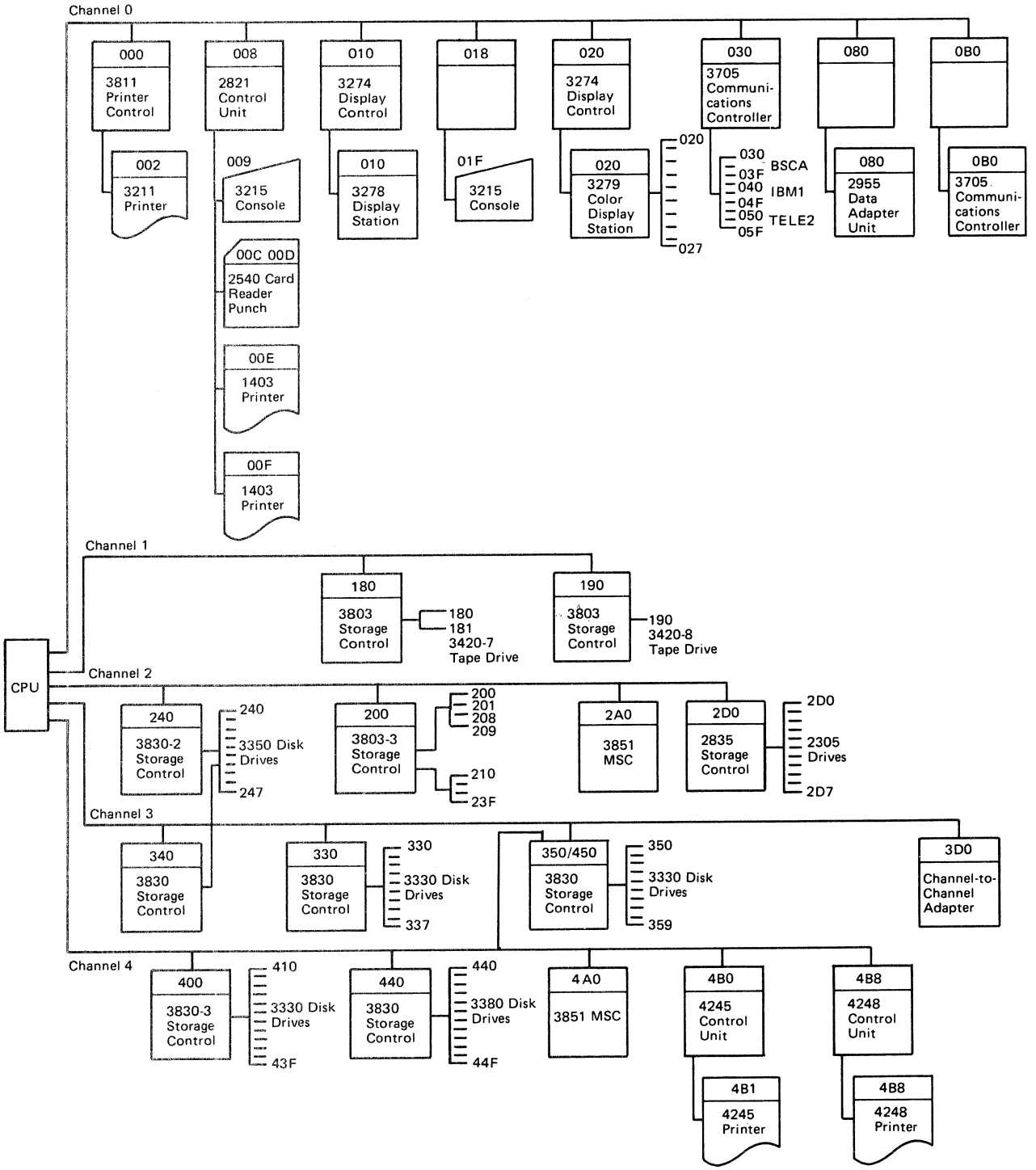


Figure 36. Example of a Real Configuration

```
DMKRIO CSECT
       COPY OPTIONS
       RDEVICE ADDRESS=002,DEVTYPE=3211,CLASS=(X,A),FEATURE=UNVCHSET
       RDEVICE ADDRESS=009,DEVTYPE=3215
       RDEVICE ADDRESS=00C,DEVTYPE=2540R
       RDEVICE ADDRESS=00D,DEVTYPE=2540P,CLASS=(X,A)
       RDEVICE ADDRESS=00E,DEVTYPE=1403,CLASS=(X,A),FEATURE=UNVCHSET
       RDEVICE ADDRESS=00F,DEVTYPE=1403,CLASS=(S),FEATURE=UNVCHSET
       RDEVICE ADDRESS=010,DEVTYPE=3278,MODEL=2A
       RDEVICE ADDRESS=01F,DEVTYPE=3215
       RDEVICE ADDRESS=(020,8),DEVTYPE=3279,MODEL=3
       RDEVICE ADDRESS=(030,16),DEVTYPE=3705,ADAPTER=BSCA,BASEADD=0B0
       RDEVICE ADDRESS=(040,16),DEVTYPE=3705,ADAPTER=IBM1,BASEADD=0B0
       RDEVICE ADDRESS=(050,16),DEVTYPE=3705,ADAPTER=TELE2,BASEADD=0B0
       RDEVICE ADDRESS=080,DEVTYPE=2955
       RDEVICE ADDRESS=0B0,DEVTYPE=3705,ADAPTER=TYPE4,MODEL=F4,CPTYPE=EP
       RDEVICE ADDRESS=(180,2),DEVTYPE=3420,FEATURE=DUALDENS,MODEL=7
       RDEVICE ADDRESS=190,DEVTYPE=3420,FEATURE=DUALDENS,MODEL=8
       RDEVICE ADDRESS=(200,2),DEVTYPE=3330,MODEL=1
       RDEVICE ADDRESS=(208,2),DEVTYPE=3330,MODEL=1
       RDEVICE ADDRESS=(210,48),DEVTYPE=3330,MODEL=1,FEATURE=SYSVIRT
       RDEVICE ADDRESS=(240,8),DEVTYPE=3350,ALTCU=340
       RDEVICE ADDRESS=2A0,DEVTYPE=3851
       RDEVICE ADDRESS=2D0,DEVTYPE=2305,MODEL=2
       RDEVICE ADDRESS=(330,8),DEVTYPE=3330,MODEL=1
       RDEVICE ADDRESS=(350,8),DEVTYPE=3330,MODEL=1
       RDEVICE ADDRESS=(358,2),DEVTYPE=3330,MODEL=11
       RDEVICE ADDRESS=3D0,DEVTYPE=CTCA
       RDEVICE ADDRESS=(410,48),DEVTYPE=3330,MODEL=1,FEATURE=VIRTUAL
       RDEVICE ADDRESS=(440,16),DEVTYPE=3380
       RDEVICE ADDRESS=4A0,DEVTYPE=3851
       RDEVICE ADDRESS=4B1,DEVTYPE=4245,CLASS=(A)
       RDEVICE ADDRESS=4B8,DEVTYPE=4248,CLASS=(A)
       RCTLUNIT ADDRESS=000,CUTYPE=3811
       RCTLUNIT ADDRESS=008,CUTYPE=2821
       RCTLUNIT ADDRESS=010,CUTYPE=3274
       RCTLUNIT ADDRESS=018,CUTYPE=3215
       RCTLUNIT ADDRESS=020,CUTYPE=3274
       RCTLUNIT ADDRESS=030,CUTYPE=3705,FEATURE=48-DEVICE
       RCTLUNIT ADDRESS=080,CUTYPE=2955
       RCTLUNIT ADDRESS=0B0,CUTYPE=3705
       RCTLUNIT ADDRESS=180,CUTYPE=3803
       RCTLUNIT ADDRESS=190,CUTYPE=3803
       RCTLUNIT ADDRESS=200,CUTYPE=3830,FEATURE=64-DEVICE
       RCTLUNIT ADDRESS=240,CUTYPE=3830,FEATURE=16-DEVICE
       RCTLUNIT ADDRESS=2A0,CUTYPE=3851
       RCTLUNIT ADDRESS=2D0,CUTYPE=2835
```

Figure 37 (Part 1 of 2). Real I/O Configuration File

```
RCTLUNIT ADDRESS=330,CUTYPE=3830,FEATURE=16-DEVICE
RCTLUNIT ADDRESS=340,CUTYPE=3830,FEATURE=16-DEVICE
RCTLUNIT ADDRESS=350,CUTYPE=3830,FEATURE=16-DEVICE,ALTCH=4
RCTLUNIT ADDRESS=3D0,CUTYPE=CTCA
RCTLUNIT ADDRESS=400,CUTYPE=3830,FEATURE=64-DEVICE
RCTLUNIT ADDRESS=440,CUTYPE=3880,FEATURE=16-DEVICE
RCTLUNIT ADDRESS=4A0,CUTYPE=3851
RCTLUNIT ADDRESS=4B0,CUTYPE=4245
RCTLUNIT ADDRESS=4B8,CUTYPE=4248
RCHANNEL ADDRESS=0,CHTYPE=MULTIPLEXOR
RCHANNEL ADDRESS=1,CHTYPE=SELECTOR
RCHANNEL ADDRESS=2,CHTYPE=BLKMPXR
RCHANNEL ADDRESS=3,CHTYPE=BLKMPXR
RCHANNEL ADDRESS=4,CHTYPE=BLKMPXR
RIOGEN CONS=01F,ALTCONS=(010,009)
END
```

Note: It is advisable to code FEATURE=16-DEVICE for 3830 and 3880 control units, even if only 8 devices are attached. If the control unit is physically capable of 16 devices and the 16-DEVICE feature is not coded, missing interrupts will result.

Figure 37 (Part 2 of 2). Real I/O Configuration File

Figure 38 is an example of an IOCP source file. For additional information about the Input/Output Configuration program, refer to "Part 1. Planning for System Generation."

Example: Coding the Input/Output Configuration Source File for the 308x Processor Complex

```
ID    MSG1='SAMPLE IOCP FILE'
CHPID PATH=((00,0,0)),TYPE=BY
CHPID PATH=((01,1,0)),TYPE=BL
CHPID PATH=((02,2,0)),TYPE=BL
CHPID PATH=((03,3,0)),TYPE=BL
CHPID PATH=((04,4,0)),TYPE=BL
CNTLUNIT CUNUMBR=000,PATH=00,SHARED=N,UNIT=3811,                    X
      UNITADD=((00,8))
CNTLUNIT CUNUMBR=001,PATH=00,SHARED=N,UNIT=2821,                    X
      UNITADD=((08,8))
CNTLUNIT CUNUMBR=002,PATH=00,SHARED=N,UNIT=3274,                    X
      UNITADD=((10,8))
*IOCP  DUMMY DEVICE FOR CP ONLY
*IOCP CNTLUNIT CUNUMBR=003,PATH=00,SHARED=N,UNIT=3215,              X
*IOCP       UNITADD=((18,8))
CNTLUNIT CUNUMBR=004,PATH=00,SHARED=N,UNIT=3274,                    X
      UNITADD=((20,8))
CNTLUNIT CUNUMBR=005,PATH=00,SHARED=N,UNIT=3705,                    X
      UNITADD=((30,48))
CNTLUNIT CUNUMBR=006,PATH=00,SHARED=N,UNIT=2955,                    X
      UNITADD=((80,8))
CNTLUNIT CUNUMBR=007,PATH=00,SHARED=N,UNIT=3705,                    X
      UNITADD=((B0))
CNTLUNIT CUNUMBR=010,PATH=01,SHARED=Y,UNIT=3803,                    X
      UNITADD=((80,8))
CNTLUNIT CUNUMBR=011,PATH=01,SHARED=Y,UNIT=3803,                    X
      UNITADD=((90,8))
CNTLUNIT CUNUMBR=020,PATH=02,SHARED=N,UNIT=3830,                    X
      UNITADD=((00,64))
CNTLUNIT CUNUMBR=021,PATH=02,SHARED=N,UNIT=3830,                    X
      UNITADD=((40,16))
CNTLUNIT CUNUMBR=024,PATH=02,SHARED=N,UNIT=3851,                    X
      UNITADD=((A0,8))
CNTLUNIT CUNUMBR=025,PATH=02,SHARED=N,UNIT=2835,                    X
      UNITADD=((D0,8))
CNTLUNIT CUNUMBR=031,PATH=03,SHARED=N,UNIT=3830,                    X
      UNITADD=((30,16))
CNTLUNIT CUNUMBR=032,PATH=03,SHARED=N,UNIT=3830,                    X
      UNITADD=((40,16))
CNTLUNIT CUNUMBR=033,PATH=(03,04),SHARED=N,UNIT=3830,               X
      UNITADD=((50,16))
CNTLUNIT CUNUMBR=034,PATH=03,SHARED=N,UNIT=CTCA,                    X
      UNITADD=((D0,8))
CNTLUNIT CUNUMBR=040,PATH=04,SHARED=N,UNIT=3830,                    X
      UNITADD=((00,64))
CNTLUNIT CUNUMBR=041,PATH=04,SHARED=N,UNIT=3880,                    X
      UNITADD=((40,16)),PROTOCL=S
CNTLUNIT CUNUMBR=042,PATH=04,SHARED=N,UNIT=3851,                    X
      UNITADD=((A0,8))
CNTLUNIT CUNUMBR=043,PATH=04,SHARED=N,UNIT=4245,                    X
      UNITADD=((B0,8))
CNTLUNIT CUNUMBR=044,PATH=04,SHARED=N,UNIT=4248,                    X
      UNITADD=((B8,8))
```

Figure 38 (Part 1 of 2). Input/Output Configuration Program Source File

```
IODEVICE ADDRESS=002,UNIT=3211,CUNUMBR=000
IODEVICE ADDRESS=009,UNIT=3215,CUNUMBR=001
IODEVICE ADDRESS=00C,UNIT=2540R,CUNUMBR=001
IODEVICE ADDRESS=00D,UNIT=2540P,CUNUMBR=001
IODEVICE ADDRESS=00E,UNIT=1403,CUNUMBR=001
IODEVICE ADDRESS=00F,UNIT=1403,CUNUMBR=001
IODEVICE ADDRESS=010,UNIT=3278,MODEL=2A,CUNUMBR=002
*IOCP    DUMMY DEVICE FOR CP ONLY
*IOCP    IODEVICE ADDRESS=01F,UNIT=3215,CUNUMBR=003
IODEVICE ADDRESS=(020,8),UNIT=3279,MODEL=3,CUNUMBR=004
IODEVICE ADDRESS=(030,16),UNIT=3705,CUNUMBR=005
IODEVICE ADDRESS=(040,16),UNIT=3705,CUNUMBR=005
IODEVICE ADDRESS=(050,16),UNIT=3705,CUNUMBR=005
IODEVICE ADDRESS=080,UNIT=2955,CUNUMBR=006
IODEVICE ADDRESS=0B0,UNIT=3705,CUNUMBR=007
IODEVICE ADDRESS=(180,2),UNIT=3420,MODEL=7,CUNUMBR=010
IODEVICE ADDRESS=190,UNIT=3420,MODEL=8,CUNUMBR=011
IODEVICE ADDRESS=(200,2),UNIT=3330,MODEL=1,CUNUMBR=020
IODEVICE ADDRESS=(208,2),UNIT=3330,MODEL=1,CUNUMBR=020
IODEVICE ADDRESS=(210,48),UNIT=3330,MODEL=1,CUNUMBR=020
IODEVICE ADDRESS=(240,8),UNIT=3350,CUNUMBR=(021,032)
IODEVICE ADDRESS=2A0,UNIT=3851,CUNUMBR=024
IODEVICE ADDRESS=2D0,UNIT=2305,MODEL=2,CUNUMBR=025
IODEVICE ADDRESS=(330,8),UNIT=3330,MODEL=1,CUNUMBR=031
IODEVICE ADDRESS=(350,8),UNIT=3330,MODEL=1,CUNUMBR=033
IODEVICE ADDRESS=(358,2),UNIT=3330,MODEL=11,CUNUMBR=033
IODEVICE ADDRESS=3D0,UNIT=CTC,CUNUMBR=034,TIMEOUT=N
IODEVICE ADDRESS=(410,48),UNIT=3330,MODEL=1,CUNUMBR=040
IODEVICE ADDRESS=(440,16),UNIT=3380,CUNUMBR=041
IODEVICE ADDRESS=4A0,UNIT=3851,CUNUMBR=042
IODEVICE ADDRESS=4B1,UNIT=4245,CUNUMBR=043
IODEVICE ADDRESS=4B8,UNIT=4248,CUNUMBR=044
```

Figure 38 (Part 2 of 2). Input/Output Configuration Program Source File

Example of Coding an SNA Controller

CP treats each of the following devices in the same way:

- 3705 running NCP
- 3725 running NCP
- 3274 local SNA (Model 41A) controller

CP does not communicate directly with these devices. Native VM/VTAM or a guest machine running an SNA access method is required to communicate with these control units. (CP also does not communicate directly with the NCP portion of a PEP controller and this environment follows the following rules as well.)

To define these SNA controllers to VM/SP, use the following RDEVICE specifications:

```
RDEVICE ADDRESS=0F0, DEVTYPE=3725, ADAPTER=TYPE5,       X
MODEL=2, CPTYPE=NCP
RDEVICE ADDRESS=0E0, DEVTYPE=3705, ADAPTER=TYPE4,       X
MODEL=E8, CPTYPE=NCP
```

where:

ADDRESS should specify the native subchannel address of the 37X5 controller.

DEVTYPE specifies the controller type. This is really for documentation purposes only. For a 3274-41A, either of the above examples will work. In fact, the 3705 definition will work for the 3725 and vice-versa.

ADAPTER specifies the 37X5 adapter type. For a 3274, use one of the above definitions.

MODEL is device dependent and must be specified. This parameter, however, is a leftover from when CP loaded an EP control program. It is ignored by CP when an NCP or PEP is to be loaded.

CPTYPE = NCP tells CP not to load an EP. You can specify either this or CPTYPE = PEP.

Chapter 22. Preparing the System Name Table File (DMKSNT)

The system name table consists of entries that identify the name and location of saved systems. Three macros generate entries for the system name table:

- The NAMESYS macro creates an entry in the system name table for a virtual machine operating system or saved segment.
- The NAMENCP macro creates an entry in the system name table for a 3704/3705 control program.
- The NAME3800 macro creates an entry in the system name table for a 3800 named system.

The system name table is assembled by a system programmer. It describes the system to be saved via the SAVESYS command and to be IPLed by name. Shared segments can be specified. These segments must consist of all reentrant code.

The system name table is a pageable module.

Sample DMKSNT entries provided with the starter systems and sample allocations provided in the system generation procedure are found in the *VM/SP HPO System Definition Files.*

If you create your own version of the system name table, your file must have a CSECT statement and an END statement. You must group like macros together in the DMKSNT CSECT; that is, you must group all NAMESYS macros together, all NAMENCP macros together, and all NAME3800 macros together. If you arrange the macros in any other order, the system generates an MNOTE warning you that the macros are out of sequence.

You should use the DMKSNT module as a pageable module. If you make this module resident and less than one page, you should punch an SPB (Set Page Boundary) LOADER CARD at the beginning of the DMKSNT to assure alignment at a page boundary. The loader automatically inserts an SPB card to force the DMKSNT module to a page boundary when you build the CP system.

Information about coding the NAMESYS, NAMENCP, and NAME3800 macros follows.

Coding the NAMESYS Macro

The NAMESYS macro describes the name and location of the saved system or discontiguous saved segment. Shared segments may be specified, but they must consist of reenterable code, with no alteration of its storage space permitted.

The format of the NAMESYS macro is:

```
label   NAMESYS   SYSSIZE=nnnnnK,SYSNAME=name,[VSYSRES=cccccc,]
                  VSYSADR=[cuu   ],SYSVOL=cccccc,{SYSCYL=nnn,    }
                          [IGNORE]               {SYSBLOK=nnnnnn,}
                  SYSSTRT={(ccc,p),} [SYSPGCT=pppp,]
                          {pppppp, }
                  SYSPGNM=(nn,nn,nn-nn,...),
                  SYSHRSG=(s,s,...),
                  PROTECT={OFF}
                          {ON }
                  VMGROUP={YES}
                          {NO }
                  [USERID=userid,][RCVRID=rcvrid,][SAVESEQ={nnn}]
                                                  {10 }
```

where:

label
is any desired user label.

SYSSIZE = nnnnnK
is up to five decimal digits representing the minimum storage you must have available in order to IPL the saved system. K must be specified. Although you must code this operand for discontiguous saved segments, it is not used for them.

SYSNAME = name
is the name (up to eight alphameric characters) given to the system or segment to be used for identification by the SAVESYS and/or IPL commands. The name selected must not be one that could be interpreted as a hexadecimal device address (for example, A or E).

VSYSRES = cccccc
is the real volume serial number (up to six alphameric characters) of the DASD volume containing the minidisk that is the system residence volume of the system to be saved. This operand is ignored if VSYSADR = IGNORE, but you must specify it as null (VSYSRES =,). This operand is flagged and ignored if USERID is specified.

VSYSADR = cuu
is the virtual address of the minidisk that is the system residence volume of the system to be saved. This operand defaults to IGNORE if USERID is specified. Other values are flagged. If the system being saved requires a system residence, this operand must be coded with its virtual address. "VSYSADR = IGNORE" must be coded when you are defining a discontiguous saved segment. It may also be used when defining a saved system to improve performance.

Note: If you are likely to have a large number of CMS users active at one time, you should distribute CMS activity over two volumes by

- *Setting up a second CMS system residence volume and dividing the users between the two CMS system residence volumes*

  or

- *Putting your program products on one spindle and CMS non-resident commands on another spindle.*

VSYSADR=IGNORE
indicates that the NAMESYS macro is describing a system or segment that does not require a virtual system residence volume. Code VSYSADR=IGNORE when you are defining a discontiguous saved segment.

SYSVOL=cccccc
is the volume serial number (up to 6 alphameric characters) of the DASD volume designated to receive the saved system or segment. This must be a CP-owned volume.

{SYSCYL=nnn, }
{SYSBLOK=nnnnnn,}
is the real starting location of the virtual disk (specified by VSYSRES and VSYSADR) that is the system residence volume for the system to be saved.

- For count-key-data DASD, specify the SYSCYL= parameter.
- For fixed-block DASD, specify the SYSBLOK= parameter.

This operand is flagged and ignored if USERID= is specified. It is also ignored if VSYSADR=IGNORE, but in this case you must specify it as null (SYSCYL=, or SYSBLOK=,).

SYSSTRT= {(ccc,p), }
{ppppp,}
is the starting location on SYSVOL where this named system is to be saved. During SAVESYS and IPL processing, this location is used to generate DASD addresses for I/O operations.

- For count-key-data SYSVOL devices, specify (ccc,p), where cc designates the starting cylinder address and p the starting page address. Since the page numbers of a cylinder start with 1, not 0, p must be greater than or equal to 1. cc and p are decimal numbers.
- For fixed-block SYSVOL devices, specify ppppp, representing the starting page address. ppppp must be greater than or equal to 1. ppppp is a decimal number.

The number of pages written to this area is the total number you specify on the SYSPGCT operand, plus up to three information pages. See Figure 10 on page 40 for a table showing the number of information pages CP needs, which depends on the number of virtual machine pages you save. See "Relationship of Page Numbers, Segment Numbers, and Hexadecimal Addresses" on page 300 for help converting segment and page numbers to hexadecimal addresses.

SYSPGCT=pppp
is the exact number of virtual machine pages (pppp) to be saved (that is, the total number of pages you indicate via the SYSPGNM operand). Information pages are not counted. This is a decimal number from 1 to 4096. The SYSPGCT operand is optional. If you do not specify it, the NAMESYS macro calculates the number of pages to be saved.

SYSPGNM=(nn,nn,nn-nn,...)
are numbers of the pages to be saved. Pages may be specified singly or in groups. For example, if pages 0, 4, and 10 through 13 are to be saved, use the format:

```
SYSPGNM=(0,4,10-13)
```

The total must be equal to the SYSPGCT specification.

SYSHRSG=(s,s,...)
are segment numbers designated as shared (numbered from zero up, with the first segment, for example, specified as 0). Pages in these segments are set at IPL time to be used by any user loading by this name. All shared segments must be reenterable. The maximum number of defined shared segments is 78. This operand is flagged and ignored if USERID is specified.

PROTECT= { **OFF** / **ON** }
indicates that CP is to run either with protected (ON) or unprotected (OFF) shared segments for the named system. ON is the default. If a named system is specified as unprotected, any changes made to shared pages in the named system will not be detected by CP; the change will be seen by all users of the shared page.

Note: On the 308x Processor Complex, PROTECT=ON specifies that the hardware segment protection feature is to protect segments. On other processors, PROTECT=ON specifies that CP is to protect segments. (See "Shared Segment Protection" on page 137.)

Virtual machines in a Virtual Machine Group always run with PROTECT=OFF. If you specify both PROTECT=ON and VMGROUP=YES, an MNOTE is generated indicating that PROTECT=OFF is assumed.

VMGROUP={YES / NO}

determines if the saved system being defined is to be treated as a Virtual Machine Group. Specifying VMGROUP=YES indicates that all virtual machines that access the shared segment will become members of the Virtual Machine Group. With VMGROUP=YES specified, virtual machines may access the segment only by using the IPL command. Users cannot access the saved system using the FINDSYS or LOADSYS functions of Diagnose X'64'. The VMGROUP=YES specification also forces PROTECT=OFF.

The VMGROUP=NO specification indicates that the shared segment will not be treated as a Virtual Machine Group. This is the default setting.

USERID=userid

is the userid of the virtual machine that is saved in the designated area. (This user can IPL from that area.) Any value specified for for USERID indicates that this is a VMSAVE target area specification.

Note: Multiple target areas may be specified for a single userid by including more than one NAMESYS macro with the same USERID= parameter.

RCVRID=rcvrid

is the userid of the virtual machine authorized to access a system save area. This is an optional parameter. It defaults to the value specified for USERID. The parameter is flagged with an MNOTE and ignored if specified when USERID is not specified.

SAVESEQ={nnn / 10}

specifies the order in which multiple virtual machines will be saved. Valid values are from 0 to 255. The virtual machine with the lowest number is saved first. If two virtual machines have the same value, the one that enabled the VMSAVE option first is dumped first. This parameter is flagged with an MNOTE and ignored if specified when USERID is not specified. If the SAVESEQ priority value is not supplied, the default value is 10.

Available Space for Each DASD Type

The number of 4K pages available per DASD cylinder is:

Pages/Cylinder	DASD Type
24	3340-35, 3340-70, 2305
32	2314, 2319
57	3330, 3333
120	3350 (in native mode)
96	3375
150	3380

Relationship of Page Numbers, Segment Numbers, and Hexadecimal Addresses

Since the NAMESYS macro requires you to specify page and segment numbers, and the DOSGEN, SAMGEN, and VSAMGEN procedures require you to enter hexadecimal addresses, you may find the following reference information useful.

```
 1K =   1024 = X' 400'
 4K =   4096 = X' 1000' =    1 page
64K = 65536 = X'10000' = 16 pages = 1 segment
```

To convert a page number to a segment number, divide the page number by 16. Since one segment is 10000 in hexadecimal, then 20000 is segment 2, 100000 is segment 16, 180000 is segment 24, and so on.
Appendix E, "Worksheet to Aid in Coding the NAMESYS Macro" on page 427 contains a worksheet to help you convert segment and page numbers to hexadecimal addresses.

More Information

Information on the following subjects is in the *VM/SP HPO System Programmer's Guide*:

- Saving the CMS system
- Saved system restrictions for CMS
- Saving OS
- Determining when to save a system
- Using the SAVESYS command

Coding the NAMENCP Macro

This section applies only to EPs as defined, created, and loaded via VM/SP HPO. If you have a 3725 communications controller or a 3705 that will be loaded via ACF/SSP, refer to the following manuals:

ACF/NCP-SSP, V3 Installation and Resource Definition Guide, SC30-3253 (explains how to generate and load the NCP)

EP/3725 Installation and Resource Definition Guide and Reference, SC30-3172

EP/3705 Generation and Utilities Guide and Reference, GC30-3242

ACF/NCP V4, ACF/SSP V3 Diagnosis Guide, SC30-3255 (explains how to dump the contents of the 37X5 and how to run ACF/TAP and CRP).

You must create an entry in the system name table (DMKSNT) for each unique 3704/3705 control program that you generate. You must specify shared segments, but the segments must consist of reentrant code. If you can foresee generating several versions of the 3704/3705 control program, define extra entries in the system name table when you generate VM/SP HPO. In this way, you do not have to regenerate the VM/SP HPO system just to update the system name table.

Use the NAMENCP macro to define 3704/3705 program entries in the system name table.

The format of the NAMENCP macro is:

```
label   NAMENCP   CPSIZE=nnnK,
                  CPNAME=ncpname,
                  CPTYPE=[EP]
                  SYSPGCT=pp,
                  SYSVOL=volser,
                  SYSSTRT={(ccc,p)}
                          {pppppp }
```

where:

label
is any desired user label.

CPSIZE = nnnK
is the storage size of the 3704/3705 specified during 3704/3705 control program generation. A maximum of 256K can be specified.

CPNAME = ncpname
is the name of the 3704/3705 control program image. This name is used in the SAVENCP and NETWORK LOAD commands. The name must be from one- to eight-alphameric characters.

CPTYPE = [EP]
is the 3704/3705 control program type.

SYSPGCT = pp
is the total number of pages (pp) to be saved. This value may be equal to the number of pages implied by the CPSIZE operand plus four pages for control information, but it must not exceed that total.

SYSVOL = volser
is the volume serial number (volser) of the DASD volume designated to receive the CP image. That volume must be CP-owned.

SYSSTRT = {(ccc,p) / ppppp}
is the starting location on SYSVOL where this named system is to be saved. During SAVESYS and IPL processing this is used to generate the DASD addresses for I/O operation.

- For count-key-data SYSVOL devices, specify (ccc,p), where ccc designates the starting cylinder address and p the starting page address. Since the page numbers of a cylinder start with 1, not 0, p must be greater than or equal to 1. ccc and p are decimal numbers.

- For fixed-block SYSVOL devices, specify ppppp, the starting page address. ppppp is a decimal number and must be greater than or equal to 1.

Coding the NAME3800 Macro

The NAME3800 macro describes the name and location of the named system that will contain the 3800 character arrangement tables, graphic modifications, FCBs, and copy modifications for the 3800 printers. Multiple named systems may be specified. The 3800's RDEVBLOK contains a pointer to the named system currently in use for that particular 3800.

The format of the NAME3800 macro is:

```
label   NAME3800   CPNAME=libname,
                   SYSPGCT=pppp,
                   SYSVOL=volser,
                   SYSSTRT= {(ccc,p)}
                            {pppppp }
```

where:

label
is any desired user label.

CPNAME = libname
is the name of the 3800 image library. This name is used in the IMAGELIB or IMAGEMOD command. The name must be from one- to eight-alphameric characters.

SYSPGCT = pppp
is the total number of pages (pppp) to be saved for the image library. This value is a decimal number up to four digits. To determine the number of pages to be saved, use the following steps:

1. The image library consists of one or more members. These members are core images of the TEXT decks created by GENIMAGE. To find the size of the members, issue the CMS LOAD and GENMOD commands for each TEXT deck created by GENIMAGE. The eleventh and twelfth bytes (that is, bytes 10 and 11) of the resulting MODULE files contain a hexadecimal representation of each member's size in bytes. Add eight bytes to each member's size.

2. Sum the sizes and add 16 bytes to the total.

3. Divide the total by 4096 bytes to achieve the page count (pppp). Be sure to round up to the next whole page.

SYSVOL = volser
is the volume serial number (volser) of the DASD volume designated to receive the 3800 image library. The volume must be a CP-owned volume.

SYSSTRT = $\left\{\begin{matrix}\textbf{(ccc,p)} \\ \textbf{ppppppp}\end{matrix}\right\}$

is the starting location on SYSVOL at which this image library is to be saved.

- For count-key-data SYSVOL devices, specify (ccc,p), where ccc designates the starting cylinder address and p the starting page address. Since the page numbers of a cylinder start with 1, not 0, p must be greater than or equal to 1. ccc and p are decimal numbers.
- For fixed-block SYSVOL devices, specify pppppp, the starting page address. pppppp is a decimal number and must be greater than or equal to 2.

Chapter 23. Preparing the CP System Control File (DMKSYS)

The CP system control file consists of macro statements that describe the following:

- CP system residence device
- System storage size
- CP-owned direct access devices
- Allocation of CP-owned volumes
- System operator's user identification
- System timer value
- System pointer variables
- Automatic performance monitoring parameters
- Accounting parameters
- System identification
- System spool parameters
- Security journalizing parameters
- System dump space parameters
- System T-DISK security parameters
- Missing I/O interruption timer intervals

The installation is responsible for ensuring the presence and accuracy of the macros described below.

The DMKSYS ASSEMBLE file provided with the starter system does not assemble properly unless you have reserved adequate space for the CP nucleus.

The file should be created in the order shown:

```
DMKSYS  CSECT
        SYSOWN   macro
        SYSPAG   macro(s) (optional)
        SYSRES   macro
        SYSMON   macro
        SYSJRL   macro
        SYSCOR   macro
        SYSOPR   macro
        SYSACNT  macro
        SYSTIME  macro
        SYSFORM  macro
        SYSPCLAS macro
        SYSID    macro
        SYSMIH   macro (optional)
        SYSFCN   macro
        SYSLOCS  macro
        END
```

Notes:

1. *Samples of the DMKSYS and DMKSNT files are provided with the starter system and are listed in the* VM/SP HPO System Definition Files. *If you use these, you can save the CMS system at the end of system generation. You may modify the DMKSYS file if you wish. For example, you could modify the starter system DMKSYS file to add other CP-owned volumes.*

2. *You must code all macro statements, except for SYSPAG and SYSMIH, even if you code no operands on some macro statements. If SYSPAG is coded an area-specification (volid with optional first/last cylinders) must be coded.*

3. *If you code SYSPAG macros, they must follow the SYSOWN macro and precede the SYSRES macro.*

4. *If the SYSMIH macro is coded, it must immediately precede the SYSFCN macro.*

5. *SYSLOCS must always be the last macro coded.*

The 3330 sample DMKSYS module supplied with the starter system is:

```
        PRINT   NOGEN
DMKSYS  CSECT
        SYSOWN  VMSRES,                                                 X
                VMSEXT,                                                 X
                VMPK01,                                                 X
                VMGCS1,                                                 X
                PROFPK
        SYSPAG  (VMPK01,275,334),TYPE=PS
        SYSRES  SYSVOL=VMSRES,                                          X
                SYSRES=123,                                             X
                SYSTYPE=3330,                                           X
                SYSCLR=YES,                                             X
                SYSNUC=397,                                             X
                SYSWRM=(391,2),                                         X
                SYSERR=(393,2),                                         X
                SYSCKP=(198,2)
        SYSMON  USERID=VMAP,                                            X
                AUTO=NO,                                                X
                BUFFS=3,                                                X
                TIME=(08:00,17:00),                                     X
                CLASS=M,                                                X
                ENABLE=(PERFORM,USER,DASTAP),                           X
                LIMIT=(50000,NOSTOP)
        SYSJRL
        SYSCOR  RMSIZE=4M,                                              X
                AP=NO,                                                  X
                MP=NO
        SYSOPR  SYSOPER=OPERATOR,                                       X
                SYSDUMP=OPERATNS
        SYSACNT USERID=DISKACNT,                                        X
                OUTPUT=READER,                                          X
                CLASS=C,                                                X
                LIMIT=100
        SYSTIME ZONE=4,                                                 X
                LOC=WEST,                                               X
                ID=EDT
        SYSFORM
        SYSPCLAS
        SYSID
        SYSMIH
        SYSFCN
        SYSLOCS
        END
```

The 3350 sample DMKSYS module supplied with the starter system is:

```
        PRINT   NOGEN
DMKSYS  CSECT
        SYSOWN  VMSRES,                                          X
                VMPK01,                                          X
                VMGCS1,                                          X
                PROFPK
        SYSPAG  (VMPK01,277,326),TYPE=PS
        SYSRES  SYSVOL=VMSRES,                                   X
                SYSRES=123,                                      X
                SYSTYPE=3350,                                    X
                SYSCLR=YES,                                      X
                SYSNUC=549,                                      X
                SYSWRM=(545,2),                                  X
                SYSERR=(547,2),                                  X
                SYSCKP=(277,1)
        SYSMON  USERID=VMAP,                                     X
                AUTO=NO,                                         X
                BUFFS=3,                                         X
                TIME=(08:00,17:00),                              X
                CLASS=M,                                         X
                ENABLE=(PERFORM,USER,DASTAP),                    X
                LIMIT=(50000,NOSTOP)
        SYSJRL
        SYSCOR  RMSIZE=4M,                                       X
                AP=NO,                                           X
                MP=NO
        SYSOPR  SYSOPER=OPERATOR,                                X
                SYSDUMP=OPERATNS
        SYSACNT USERID=DISKACNT,                                 X
                OUTPUT=READER,                                   X
                CLASS=C,                                         X
                LIMIT=100
        SYSTIME ZONE=4,                                          X
                LOC=WEST,                                        X
                ID=EDT
        SYSFORM
        SYSPCLAS
        SYSID
        SYSMIH
        SYSFCN
        SYSLOCS
        END
```

The 3375 sample DMKSYS module supplied with the starter system is:

```
        PRINT   NOGEN
DMKSYS  CSECT
        SYSOWN  VMSRES,                                                 X
                VMPK01,                                                 X
                VMGCS1,                                                 X
                PROFPK
        SYSPAG  (VMPK01,277,326),TYPE=PS
        SYSRES  SYSVOL=VMSRES,                                          X
                SYSRES=123,                                             X
                SYSTYPE=3375,                                           X
                SYSCLR=YES,                                             X
                SYSNUC=953,                                             X
                SYSWRM=(947,2),                                         X
                SYSERR=(949,2),                                         X
                SYSCKP=(479,2)
        SYSMON  USERID=VMAP,                                            X
                AUTO=NO,                                                X
                BUFFS=3,                                                X
                TIME=(08:00,17:00),                                     X
                CLASS=M,                                                X
                ENABLE=(PERFORM,USER,DASTAP),                           X
                LIMIT=(50000,NOSTOP)
        SYSJRL
        SYSCOR  RMSIZE=4M,                                              X
                AP=NO,                                                  X
                MP=NO
        SYSOPR  SYSOPER=OPERATOR,                                       X
                SYSDUMP=OPERATNS
        SYSACNT USERID=DISKACNT,                                        X
                OUTPUT=READER,                                          X
                CLASS=C,                                                X
                LIMIT=100
        SYSTIME ZONE=4,                                                 X
                LOC=WEST,                                               X
                ID=EDT
        SYSFORM
        SYSPCLAS
        SYSID
        SYSMIH
        SYSFCN
        SYSLOCS
        END
```

The 3380 sample DMKSYS module supplied with the starter system is:

```
        PRINT   NOGEN
DMKSYS  CSECT
        SYSOWN  VMSRES,                                          X
                VMPK01,                                          X
                VMGCS1,                                          X
                PROFPK
        SYSPAG  (VMPK01,337,426),TYPE=PS
        SYSRES  SYSVOL=VMSRES,                                   X
                SYSRES=123,                                      X
                SYSTYPE=3380,                                    X
                SYSCLR=YES,                                      X
                SYSNUC=881,                                      X
                SYSWRM=(877,2),                                  X
                SYSERR=(879,2),                                  X
                SYSCKP=(442,1)
        SYSMON  USERID=VMAP,                                     X
                AUTO=NO,                                         X
                BUFFS=3,                                         X
                TIME=(08:00,17:00),                              X
                CLASS=M,                                         X
                ENABLE=(PERFORM,USER,DASTAP),                    X
                LIMIT=(50000,NOSTOP)
        SYSJRL
        SYSCOR  RMSIZE=4M,                                       X
                AP=NO
        SYSOPR  SYSOPER=OPERATOR,                                X
                SYSDUMP=OPERATNS
        SYSACNT USERID=DISKACNT,                                 X
                OUTPUT=READER,                                   X
                CLASS=C,                                         X
                LIMIT=100
        SYSTIME ZONE=4,                                          X
                LOC=WEST,                                        X
                ID=EDT
        SYSFORM
        SYSPCLAS
        SYSID
        SYSMIH
        SYSFCN
        SYSLOCS
        END
```

Performance Considerations for Coding the DMKSYS File Macros

The following recommendations may help reduce arm and channel contention, and may improve the performance of a VM/SP HPO system.

- Provide separate CP volumes for paging and spooling and have the volumes mounted on separate channels.

- If you have a heavy I/O production virtual machine (for example, one that is executing OS/VS1 or VSE/AF), try to keep all its major I/O devices on a separate channel from a channel handling the CMS system residence volume or other users' disks.

- If possible, put read-only minidisks (for example, the CMS system residence disk and source disks) that are frequently accessed on on a 3880 Model 13 or 23 Storage Subsystem. Whatever device you put them on, try to keep them on separate volumes from users' read/write minidisks. If possible, also keep them on separate channels.

- If your installation is likely to have a large number of CMS users active at one time, you should distribute CMS activity over two volumes by (1) setting up a second CMS system residence volume and dividing the users between the two CMS system residence volumes or (2) putting your program products on one spindle and CMS nonresident commands on another spindle.

- If you have disks with fixed-head areas (2305, 3340, 3350), or with areas that are part of high-speed cache devices, you should use these areas for preferred paging. To do this, allocate the fixed-head or high-speed cache areas as PAGE space, using the Format/Allocate program, and specify them as TYPE=PP on a SYSPAG macro. Allocate other disk areas used for paging as PAGE space, using the Format/Allocate program, and specify them as TYPE=PG on a SYSPAG macro. For more details, see the *VM/SP HPO Operator's Guide.*

- The relative amounts of storage for dynamic paging and free storage can be optimized by using the FREE operand of the SYSCOR macro statement. You should allocate one page of fixed free storage for each virtual machine that is logged on, based on the average number of users that you expect to have logged on at any one time.

- Extended storage support, which reduces contention for CP pages in the dynamic paging area, will improve performance in processors using this feature. The value specified in the RMSIZE operand in the SYSCOR macro controls the balance between the dynamic paging area and the V=R preferred machine assist guest in the storage area above 16 megabytes.

- Using the automatic monitoring facilities, study the load environment and performance profile for your system as soon as possible. These facilities, used with programs similar to the VM Monitor Analysis Program (5664-191), are designed to make data collection and reduction easy, thereby allowing the analyst to concentrate on analysis. Data collection can be performed on a regular basis by specifying AUTO=YES on the SYSMON macro instruction. The system assumes default values for other operands if none are specified.

- The installation can determine the monitoring interval for the missing interruption handler support. Some classes of devices in your installation may require an interval other than the default interval provided in module DMKSYS. For instance, if your installation has many direct access storage devices, you may need to lengthen the time interval for the DASD class--this would eliminate unnecessary missing interruption handler processing for devices that are functioning properly. Be careful not to lengthen the time interval too much, since you may lose the usefulness of missing interruption monitoring.

  The SYSMIH macro statement controls the time intervals for missing interruption monitoring. If you have already generated the system, you can dynamically change the time intervals with the SET MITIME command.

- The VM Real Time Monitor (SMART), listed in Appendix A, "Program Products, Installed User Programs, Field-Developed Programs, and Emulators" is another program designed specifically to help you monitor and tune your system.

SYSOWN Macro

Use the SYSOWN macro to generate the list of up to 255 CP-owned DASD volumes. A CP-owned volume is either the CP system residence volume, or a volume that contains paging, spooling, dump, directory, or temporary disk space. It must contain a CP allocation table allocating these areas at cylinder 0, record 4 for count-key-data devices, or blocks 3 and 4 for FB-512 devices. Even if a volume has an allocation table in the appropriate area, allocation data is ignored unless the volume appears as an operand in the SYSOWN macro instruction.

Note: The SYSOWN macro must appear before the SYSRES macro in the assembly listing.

The name field must not be specified for the SYSOWN macro.

The format of the SYSOWN macro is:

Name	Operation	Operands
	SYSOWN	volid,[volid,...]

where:

volid
is the CP-owned volume identification of from one to six alphameric characters.

Example

The following is an example of the SYSOWN macro:

```
SYSOWN VMSRES, VMSTGE
```

Special Considerations for Allocating Space on CP-Owned Volumes: The following considerations should help you to allocate space efficiently on CP-owned volumes:

- The basic allocation and formatting of CP-owned volumes is accomplished by the CP Format/Allocate Service Program (DMKFMT). The SYSPAG macro further defines these volumes for swapping, preferred paging, general paging, page migration, spooling, and dump. The number and order of SYSPAG macros, specified in DMKSYS, establishes an access structure for efficient use of these volumes. For a detailed description of the use of the SYSPAG macro and the associated access structure, refer to the SYSPAG macro in this chapter.

- If a volume is specified in a SYSOWN statement but is not mounted when the generated system is loaded (via the IPL command), that volume is considered unavailable to the system. Processing continues, if possible. The operator can mount and attach the volume later, if it is needed.

- Only volumes that contain paging, spooling, dump, directory, or T-DISK space need be identified as CP-owned volumes. All other volumes are described either by directory entries or by logically attaching the entire device.

- If you add another volume to the SYSOWN list, you must add it at the end of the list. (Otherwise, if you attempt a warm start after regenerating and loading CP, the relative entry number used to locate system spool buffers is incorrect.) Then reassemble DMKSYS, rebuild the CP nucleus, and reload it on the system residence volume.

- If your installation has saved systems (systems that can be loaded by name, thus bypassing the initial program load procedure), you must reserve space on a CP-owned volume to hold the named systems you want saved. The DASD space you reserve, for each named system you wish to save, should be enough to contain the number of pages specified in the SYSPGCT operand of the NAMESYS macro, plus one page for system use.

- If your VM/SP HPO system has a 3704, 3705, or 3725, you must reserve space on a CP-owned volume to contain the 3704/3705 control program image. See Chapter 14, "Planning for the 3704, 3705, and 3725 Communications Controllers" for information about how much DASD space you should reserve.

- If there is more than one directory area on the volumes in the SYSOWN list, the area with the lowest address will be the system directory.

SYSPAG Macro

The SYSPAG macro is used to define the allocation of CP-owned volumes for swapping, paging, spooling, and system dumps. The basic allocation and formatting of CP-owned volumes is accomplished by the Format/Allocate Service Program (DMKFMT). The SYSPAG macro further defines these volumes for swapping, preferred paging, general paging, page migration, spooling, and dump spool files. SYSPAG macros must follow the SYSOWN macro and precede the SYSRES macro. The number and order of the SYSPAG macros, specified in DMKSYS, establishes an access structure for efficient use of these volumes.

The SYSPAG macro also allocates Paging Storage for swapping and preferred paging.

Paging Storage, also called Expanded Storage, is a high-speed storage facility which serves as a paging assist for the 3090 Processor Complex. Paging Storage reduces system response time while providing increased system security, since only the control program has access to it. No system interfaces permit virtual machines to access the data in Paging Storage. Any virtual machine attempting to move a page into or out of Paging Storage will receive an operation exception.

The format of the SYSPAG macro is:

Name	Operation	Operands
	SYSPAG	{ volid, (volid[,[first-cyl][,[last-cyl]]]), ... PAGESTORE (PAGESTORE[,[first-incre][,[last-incre]]]), ... } TYPE= { SW PP PG[1] PM[1] PS[1] DU[1] } [,SWSIZE= { 1-99 8 }] [2]

[1]Not valid for Paging Storage.

[2]Specified only on the first SYSPAG macro of TYPE=SW.

where:

The area-specification operands (volid, first-cyl, last-cyl, PAGESTORE, first-incre, last-incre) specify a complete volume, a range of cylinders within a volume, all of Paging Storage, or a range of increments of Paging Storage. One of the first two formats must be used to define space on a CP-owned volume. The last two formats are used to define Paging Storage.

You must not allocate multiple areas of the same CP-owned volume on one SYSPAG macro. You may allocate areas of different CP-owned volumes, or multiple Paging Storage areas, or both, on the same SYSPAG macro.

volid
specifies the serial number (up to six characters) of a CP-owned volume. Volids of greater than six characters are truncated on the right.

first-cyl
specifies the first cylinder of the area to be defined. The default is 0. For FBA volumes this value specifies the first pseudo-cylinder of the area to be defined. For a description of pseudo-cylinders and a method of conversion for FBA devices, see "FB-512 Device Geometry" on page 36 and "FBA Page/Pseudo-Cylinder Conversion" on page 37.

last-cyl
specifies the last cylinder of the area to be defined. This value must be greater than or equal to the value specified for the first-cyl operand. For FBA volumes this value specifies the last pseudo-cylinder of the area to be defined. The default is end-of-volume.

PAGESTORE
indicates that a Paging Storage area is being defined. If you do not enter this operand exactly as shown above, the system will assume that the area-specification operand defines an area on a CP-owned paging device. PAGESTORE can be specified only on SYSPAG macros of TYPE=PP, or on the first SYSPAG macro of TYPE=SW.

first-incre
defines the first increment of Paging Storage to be allocated to the TYPE specified. Increments are numbered from 1 to a maximum of 64. The default is increment 1.

last-incre
defines the last increment of Paging Storage to be allocated. This value must be greater than or equal to the value specified for the first increment. The default is the last increment of Paging Storage attached to the 3090 Processor.

If neither the first increment nor the last increment is specified, all available Paging Storage is allocated.

Coding Considerations

- The maximum number of area specifications in all the SYSPAG macros specified cannot exceed 255.
- Except for PAGESTORE, a given volid cannot be specified more than once on the same SYSPAG macro.

TYPE = xx
specifies the type of area to be defined as:

SW — Swap area
PP — Preferred paging area
PG — General paging area
PM — Page migration area
PS — Spooling area
DU — Dump area

Coding Considerations

- All SYSPAG macros of the same type must be grouped and placed in DMKSYS in the sequence shown above (SW, PP, ...).
- To use Paging Storage to contain swap set pages, code a type of SW. (You may also define DASD devices as TYPE = SW). To use Paging Storage for preferred paging, code a type of PP. (You may also define DASD devices as TYPE = PP). No other type codes are permitted for Paging Storage.
- Paging Storage can be specified only on the first SYSPAG macro of TYPE = SW. It can be specified on any SYSPAG macro of TYPE = PP.
- When a SW (swap) area is not specified, or if the designated area becomes full, the working set pages will be paged to the paging areas.
- Swap (SW) areas should be specified on Paging Storage, on high-speed, large-capacity devices (3350s, 3375s, or 3380s), or on a 3880 Model 11 or 21 Storage Subsystem.
- Swap areas are not supported on 3380s and 3375s with the speed matching buffer on low-speed channels.
- Do not use a 3880 Model 13 or Model 23 Storage Subsystem for paging, spooling, swapping, or dumping unless you have disabled the cache by issuing the CACHE OFF command.
- Cached devices used for preferred paging should have a ratio of backing storage to cache close to 1:1.
- Cached devices used for page migration, or for general paging when little overflow to page migration areas is expected, should have a ratio of backing storage to cache between 2:1 and 4:1.
- If you are migrating from another VM/SP or VM/SP HPO system, the size of the total area specified for SW should be proportional to the area allocated to preferred paging in that system.
- For further considerations, see "Page Migration" on page 320.

SWSIZE = nn
specifies the number of pages (nn = 1-99) in a swap set. The default is 8.

Coding Considerations

- SWSIZE must be specified on the first SYSPAG macro of TYPE = SW. It cannot be specified on any other SYSPAG macro.
- For best performance, the swap set size should *not* divide evenly into the track size.
- For Paging Storage, the swap set *should* divide evenly into the the pseudo-cylinder size (128).
- The swap set size should be less than the total number of pages in the user's entire working set to allow parallel processing of the user's swap sets.
- If the swap set size exceeds the cylinder capacity of the device, the device will not be used for swapping.

General Coding Considerations

Follow these recommendations when you code SYSPAG macros:

- In order to use all of Paging Storage before using any DASD, allocate only Paging Storage on the first SYSPAG macro of type SW or PP.
- Do not code more than one area specification for a given CP-owned volume on a single SYSPAG macro. If you do, you will receive a warning MNOTE; and only the first area you specify will be allocated. You may specify multiple areas of Paging Storage on a single SYSPAG macro.
- To minimize DASD access time, do not allocate certain types of areas on the same volume. Follow these guidelines:
  - SW areas should only be on volumes used for swapping.
  - PP, PG, and PM areas can be on the same volume, but no more than one of each.
  - PS and DU areas can be on the same volume.
- All paging and spooling areas allocated by DMKFMT should be defined by area-specifications in SYSPAG macros. The areas not defined are placed on a default paging and spooling level. The default assignment of these areas may result in two or more areas on the same volume for the same type (SW, PP, PG, ...) and could increase DASD access time. Message DMKVDG341I is issued for each defaulting area. For the assignment of these areas, see "SYSPLIST Defaults" on page 322.

- Any Paging Storage that you do not allocate with SYSPAG macros will not be used for swapping or preferred paging. It will not be allocated by default. Remember that you can allocate more than one Paging Storage area on a single SYSPAG macro.

For SYSPAG macro coding restrictions, see Appendix D, "VM/SP HPO Restrictions" on page 403.

Format/Allocate (DMKFMT) Considerations

The following recommendations should be followed when formatting CP-owned volumes and coding associated SYSPAG macros:

- Cylinders that will be mapped by SYSPAG macros as TYPE=SW, PP, PG, or PM should be allocated by DMKFMT as PAGE type cylinders. This avoids the possibility of having the CP dump file allocated in a swapping or paging area by DMKCPI. In addition, it avoids the possibility of spool files in the paging areas when the system is initialized.

- Cylinders allocated as TDSK, PERM, or DRCT by DMKFMT cannot be defined by the area-specifications of SYSPAG macros. If a SYSPAG macro attempts to define these areas, message DMKVDG340I is issued.

- Cylinders allocated as DUMP by DMKFMT can only be mapped by SYSPAG macros as TYPE=DU. If the SYSPAG type is other than DU, message DMKVDG340I is issued.

- Cylinders defined by a SYSPAG area-specification should be within the same DMKFMT area allocation. This prevents the generation of more than one ALOCBLOK for the SYSPAG area-specification. For example, if a SYSPAG area-specification spans a DMKFMT allocation type boundary (PAGE,TEMP), two ALOCBLOKs will be generated: one for the area allocated as PAGE by DMKFMT and one for the TEMP area.

- Cylinders that will be mapped by SYSPAG macros as TYPE=PS must be allocated as TEMP by DMKFMT and should be on a separate volume(s).

- Pages allocated for FBA devices should be on pseudo-cylinder boundaries. If this is not done, pages outside of cylinder boundaries, but within allocation extents, will not be used when mapped by SYSPAG macros. For a description of pseudo-cylinders for FBA devices, see "FB-512 Device Geometry" on page 36.

Page Migration

How Page Migration Works

As the preferred paging (PP) areas fill, pages are migrated to a general paging (PG) area. As the general paging areas fill, pages are migrated to a page migration (PM) area. The least frequently referenced pages are moved first. You can specify how full an area must be before migration takes place using the SET SRM PGFULL command. (See the *VM/SP HPO System Programmer's Guide* for more information.)

Swap sets from Paging Storage are migrated to swap areas on lower levels. If no space is available on lower SW levels, or if there are no lower SW areas, the swap sets are broken up. The pages are migrated to the first demand page level without Paging Storage on which space is available. Figure 39 shows migration down the paging hierarchy.

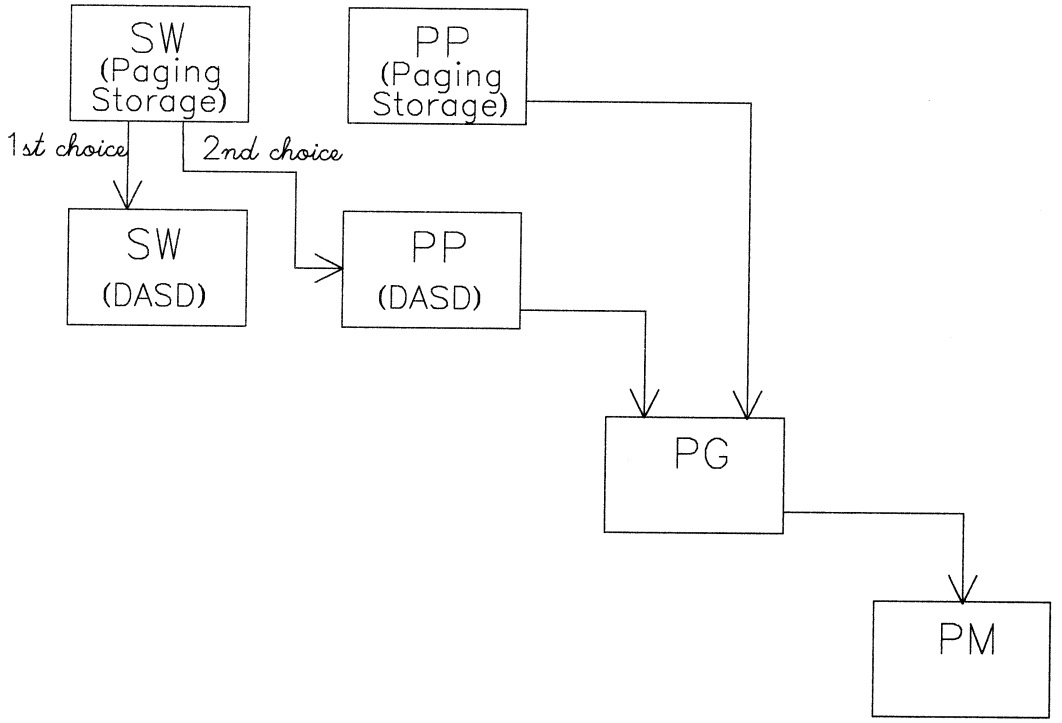


Figure 39. How Pages and Swap Sets Are Migrated Down the Paging Hierarchy

In Release 4.0 and earlier releases, pages were migrated directly to a PM area. The new method of migration is more efficient as long as the following conditions hold:

- Most of the system I/O activity is done to preferred devices.
- There is significantly more DASD space in the nonpreferred areas.

What Is an Effective Paging Hierarchy?

An effective paging hierarchy uses preferred devices more often than nonpreferred devices. If the system load is small, all paging activity should be to preferred devices or swap devices. As the system load gets larger, some paging to nonpreferred devices must take place, but preferred devices should still be used for most of the system paging activity.

Pages are now migrated down the paging hierarchy instead of directly to a PM area. Systems using the old method of migration needed enough PM space to catch the overflow from the PP and PG areas, although most I/O was done to PP areas. Now you should allocate more PG space than PP space and more PM space than PG space. You should also use the QUERY SRM PGMACT command to verify that more I/O is done to the preferred devices. If this is not so, use the SET SRM PGFULL and SET SRM PGMTLIMIT commands to adjust the I/O activity. (See the *VM/SP HPO Operator's Guide* and the *VM/SP HPO System Programmer's Guide* for information on these commands.)

You can tell how effective your paging hierarchy is by measuring the percentage of I/O activity on each storage level. Use the QUERY SRM PGMACT command to get this information. Figure 40 and Figure 41 show some acceptable and unacceptable distributions of I/O activity. In both figures, it is assumed that the PP areas are in Paging Storage, PG areas on cached DASD, and PM areas on 3350s or 3380s.

Level	Good Percent of I/O	Better Percent of I/O	Best Percent of I/O
SW, PP	80%	90%	100%
PG	15%	10%	0%
PM	5%	0%	0%

Figure 40. I/O Distribution across Good Paging Hierarchies

Level	Percent of I/0
SW, PP	80%
PG	5%
PM	15%

Figure 41. I/O Distribution across a Bad Paging Hierarchy

In Figure 41 on page 321, the PG areas are filling up too quickly. You can do two things to correct this situation:

- Allocate more PG space
- Move the devices on the PG level to the PP level, keeping your new SYSPAG macros of TYPE=PP after the old TYPE=PP macros. If you do this, the former PG devices are still used after the devices allocated on earlier SYSPAG macros of TYPE=PP, but pages are be migrated directly from PP areas to PM areas, since there is now no PG space. In this case, one-step migration is more efficient than two-step migration.

For more information on using the QUERY SRM PGMACT command to analyze the performance of your paging hierarchy, see the *VM/SP HPO System Programmer's Guide.*

Establishing the Access Structure

The number and order of SYSPAG macros specified in DMKSYS establishes an access structure for the CP-owned volumes and Paging Storage. Each SYSPAG macro specified creates a SYSPLIST control block. The SYSPLIST control block contains the area-specification information and serves as the anchor for the ALOCBLOKs that will be built at system IPL or when a CP-owned volume is attached to the system. It is through the SYSPLIST control blocks that an allocation hierarchy is created by providing a leveled search order through the ALOCBLOKs for page allocation. An overview of the generation and relationship of SYSPLIST control blocks and ALOCBLOKs is shown in Figure 42.

SYSPLIST Defaults

Any area ALOCBLOKs not covered by a SYSPAG area-specification are placed on a default level SYSPLIST. The allocation type (defined by DMKFMT) for the cylinders and the device type determine what SYSPLIST chain to use. These area ALOCBLOKs are chained as follows:

- Area ALOCBLOKs defined as PAGE for a 2305 or 3380-11 DASD device are chained from the last SYSPLIST control block for TYPE=PP areas.
- Area ALOCBLOKs defined as PAGE which are not for a 2305 or 3380-11 DASD device are chained from the last SYSPLIST control block for TYPE=PG areas.
- Area ALOCBLOKs defined as TEMP are chained from the last SYSPLIST control block for TYPE=PS areas.
- Area ALOCBLOKs defined as DUMP are chained from the last SYSPLIST control block for TYPE=DU areas.

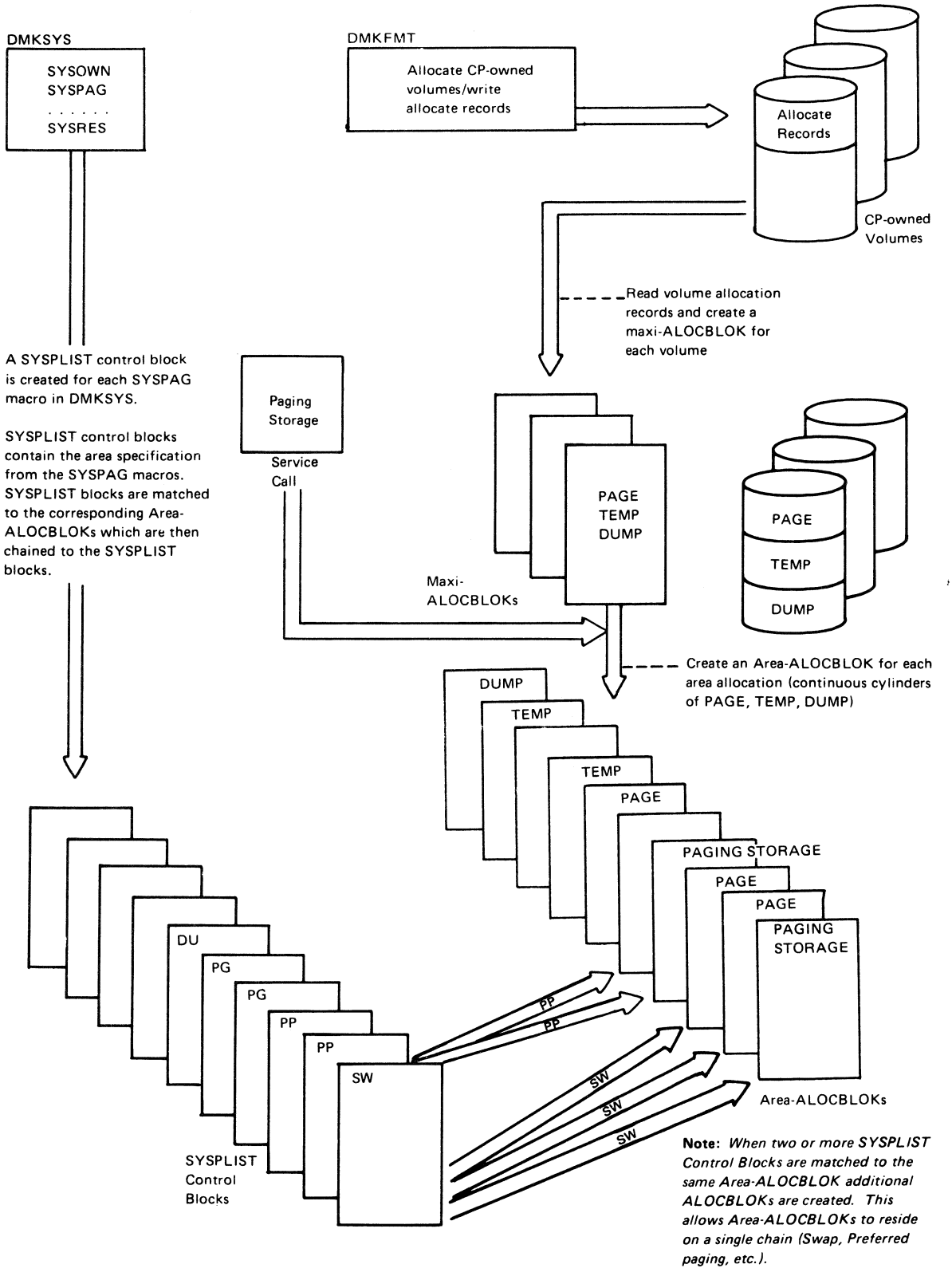


Figure 42. SYSPLIST/ALOCBLOK Generation and Relationship

SYSPAG Coding, SYSPLIST/ALOCBLOK Examples

The following example illustrates the use of the SYSPAG macro to control page and spool allocation.

```
SYSPAG VMSWP1,VMSWP2,VMSWP3,VMSWP4,TYPE=SW,SWSIZE=8
SYSPAG (PAGESTORE),TYPE=PP
SYSPAG (VM1D01,1,2),(VM1D02,1,2),                          X
      (VM1D03,1,2),(VM1D04,1,2),                           X
      TYPE=PP
SYSPAG (VM4PG3,394,413),TYPE=PG
SYSPAG (VM4PG4,384,393),                                   X
      (VM4MD7,0,413),                                      X
      TYPE=PG
SYSPAG (VM4PG4,,383),TYPE=PM
SYSPAG (VM4PG3,424,),TYPE=PM
SYSPAG (VM4SP1,371,530),TYPE=PS
SYSPAG (VM4MD7,414,807),TYPE=DU
```

Swap allocation is distributed across the CP-owned volumes VMSWP1, VMSWP2, VMSWP3, and VMSWP4. Paging allocation is distributed across Paging Storage and the CP-owned volumes VM1D01, VM1D02, VM1D03, and VM1D04. Paging Storage will be used before the CP-owned volumes. When these areas are full, swapping occurs to the paging areas. Multiple, but not necessarily consecutive, pages are moved out/in. Each page, for as many pages as there are in the swap set, is separately handled and causes a PCI interrupt.

Figure 43 shows a sample SYSPLIST/ALOCBLOK chaining structure for the above SYSPAG macros. It shows the relationship between the control blocks and the allocation levels.

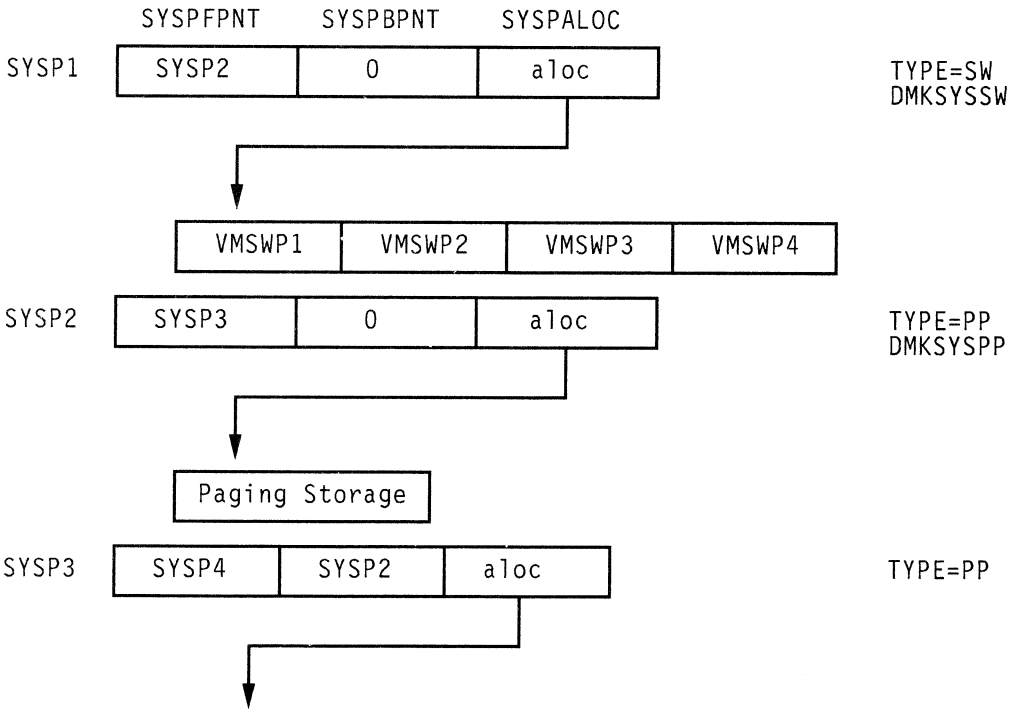


Figure 43 (Part 1 of 3). Sample SYSPLIST/ALOCBLOK Chaining Structure

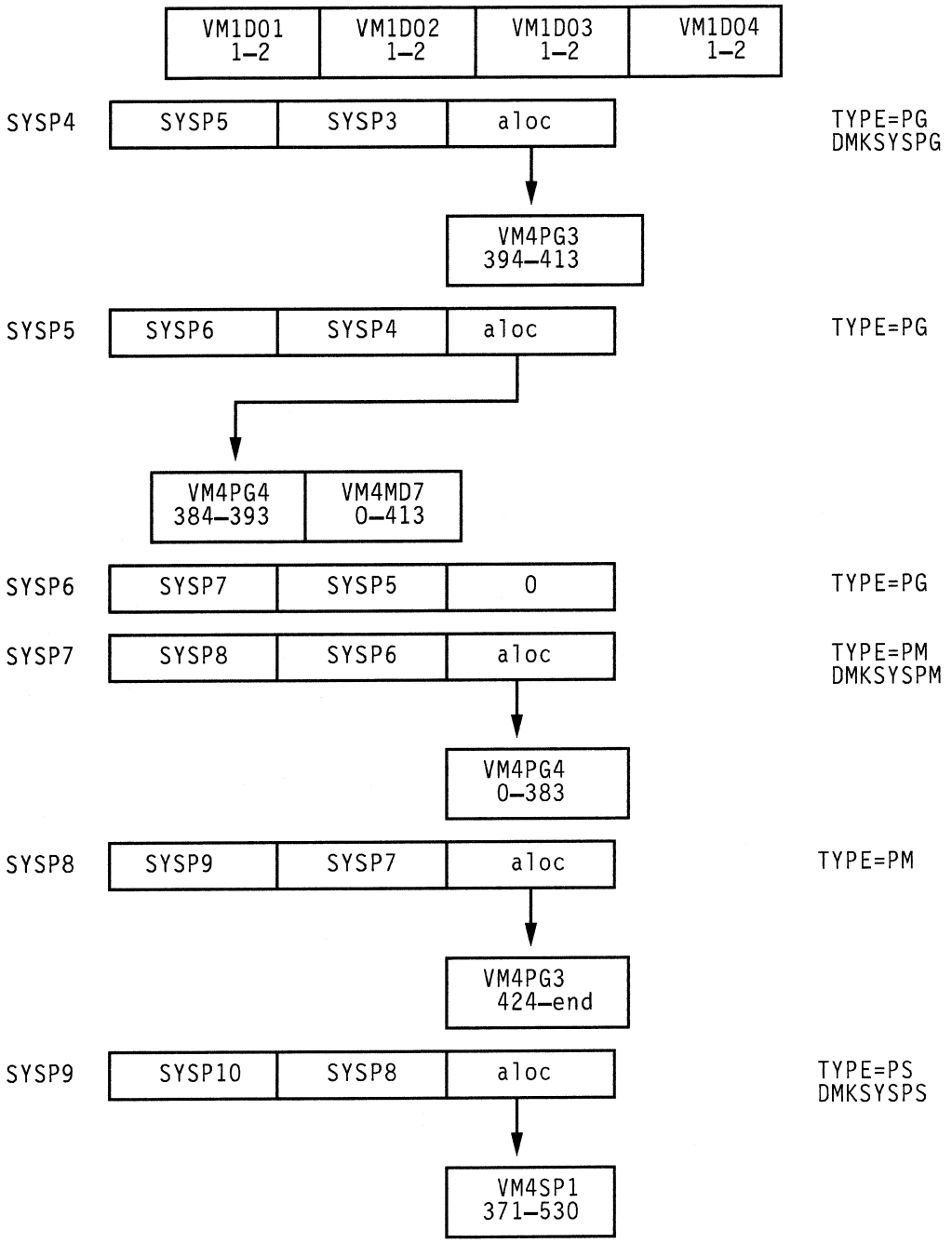


Figure 43 (Part 2 of 3). Sample SYSPLIST/ALOCBLOK Chaining Structure

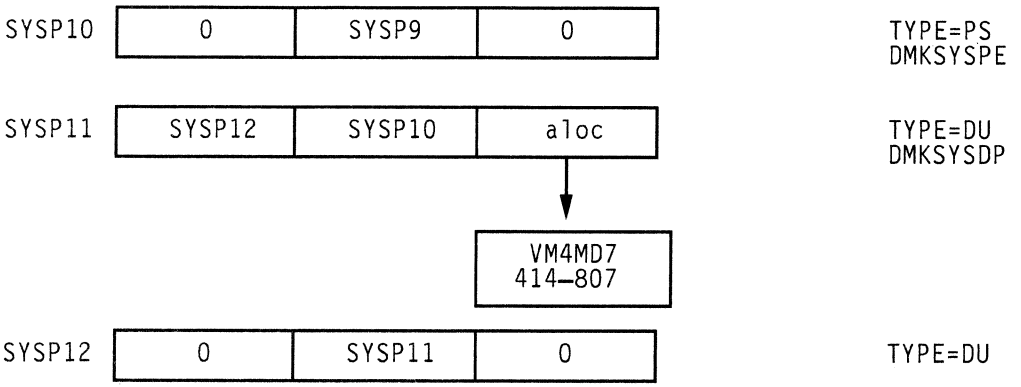


Figure 43 (Part 3 of 3). Sample SYSPLIST/ALOCBLOK Chaining Structure

Error Messages Generated by the SYSPAG Macro

The following MNOTE message is generated by the SYSPAG macro and has a severity code of 12:

type OUT OF ORDER
Indicates that the SYSPAG macro is out of order. See TYPE=xx description above for explanation.

The following MNOTE messages are generated by the SYSPAG macro and all have a severity code of 8:

INVALID TYPE type
Indicates that the operand of the TYPE= option is not SW, PP, PG, PM, PS, or DU. Reenter the SYSPAG macro specifying a valid TYPE.

volid NOT IN SYSOWN LIST
Indicates that the volid specified has not been included in a SYSOWN definition.

NO SYSOWN VOLUMES DEFINED
Indicates that no SYSOWN macro has been specified or that the SYSPAG macro has been placed before the SYSOWN macro.

NULL ENTRY NOT VALID FOR TYPE type
Only a TYPE=DU SYSPAG macro without any volume specifications is valid.

(volid,cyl1,cyl2) CYLINDERS ALREADY DEFINED;
volid start_cyl end_cyl
The specified cylinders have already been defined. The specified volume-cylinder operands are ignored.

MORE THAN 255 AREAS SPECIFIED
The limit of 255 area definitions has been exceeded.

MULTIPLE AREAS SPECIFIED AT SAME LEVEL
Two or more areas have been specified on the same SYSPAG macro for the same CP-owned volume. This defeats the allocation algorithms. The specified volume-area operands are ignored. Specify these areas on separate SYSPAG macros.

PAGING STORAGE INCREMENTS i1, i2 ARE REJECTED; ALREADY DEFINED AS s1,s2
The specified increments of Paging Storage, i1 to i2, overlap areas of Paging Storage that have already been allocated. They are ignored.

PAGING STORAGE CANNOT BE DEFINED FOR TYPE type
An area of Paging Storage has been defined as a type other than SW or PP. This area specification is ignored.

PAGING STORAGE CAN ONLY BE DEFINED ON FIRST TYPE=SW MACRO
Paging Storage was defined on a SYSPAG macro of TYPE=SW other than the first. The swap area will not be allocated.

The following MNOTE messages are generated by the SYSPAG macro and all have a severity code of 4:

INVALID CYLINDER SPECIFIED FOR (volid,cyl1,cyl2)
The starting cylinder is larger than the ending cylinder or the ending cylinder was specified as larger than 32767 or more than two cylinder operands were specified. If the starting or ending cylinder specification was invalid, the cylinder is set to the default value (0 for starting cylinder and end-of-volume for the ending cylinder). If more than two cylinder operands were specified, the starting and ending cylinders are set to the default values.

NULL ENTRY PREVIOUSLY DEFINED
Only one SYSPAG macro (TYPE=DU) without any volume specifications is allowed. Since the SYSRES macro effectively specifies such a "null" macro, you should not code a "null" SYSPAG macro in the DMKSYS deck. This 'null' SYSPAG macro is used to create the SYSPLIST control block structure when the user has not included any SYSPAG macros in module DMKSYS.

INVALID INCREMENT NUMBER SPECIFIED FOR PAGING STORAGE (pagestore, incre1, incre2); DEFAULT VALUE USED
One of these three errors has been made in the specification of a Paging Storage area:

- The starting increment number is greater than the ending increment number. The starting increment number is set to the default value, 1.
- The ending increment number is greater than 64. The ending increment number is set to the default value, which is the number of the last increment attached to the processor.

- More than two increment numbers are specified. The starting increment number is set to 1 and the ending increment number is set to the number of the last increment attached to the processor.

SWSIZE OPTION INVALID

The SWSIZE option has been specified on the wrong SYSPAG macro. SWSIZE, if specified, must be on the first SYSPAG macro of TYPE=SW. The specification is ignored and SWSIZE is set to either the default value or to a value previously (and validly) specified.

INVALID SWSIZE SPECIFIED

The value specified for SWSIZE is less than 1 or greater than 99. SWSIZE is set to the default value, 8.

SYSRES Macro

Use the SYSRES macro instruction to describe characteristics of the CP system residence volume.

The name field must not be specified for the SYSRES macro instruction.

Special Considerations for Coding the SYSRES Macro: The following information should help you when you code the SYSRES macro:

- All operands must be specified with appropriate values.
- Areas required for SYSNUC, SYSERR, SYSWRM, and SYSCKP must be formatted using the CP Format/Allocate service program, and must be allocated as permanent space on the SYSRES volume, but not in cylinder 0.
- VM/SP HPO allows 2314 or 2319 "alternate tracks" cylinders 200-202 to be used for normal data if they are not needed to replace defective tracks.
- On a 3340, "alternate tracks" cylinders cannot be used for normal data. On a 3340 Model 35, use only cylinders 0-347. On a 3340 Model 70, use only cylinders 0-695.
- An MSS 3330V volume may not be used as the CP SYSRES volume.
- If the CP nucleus is generated on a virtual machine, at least one of the SYSRES address operands (address or altaddr) must be a three-digit hexadecimal number between 000 and FFF.

The format of the SYSRES macro is:

Name	Operation	Operands
	SYSRES	SYSVOL=serial ,SYSRES={address | (address [,altaddr])} ,SYSTYPE={2305 | 2319 | 3310 | 3330 | 3340 | 3350 | 3370 | 3375 | 3380 | FB-512} ,SYSNUC={strtcyl | strtpage} ,SYSCLR={YES | NO} ,SYSERR=[({strtcyl [,cylcount | ,2] | strtpage [,pagecount | ,100]})] ,SYSCKP=[({strtcyl [,cylcount | ,1] | strtpage [,pagecount | ,50]})] ,SYSWRM=[({strtcyl [,cylcount | ,1] | strtpage [,pagecount | ,50]})]

where:

SYSVOL = serial
is the volume identification of the system residence disk. 'serial' is a string of up to six characters.

SYSRES = { address | (address [,altaddr]) }

address
designates a three- or four- digit hexadecimal device address (ccuu) between 000 and 1FFF for the DASD volume to contain the newly generated system.

altaddr
identifies an alternate address for writing the CP nucleus. For multiprocessing systems, specification of an alternate address facilitates native execution of CP module DMKSAV on either processor. DMKSAV writes the newly generated CP nucleus on the IPL volume.

If both address and altaddr are specified, parentheses are required. If the CP nucleus is generated on a virtual machine, address, altaddr, or both must be a three-digit hexadecimal number between 000 and FFF.

Note: Executed on a virtual machine, the addresses are virtual addresses; on a real machine (stand-alone), they are real addresses.

When the CP nucleus is written from tape to the system residence volume, the following steps occur after the CP nucleus tape is IPLed.

1. DMKLD00E (the loader) passes control to DMKSAV.
2. DMKSAV uses the SYSRES operands of the SYSRES macro in DMKSYS to find the DASD address on which to write the CP nucleus:
   a. The address operand is sought first on the processor where the tape was IPLed. If the DASD at address is found, the CP nucleus is written to address.
   b. If the DASD specified by address is not found, the PSW is loaded with a wait state of X'00000010'. To recover from the wait state, attach the system residence volume to the IPLed processor as either address or altaddr. Then, execute an external interrupt to DMKSAV to write the CP nucleus to this device.

SYSTYPE = { **2305** | **2319** | **3310** | **3330** | **3340** | **3350** | **3370** | **3375** | **3380** | **FB-512** }

is the device type of the system residence device.

For a 3350 device in native mode, specify 3350 as the device type. For a 3350 being used in 3330 compatibility mode, specify 3330. Specify a 3344 disk as a 3340, and a 3333 as a 3330. 2305 applies to both 2305-1 and 2305-2. For a 3380 device, specify 3380 as the device type.

SYSNUC = { **strtcyl** | **strtpage** },

is the number of the real starting cylinder where the CP nucleus resides (strtcyl) or the starting page number of the CP nucleus for FB-512 (strtpage).

strtcyl
is a one- to four-digit decimal number.

strtpage
is a one- to six-digit decimal number.

Normally,

- A 2305 or 3340 device requires ten contiguous cylinders
- A 3330/3333 device requires eight contiguous cylinders
- A 3350 device requires three cylinders
- A 3375 device requires four cylinders
- A 3380 device requires three cylinders
- An FB-512 requires approximately 220 pages

to contain the CP nucleus.

SYSCLR = {**YES** / **NO**}

specifies automatic clearing of all previously written data and directory areas residing on T-DISK DASD space. This prevents other users from accidentally accessing these areas. If YES is specified, T-DISK DASD space is cleared to binary zeros at CP initialization and when attaching a CP-owned volume containing T-DISK allocations. T-DISK space is also cleared when a user detaches a T-DISK. Specifying SYSCLR = YES provides additional security for your VM/SP HPO installation. If you specify NO, the system will clear cylinder 0, track 0 on T-DISK DASD.

The SYSCLR option is required. Specifying SYSCLR with invalid or misspelled options produces the following MNOTE

```
INVALID SYSCLR OPTION
```

and the specification defaults to YES.

SYSERR = **[(** {**strtcyl** [**,cylcount**] [**,2**] / **strtpage** [**,pagecount**] [**,100**]} **)]**

is the number of the real starting cylinder (strtcyl), or the starting page number for FB-512 (strtpage), where error records are written. Optionally, it is the number of cylinders required for error recording (cylcount) or the number of FB-512 pages (pagecount).

strtcyl
is a one- to four-digit decimal number designating the starting cylinder of the error recording area.

cylcount
is a one-digit decimal number between 2 and 9 designating the number of cylinders.

strtpage
is a one- to six-digit decimal number designating the starting page of the error recording area.

pagecount
is a one- to six-digit decimal number.

SYSCKP = [(] { **strtcyl** [**,cylcount**] [**,1**] / **strtpage** [**,pagecount**] [**,50**] } [)]

is the number of the real starting cylinder (strtcyl), or the starting page number of the checkpoint area for FB-512 (strtpage). It is optionally the maximum number of cylinders to contain the dynamic checkpoint start data (cylcount), or the number of FB-512 pages to be reserved (pagecount). The default for pagecount is 50.

The checkpoint space available determines how many spool file IDs may be assigned. The maximum number of unique spool file IDs is the lesser of either the 9900 system limit or the number of checkpoint slots allocated.

strtcyl
is a one- to four-digit decimal number designating the first real cylinder where CP checkpoint start information is to be saved.

cylcount
is a two-digit decimal number (1 through 20) that defines the maximum number of cylinders to contain checkpoint start data. If cylcount is not specified, 1 is the default value. For example: one cylinder of a 3330 will allow slightly less than 2000 unique spool file IDs.

strtpage
is a one- to six-digit decimal number designating the starting page of the checkpoint area.

pagecount
is a one- to six-digit decimal number.

The cylcount/pagecount operand is optional. If included, the strtcyl/strtpage and cylcount/pagecount operands must be separated by a comma and enclosed in parentheses. Parentheses are optional when only the strtcyl operand is specified.

The number of cylinders or pages required for the checkpoint start data is dependent upon the device type. They are as follows:

Device Type	Minimum Recommended No. of Cylinders/Pages	Cylinders/Pages for 9900 Spool File IDs
2305	3 cylinders	14 cylinders
2314	2 cylinders	11 cylinders
2319	2 cylinders	11 cylinders
3330	1 cylinder	6 cylinders
3340	3 cylinders	14 cylinders

Device Type	Minimum Recommended No. of Cylinders/Pages	Cylinders/Pages for 9900 Spool File IDs
3350	1 cylinder	4 cylinders
3375	1 cylinder	5 cylinders
3380	1 cylinder	3 cylinders
FB-512	50 pages	298 pages

Use the formulas shown in Figure 44 to calculate space needed for the dynamic checkpoint start area. The formula for each device type is:

Device Type	Formula
2314/2319	$N = \frac{[34 + (NSP/32) + (NRS/34)]}{32}$
2305/3340	$N = \frac{[26 + (NSP/32) + (NRS/34)]}{24}$
3330	$N = \frac{[59 + (NSP/32) + (NRS/34)]}{57}$
3350	$N = \frac{[122 + (NSP/32) + (NRS/34)]}{120}$
3375	$N = \frac{[98 + (NSP/32) + (NRS/34)]}{96}$
3380	$N = \frac{[152 + (NSP/32) + (NRS/34)]}{150}$
FB-512	N = [2 + (NSP/32) + (NRS/34)]

where:

N is the number of cylinders or pages required for checkpoint start data.

NSP is the number of spool files to be checkpointed. There are 32 entries per 4096-byte record.

NRS is the number of real spooling devices defined in DMKRIO.

Note: When using the formulas above for count-key-data DASD, disregard any remainder in the answer.

Figure 44. Dynamic Checkpoint Start Area Calculations

$$\textbf{SYSWRM} = [(] \left\{ \begin{array}{l} \textbf{strtcyl} \left[\begin{array}{l} \textbf{,cylcount} \\ \textbf{,}\underline{\textbf{1}} \end{array} \right] \\ \textbf{strtpage} \left[\begin{array}{l} \textbf{,pagecount} \\ \textbf{,}\underline{\textbf{50}} \end{array} \right] \end{array} \right\} [)]$$

is the number of the real starting cylinder (strtcyl), or the starting page number of the warm start area for FB-512 (strtpage). It is optionally the maximum number of cylinders, to contain the warm start data (cylcount) or the number of FB-512 pages to be reserved (pagecount).

strtcyl
is a one- to four-digit decimal number designating the first real cylinder where CP warm start information is to be saved.

cylcount
is a two-digit decimal number (1 through 20) that defines the maximum number of cylinders to contain warm start data. If cylcount is not specified, one is the default value.

strtpage
is a one- to six-digit decimal number.

pagecount
is a one- to six-digit decimal number.

The cylcount/pagecount operand is optional. If included, the strtcyl/srtpage and cylcount/pagecount operands must be separated by a comma and enclosed in parentheses. Parentheses are optional when only the strtcyl/strtpage operand is specified.

Use the formulas in Figure 45 on page 336 to calculate the number of warm start cylinders or pages required. When you use the formulas, disregard all remainders. For example, for a 3330 system residence volume plus

- A maximum of 34 spool files in the system at one time
- A maximum of 128 cylinders available for spool files
- A maximum of 50 active users at one time

the calculation is

$$N = \frac{[59 + 32/32 + 128/128 + 200/50]}{57} = \frac{65}{57} = 1$$

Example

The following SYSRES macro defines the system residence volume as the 3380 volume with a serial number of VMSRES. During the system generation procedure this volume is found at address 230. The VM/SP HPO system starts at cylinder 882 and the error recording area starts at cylinder 879. The warm start storage area is cylinder 887 and the checkpoint start storage area is cylinder 442. The format of the SYSRES macro is:

```
SYSRES SYSVOL=VMSRES,SYSRES=230,SYSTYPE=3380,SYSNUC=882,    X
       SYSCLR=,SYSERR=(879,2),SYSWRM=(887,2),SYSCKP=(442,1)
```

The formula for each device type is shown in Figure 45.

Device Type	Formula
2314/2319	$N = \frac{[34 + (NSF/32) + (NCS/128) + ((NAU \times 4)/50)]}{32}$
3340/2305	$N = \frac{[26 + (NSF/32) + (NCS/128) + ((NAU \times 4)/50)]}{24}$
3330	$N = \frac{[59 + (NSF/32) + (NCS/128) + ((NAU \times 4)/50)]}{57}$
3350	$N = \frac{[122 + (NSF/32) + (NCS/128) + ((NAU \times 4)/50)]}{120}$
3375	$N = \frac{[98 + (NSF/32) + (NCS/128) + ((NAU \times 4)/150)]}{96}$
3380	$N = \frac{[152 + (NSF/32) + (NCS/128) + ((NAU \times 4)/50)]}{150}$
FB-512	$N = [2 + (NSF/32) + (NPS/85) + ((NAU \times 4)/50)]$

where:

N	is the number of cylinders or pages required for warm start data.
NSF	is the maximum number of spool files in the system at any one time. There are 32 spool file blocks per 4096-byte record.
NCS	is the number of cylinders available for spool files. There are 128 allocation blocks per 4096-byte record.
NPS	is the number of pages available for spool files on FB-512 devices. There are 85 FB-512 allocation blocks per 4096-byte record.
NAU	is the maximum number of active users in the system at any one time. There are 50 accounting records per 4096-byte record.

Figure 45. Warm Start Area Calculations

SYSMON Macro

The SYSMON macro is used to invoke daily automatic performance data collection with the IBM Field Developed Program Virtual Machine Facility/370: Performance/Monitor Analysis Program. This program is equipped with a front end assembly language routine that contains the appropriate diagnose commands to read the file and perform data reduction.

The format of the SYSMON macro is:

Name	Operation	Operands
	SYSMON	[USERID=OPERATOR] [USERID=userid] [,CLASS=M] [,CLASS=class] [,AUTO=NO] [,AUTO=YES] [,ENABLE=(PERFORM,USER,DASTAP)] [,ENABLE=(classa,classb,classc,...)] [,TIME=(09:00,17:00)] [,TIME=(h1:m1,h2:m2)] [,TIME=ALL] [,TIME=NONE] [,LIMIT=(50000,NOSTOP)] [,LIMIT=(limit,STOP)] [,LIMIT=(limit,NOSTOP)] [,LIMIT=(limit,SAMPLE)] [,BUFFS=cpu] [,BUFFS=nn]

where:

[USERID=OPERATOR]
[USERID=userid]

is the userid of the virtual machine that will receive the monitor spool file in its virtual reader. The default is OPERATOR but any valid system directory entry may be specified.

[CLASS=M]
[CLASS=class]

specifies the spool file to be generated to contain monitor data. Any valid class (A through Z and 0 through 9) may be used; but the default, M, is preferred, since the VMMAP data reduction Field Developed Program is designed to reduce only spool files of that class.

[AUTO=NO]
[AUTO=YES]

specifies whether or not automatic monitoring should take place according to remaining SYSMON parameter specifications. The default, NO, requires the installation to make a specific change to cause automatic monitoring. All other parameters may be system system default values, giving positive and useful monitoring results.

ENABLE = (PERFORM,USER,DASTAP)
ENABLE = (classa,classb,classc,...)

specifies any combination of valid monitor classes of data collection. It is assumed that the system analyst understands the use of various classes, overhead incurred in data collection, and relative magnitude of the corresponding data reduction. The default specifies sampled data classes only and are considered minimal for useful data reduction. The default classes are sufficient for analysis of a system's load environment and performance profile with a view to diagnosis of possible bottlenecks and for establishing long term growth trends. See the description of the MONITOR command in the *VM/SP HPO System Programmer's Guide* for information about classes, overhead, and recommendations.

TIME = (09:00,17:00)
TIME = (h1:m1,h2:m2)
TIME = ALL
TIME = NONE

specifies the time period in each day that automatic monitoring (performance data collection) should take place. This parameter may indicate a start and stop time in hours and minutes using a 24-hour clock. Continuous monitoring (if ALL is specified), or no monitoring (if NONE is specified) unless the operator or system analyst overrides this specification with the MONITOR command. If a system restart occurs during an automatic monitoring period, the old spool file is closed out and a new one is started, according to the SYSMON specifications. For useful data reduction, several hours of monitoring are suggested.

Note: The same closeout occurs at midnight if ALL is specified.

LIMIT = (50000,NOSTOP)
LIMIT = (limit,STOP)
LIMIT = (limit,NOSTOP)
LIMIT = (limit,SAMPLE)

specifies the maximum number of monitor record buffers that can be added to the monitor spool file before it is closed, whether or not monitoring should be terminated when the limit is reached or the periodic closing of the monitor spool file after a specified number of samples (also defined by the value of LIMIT) have been collected. This parameter gives the installation more control over the amount of spool space that can be used by the automatic monitoring facility. It can also be used to create several small monitor spool files, rather than one large file, and give the data reduction facility an opportunity to start processing the morning's data while collecting the afternoon's data. 'limit' can be any decimal number between 10 and 50000. When determining the value for 'limit', take into consideration the classes of data collection enabled, the size of the associated records, and the sampling interval. Remember that each monitor buffer contains approximately 4000 bytes of data space.

Specifying SAMPLE allows the installation analyst to define the rate at which spool files will be produced. Since sampled data is collected at very precise intervals of time, according to the value specified in the MONITOR INTERVAL command (default 60 seconds), the spool file may be consistently and repeatedly closed. Monitor spool files obtained in this manner contain performance data covering consecutive, and equal intervals of time. This data contains the same number of PERFORM, DASTAP, and, possibly, USER (if no users logged on or off) records. This capability could form the basis of a real time performance analysis facility.

[BUFFS=cpu]
[BUFFS=nn]

specifies the number of data collection buffers needed by the monitor to avoid suspension occurrences. Data collection suspension occurs when output to tape or spool files cannot keep ahead of data collection and an overflow condition occurs. By increasing the number of monitor buffers, suspension occurrences can be reduced or eliminated. (See the *VM/SP HPO Operator's Guide* for a description of the MONITOR command.) Buffs=nn can specify from 1 to 99 buffers. The default depends on the real storage size of the processor on which the system is running:

Storage Size	Default Number of Buffers
Less than 1 M	2 buffers
Between 1 and 2 M	3 buffers
Between 2 and 4 M	4 buffers
Between 4 and 8 M	5 buffers
Between 8 and 16 M	6 buffers
Between 16 and 32 M	7 buffers
Over 32 M	8 buffers

One additional buffer is added on an AP or MP system.

As monitoring the 3880 Storage Subsystem adds bytes to the monitor records, users of this system should consider increasing the number of monitor buffers if suspensions occur.

Example

```
SYSMON USERID=ANALYST,AUTO=YES,ENABLE=(PERFORM),        X
       TIME=ALL,BUFFS=1
```

This example specifies automatic monitoring for 24 hours a day using only the PERFORM class of data collection and one buffer. The spool file created is practically unlimited in size, taking the 50000 default and will be sent to the ANALYST virtual machine's reader each midnight, at system restart, or shutdown. M is the default spool file class.

Note: All of the above automatic monitoring specifications may be overridden by the operator or system analyst using the MONITOR command.

SYSJRL Macro

The SYSJRL macro is used to specify the inclusion of the journalizing and/or password suppression facility.

The format of the SYSJRL macro is:

Name	Operation	Operands
	SYSJRL	[,JOURNAL=NO ,JOURNAL=YES] [,STQUERY=NO ,STQUERY=YES] [,LOGUID=OPERATOR ,LOGUID=userid] [,LOGLOC=(10,60) ,LOGLOC=(n,m)] [,LOGLMT=(2,3,4) ,LOGLMT=(x,y,z)] [,LNKUID=OPERATOR ,LNKUID=userid] [,LNKLMT=(2,5,10) ,LNKLMT=(x,y,z)] [,PSUPRS=NO ,PSUPRS=YES]

where:

[JOURNAL=NO / JOURNAL=YES]
indicates whether or not the journalizing facility is to be operative in the system being generated.

[STQUERY=NO / STQUERY=YES]
indicates whether or not the ability to SET and QUERY the journalizing function should be a part of the system being generated. STQUERY=YES may be specified only if JOURNAL=YES is also specified.

[LOGUID=OPERATOR / LOGUID=userid]
is the userid that should receive the indication that an invalid logon password count has been reached or exceeded. If this userid is disconnected or logged off, the operator will receive the message generated.

[LOGLOC=(10,60) / LOGLOC=(n,m)]
is the logon inductor. n is the maximum number of invalid password attempts. The value of n may range from 1 to 255. The default is 10 logon attempts.

m is the delay time in minutes until the next logon attempt. The value of m may range from 0 to 255. The default is 60 minutes.

If anyone makes more than n unsuccessful attempts to log on to a given userid, an error message is issued. Nobody will be able to log on to that userid, or at the terminal where the attempts were made, until m expires. Anyone who tries will receive an error message.

LOGLMT = (2,3,4)
LOGLMT = (x,y,z)

is the invalid LOGON/AUTOLOG password threshold specification. The value specified applies to all unsuccessful logon attempts at a single terminal. x is the value which, when reached or exceeded, causes a type 04 accounting record to be generated for that and each subsequent LOGON/AUTOLOG containing an invalid password. y is the value which, when reached or exceeded, causes a message to be sent to the userid specified by LOGUID for that and each subsequent LOGON/AUTOLOG containing an invalid password. z is the value which, when reached, causes the LOGON/AUTOLOG command to be disabled. A new VM logon screen will be displayed and a new logon sequence can be started.

Note: z replaces the present fixed limit of 4 and may be any decimal number from 1 to 255. x and y may be any decimal number from 0 to 255. 0 is a special case that indicates the applicable function should be bypassed. For example, if LOGLMT=(0,5,5) is specified, no accounting records are generated.

LNKUID = OPERATOR
LNKUID = userid

is the userid that should receive the indication that an invalid link password count has been reached or exceeded. If this userid is disconnected or logged off, the operator will receive the message generated.

LNKLMT = (2,5,10)
LNKLMT = (x,y,z)

is the invalid LINK password threshold specification. The value specified applies to a single userid for a single LOGON session. x is the value that, when reached or exceeded, causes a 06 accounting record to be generated for that and each subsequent LINK containing an invalid password. y is the value that, when reached or exceeded, causes a message to be sent to the userid specified by LNKUID for that and each subsequent LINK containing an invalid password. z is the value that, when reached, causes the LINK command to be disabled for the current LOGON session.

Note: z replaces the former fixed limit of 10 and may be any decimal digit from 1 to 255. x and y may be any decimal digit from 0 to 255. 0 is a special case which indicates the applicable function to be bypassed. For example, if LNKLMT=(2,0,10) is specified, no message is sent.

[**PSUPRS=NO**
PSUPRS=YES]

indicates whether or not the facility that suppresses the password on the command line should be part of the system being generated.

Note: *If PSUPRS=YES is specified, for a 2741 terminal, passwords will be typed upon a mask. Specifying PSUPRS=YES provides additional security for your VM/SP installation.*

SYSCOR Macro

Use the SYSCOR macro instruction to generate the internal control block called the CORTABLE. The AP and MP operands specify whether CP will try to make use of an additional processor.

The name field must not be specified for the SYSCOR macro instruction.

The format of the SYSCOR macro is:

Name	Operation	Operands
	SYSCOR	RMSIZE={xxxxxK / yyM} [,RSSIZE={xxxxxK / yyM}] [,FREE=ffff] [,PRIME=nnn] [,TRACE=nnn] [,AP={YES / NO}] [,MP={YES / NO}]

where:

RMSIZE = {xxxxxK / yyM}

is the amount of real storage available for VM/SP HPO. This value limits the amount of real storage used by VM/SP HPO if the value is less than the total amount of real storage available in the real machine. If the available real storage is less than this value when VM/SP HPO is initialized, a message indicating the amount of storage available is displayed at the operator's console.

xxxxx

is a four- to five-digit number that denotes the amount of real storage in terms of K bytes, where 1K = 1024 bytes. This value may range from 1024K to 65536K. It must always be a multiple of 2.

yy

is a one- or two-digit number that denotes the amount of storage in megabytes. One megabyte is equal to 1024 K bytes. The minimum value for yy is 1M, the maximum, 64M.

Notes:

1. *Do not specify a value substantially larger than the size of real storage, because the generated core table uses a large amount of real storage. Each additional megabyte of storage defined in the SYSCOR macro will generate 4K bytes of additional CORTABLE entries.*

2. *Preferred Machine Assist and Preferred Machine Assist with Control Switch Assist Support allow an MVS/SP release 3 or later preferred virtual machine to access storage up to 64 megabytes.*

RSSIZE = {xxxxxK / yyM}

is the amount of real storage the CORTABLE will be built for in processors using extended storage support. This is the amount of storage that will be available to CP if the V=R area is unlocked. RSSIZE is optional. If not specified, it will default to the value of RMSIZE. For storage maps illustrating extended storage support, see the *VM/SP HPO System Programmer's Guide.*

xxxxx
is a four- or five- digit number that denotes the amount of real storage in terms of K bytes, where 1K=1024 bytes. This value may range from 16384K to 65536K. It must always be a multiple of 2.

yy
is a one- or two- digit number that denotes the amount of real storage in terms of M bytes, where 1M=1024K bytes. This value may range from 16M to 64M.

If RSSIZE is specified, it must be greater than or equal to RMSIZE. If either operand is incorrectly specified, the following MNOTES will be generated:

```
MNOTE: 4, RSSIZE NOT GREATER THAN 16 MEGABYTES, IGNORED
```

when RSSIZE is not equal to RMSIZE and RSSIZE is less than 16 megabytes.

```
MNOTE: 4, STORAGE FROM RMSIZE TO 16 M UNUSABLE
```

when RSSIZE is greater than RMSIZE and RMSIZE is less than 16 megabytes.

```
MNOTE: 8, INVALID RSSIZE SPECIFIED
```

when the RSSIZE specified exceeds 64 megabytes or is less than RMSIZE.

FREE=ffff
is a one- to four-digit number that specifies the number of fixed free storage pages to be allocated at system initialization. This number must be three or greater.

Note: In previous releases of VM/SP HPO VMBLOK was taken out of free storage. In Release 3.4 and later releases VMBLOK is taken out of DPA.

The recommended value for ffff is one page for each virtual machine logged on, based on the average number of virtual machine users.

If FREE is not specified, CP allocates three pages for the first 256K of real storage and one page for each additional 64K up to 16 megabytes, not including the V=R size, if any. Above 16 megabytes, CP allocates 1 page for each 256K of real storage. In AP and MP modes, the default is increased by 25%.

One method of determining if you need to alter the value of FREE (after your initial sysgen) is to XEDIT your CPNUC MAP and locate the label "DMKFRENP". Write down the hex location of DMKFRENP, get out of XEDIT, and issue the command:

```
DCP hexloc.8
```

The first four bytes are the number of extends (in hex). The last four bytes are the number of disextends. If the extends are greater than the disextends, you should increase FREE by the difference. If there are no extends, you may decrease the value of FREE.

PRIME=nnn
is the decimal number (nnn) of pages to be allocated for PRIME subpools. If no PRIME is required, a value of zero should be specified. Because PRIME storage is a subset of FREE storage, the number (nnn) specified for PRIME must be smaller than the number specified for FREE. If the number is not smaller, an MNOTE is generated.

The default for the PRIME operand is 10% of the FREE pages. If the number of FREE pages is less than 10, then PRIME will default to 0 pages. If the value specified for FREE or TRACE is too large, a warning message is issued to the operator's console. This message states that the size of FREE, TRACE, and PRIME have been reduced by the same percentage in order to satisfy the minimum DPA size requirement. A more detailed analysis of FREE and PRIME usage is possible by using perform class monitor data.

AP= { **YES** / **NO** }

YES
specifies that processing is in attached processor mode if the attached processor is available at system IPL.

NO
specifies that processing is in uniprocessor mode regardless of the presence of an attached processor.

MP= { **YES** / **NO** }

YES
specifies that initialization processing occur in multiprocessor mode.

NO
specifies that initialization processing occur in uniprocessor mode regardless of the presence of a second processor.

MP=YES and AP=YES parameters are mutually exclusive. A system generated for MP execution will execute on an MP, AP, or UP system. However, it will not function with maximum efficiency for AP or UP modes of operation. If you code both AP=YES and MP=YES, the following MNOTE is issued:

```
MP=YES AND AP=YES BOTH SPECIFIED; MP=YES ASSUMED
```

Note: An additional 25% of free storage is allocated in AP or MP mode. (See FREE=.)

TRACE=nnn
is the decimal number of 4K pages to be used for the trace table. If the number of pages specified on the TRACE operand is not larger than the default trace table size provided by the system (one page for each 256K of real storage up to 16Mb, and 1 page per megabyte above 16 megabytes), the default size will be used for the trace table.

Example

This example defines real storage as 1M (1,048,576 bytes):

```
SYSCOR   RMSIZE=1M
```

This example defines real storage as 256K (262,144 bytes):

```
SYSCOR   RMSIZE=256K
```

SYSOPR Macro

Use the SYSOPR macro instruction to specify the system operator's userid, and the userid of the operator who is to receive CP system dumps. The same userid may be specified in both operands.

The name field must not be specified for the SYSOPR macro instruction.

The format of the SYSOPR macro is:

Name	Operation	Operands
	SYSOPR	[SYSOPER=OPERATOR] [SYSOPER=userid] [,SYSDUMP=OPERATNS] [,SYSDUMP=userid]

where:

[SYSOPER=OPERATOR]
[SYSOPER=userid]
is the userid of the virtual machine to be assigned to the system operator. If SYSOPER is not specified, the userid OPERATOR is used.

The userid is a character string up to eight characters long.

[SYSDUMP=OPERATNS]
[SYSDUMP=userid]
is the userid of the virtual machine whose spool input receives the system dump file after a system restart. This userid also receives guest virtual machine dumps produced by CP VMDUMP, if you specify the destination as SYSTEM. If SYSDUMP is not specified, the userid OPERATNS is used. If you intend to use IPCS, allow this operand to default to OPERATNS or specify the IPCS userid.

The userid is a string of up to eight characters.

Example

The following SYSOPR macro designates the OP virtual machine as the system operator and directs the system dumps to the CPSYS virtual machine.

```
SYSOPR SYSOPER=OP,SYSDUMP=CPSYS
```

SYSACNT Macro

SYSACNT is a required system generation macro instruction used to specify spooling of accounting records. The format of the SYSACNT macro is:

Name	Operation	Operands
label	SYSACNT	[USERID={OPERATOR / userid} ,OUTPUT={PUNCH / READER} ,CLASS={C / class} ,LIMIT={0 / nnnnn}]

where:

label
is any desired label.

USERID = {OPERATOR / userid}
is the virtual machine identification to which all files are spooled. OPERATOR is the default userid.

OUTPUT = {PUNCH / READER}
is the type of spool file to be created. PUNCH is the default type.

CLASS = {C / class}
is the class of the spool file to be created. A-Z and 0-9 are valid. Class C is the default.

LIMIT = {0 / nnnnn}
is the number of records to be accumulated before the file is closed and made available to the virtual machine. nnnnn is up to 5 decimal digits from 0 to 32,767. 0, the default, indicates that the file is not to be closed automatically.

Accounting records are accumulated in a spool file identified by the USERID and CLASS parameters until either the number of records specified in the LIMIT parameter is reached or until the ACNT command is issued with the CLOSE operand. At that time, the records are sent to the virtual punch or reader, as specified in the OUTPUT parameter.

If accounting records are stored in reader files in the user's virtual machine, files should be processed periodically by the user and not allowed to accumulate and tie up system resources.

Note: The accounting records can be moved to tape via the SPTAPE command for later processing.

SYSTIME Macro

Use the SYSTIME macro instruction to generate information needed to set the hardware time of day (TOD) clock. The value stored in the TOD clock represents time taken at Greenwich Mean Time, and must be corrected to local time whenever it is examined. The system operator can alter the defined time value by using the store clock function.

The name field must not be specified for the SYSTIME macro instruction.

The format of the SYSTIME macro is:

Name	Operation	Operands
	SYSTIME	[ZONE=0 ZONE=h ZONE=(h,m) ZONE=(h,m,s) ZONE=(h,,s)] [,LOC=EAST ,LOC=WEST] [,ID=GMT ,ID=xxx]

where:

[ZONE=0
ZONE=h
ZONE=(h,m)
ZONE=(h,m,s)
ZONE=(h,,s)]

is the time zone differential from Greenwich Mean Time. If ZONE is not specified, a value of 0 hours (Greenwich Mean Time) is used.

The variable h is a number that represents hours. It can have a value from 0 to 13, but when coupled with the m and s fields, the total effective zone differential must not exceed 13 hours.

The variable m is a number that represents minutes.

The variable s is a number that represents seconds.

[LOC=EAST
LOC=WEST]

specifies whether the time zone differential is to be taken EAST or WEST of Greenwich Mean Time. The default value for LOC is EAST. When the effective value of ZONE is 0, the setting of LOC is meaningless.

[**ID=GMT**]
[**ID=xxx**]

is the name of the time zone. The default for ID is GMT. The variable xxx is a three-character string.

Examples

The following examples show how to code the SYSTIME macro for several different time zones:

```
SYSTIME ZONE=5,LOC=WEST,ID=EST  (Eastern Standard Time)
SYSTIME ZONE=4,LOC=WEST,ID=EDT  (Eastern Daylight Time)
SYSTIME ZONE=6,LOC=WEST,ID=CST  (Central Standard Time)
SYSTIME ZONE=7,LOC=WEST,ID=MST  (Mountain Standard Time)
SYSTIME ZONE=1,LOC=EAST,ID=SET  (Standard European Time)
SYSTIME ZONE=1,LOC=EAST,ID=BST  (British Summer Time)
SYSTIME ZONE=10,LOC=EAST,ID=EST (Australian Eastern
                                Standard Time)
```

SYSFORM Macro

Use the SYSFORM macro instruction to specify:

- A list of user forms with their corresponding operator forms. Operator forms can also be specified as NARROW, so that a narrow (94-character) separator page is printed.
- Default user forms for printer, punch, and console. (The default operator forms are obtained from the list of user/operator forms.) These defaults apply:
  - When creating a virtual printer, punch, or console spool file, unless you have overridden the default with a SPOOL or CLOSE command.
  - When the operator issues a START command for a printer or punch without specifying a form, and this is the first START command since cold starting VM/SP HPO.

The format of the SYSFORM macro is:

Name	Operation	Operands
	SYSFORM	[(userform,operform [,NARROW]) ...] [,DEFPRT=userform] [,DEFPUN=userform] [,DEFCON=userform]

where:

userform
specifies the one- to eight-character user form name. Use this form in CP SPOOL, CLOSE, CHANGE, PURGE, ORDER, QUERY, and TRANSFER commands.

operform
specifies the one- to eight-character name for the corresponding operator form.

If no userform or operform pairs are specified, no form list is generated. In this case, no distinction is made between user forms and operator forms.

NARROW
specifies printing on narrow (94-character) paper. Separator information will not print beyond this width.

DEFPRT
specifies the default user (and implicitly, operator) forms when creating virtual printer spool files, or when starting the real printer.

If DEFPRT is not specified, the default is DEFPRT=STANDARD.

DEFPUN
specifies the default user (and implicitly, operator) forms when creating virtual punch spool files, or when starting the real punch.

If DEFPUN is not specified, the default is DEFPUN=STANDARD.

DEFCON
specifies the default user (and implicitly, operator) forms when creating virtual console spool files.

If DEFCON is not specified, the default is DEFCON=STANDARD.

Examples

1. Default case.

```
SYSFORM
```

No user/operator form list, therefore no distinction between user forms and operator forms.

Default forms are STANDARD for printer, punch, and console.

2. Defaults specified.

```
SYSFORM DEFPRT=VANILLA,DEFPUN=WHITE
```

No user/operator form list, therefore no distinction between user forms and operator forms.

Default forms are VANILLA for the printer, WHITE for the punch, and STANDARD for the console.

3. User/operator form list.

```
SYSFORM (LISTING,QN-8-14),                          X
        (DOCUMENT,TN-6-8,NARROW),                   X
        (OUTPUT,PN-6-14),                           X
        DEFPRT=OUTPUT,DEFCON=DOCUMENT
```

User form LISTING corresponds to operator form QN-8-14.

You issue the command SPOOL PRINTER FORM LISTING, and the operator sees form QN-8-14. Likewise, user form DOCUMENT corresponds to the narrow operator form TN-6-8, and OUTPUT corresponds to PN-6-14. All other forms have the user form equal to the operator form.

OUTPUT/PN-6-14 is the default form for the printer.

DOCUMENT/TN-6-8 is the default form for console spool files.

STANDARD is the default for the punch.

SYSPCLAS Macro

Use the SYSPCLAS macro instruction to classify printed output with a classification title. This title is printed on the output separator page, and optionally at the top or bottom of each page of output.

The installation can specify a different classification title for each output class (A-Z and 0-9), providing up to 36 different classifications.

The macro specifies a list of class/title pairs. The TOP or BOTTOM option can be specified for any pair. The TOP option specifies that this title is to be printed at the top of all pages of output, not just the separator page. The BOTTOM option prints the title at the bottom of all output pages.

The format of the SYSPCLAS macro is:

Name	Operation	Operands
	SYSPCLAS	[(c, 'title' [,TOP / ,BOTTOM]) ...]

where:

c
is a one-character spool file class. It must be alphanumeric.

title
is a classification title for this class. It may be up to 46 characters long.

The title may contain any characters. It must be enclosed in quotes. According to rules of assembler language, quotes within the title are coded as two consecutive quotes, and ampersands are coded as two consecutive ampersands.

TOP
specifies that this title is to be printed both on the separator page, and at the top of each page of output.

BOTTOM
specifies that this title be printed both on the separator page, and at the bottom of each page of output.

See "Usage Notes" for more information.

Examples

1. No classifications desired.

   ```
   SYSPCLAS
   ```

   No classification titles are generated.

2. Classifications generated.

   ```
   SYSPCLAS (R,'CUSTOMER CONFIDENTIAL'),                X
            (N,'COMPANY NAME',TOP)
   ```

   Class R output is printed with CUSTOMER CONFIDENTIAL on the separator page. Class N output has COMPANY NAME on the separator page, and at the top of each page of output.

Usage Notes:

1. The SYSPCLAS macro replaces previous support for class X files in module DMKBOX.

2. The title line is inserted at the top or bottom of each page. To do this, CP inserts a CCW to print one space, followed by the title line before or after each "skip to channel 1" CCW issued by the guest operating system. There are several implications to this:

   a. The titles are inserted as the file is being created. If the file is later changed to a different class, the titles in the file are not changed. However, the title on the separator page always reflects the true class of the file.

   b. Inserting the title on the output page adds a line of output to the printed page. If the guest application is counting lines, its count will be incorrect. This can result in alternating output and blank pages.

   c. If the guest doesn't use "skip to channel 1", the title lines are never printed.

   d. If the guest doesn't issue "skip to channel 1" at the end of the file, and the BOTTOM option is being used, the last page will not have a classification title on it. Usage of the TOP option circumvents this situation.

SYSID Macro

The SYSID macro statement allows an installation to identify a system with an eight-character identification, called the sytem ID. The system ID appears in the status area of the display terminal and is printed on the output separator page. The processor ID is specified with a model number and a serial number. If the installation has multiple VM/SP HPO processors, a list of system ID's with corresponding processor model and serial numbers can be specified. When VM/SP HPO is initialized via IPL, the initialized processor is matched with an entry in the list of system ID's, thus identifying the local system ID. The processor ID of the IPLed processor is obtained with the STORE CPUID instruction. If no match is found, a default system ID is assumed.

The format of the SYSID macro is:

Name	Operation	Operands
	SYSID	[(systemid,model, serial),...] [,DEFAULT=defid]

where:

systemid
specifies the one- to eight-character system ID that identifies a uniprocessor system.

model
designates a three- or four-digit number that specifies the processor model number (for example 158, 3031, or 4341) for a multiple processor installation.

serial
is a five- or six-digit number that specifies the processor serial number corresponding with the system ID. AP/MP installations use the serial and model number of both processors in the SYSID macro.

DEFAULT=defid
designates the one- to eight-character default system ID. This ID is selected if the processor is not found in the list, or if there is no list.

Examples

1. Default case.

   ```
   SYSID
   ```

   No system ID is printed on the separator page.

2. Single processor installation.

   ```
   SYSID DEFAULT=CUSTOMER
   ```

Gives this VM/SP HPO system the ID "CUSTOMER". This ID will be printed on the separator page.

3. Multiple processor installation.

```
SYSID (CUSTSYS1,158,13289),                              X
      (CUSTSYS2,4341,23145),DEFAULT=OTHERSYS
```

If this system is IPLed on the 370/158 serial number 13289, the system ID is CUSTSYS1. If it is IPLed on the 4341 serial number 23145, the system ID is CUSTSYS2. If it is IPLed on any other system, the system ID is OTHERSYS.

SYSMIH Macro

Use the SYSMIH macro instruction to define the time interval desired for missing interruption monitoring. I/O activity is monitored for the following devices:

- Direct access storage devices
- Graphic devices
- Tape devices
- Unit record devices
- Miscellaneous devices

If you specify zero for a device group, monitoring is set off for that group.

Do not specify the name field for the SYSMIH macro instruction.

The format of the SYSMIH macro instruction is:

Name	Operation	Operands
	SYSMIH	[DASD={0:15 | mm:ss}] [,GRAF={0:30 | mm:ss}] [,TAPE={10:00 | mm:ss}] [,UR={1:00 | mm:ss}] [,MISC={12:00 | mm:ss}]

where:

DASD = {0:15 | mm:ss}

specifies the time interval value for count-key data direct access storage devices (CLASDASD) and fixed block architecture devices (CLASFBA). You can specify a maximum value of 99 for mm (minutes) and a maximum value of 59 for ss (seconds). If you do not specify a value for this class, the time interval is set to the default value, 15 seconds.

GRAF = {0:30 | mm:ss}

specifies the time interval value for graphic devices (CLASGRAF). You can specify a maximum value of 99 for mm (minutes) and a maximum value of 59 for ss (seconds). If you do not specify a value for this class, the time interval is set to the default value, 30 seconds.

Note: 328x printers, formerly classified as graphic devices, are now classed as miscellaneous devices.

TAPE = {**10:00** / **mm:ss**}

specifies the time interval value for tape devices (CLASTAPE). You can specify a maximum value of 99 for mm (minutes) and a maximum value of 59 for ss (seconds). If you do not specify a value for this class, the time interval is set to the default value, ten minutes.

UR = {**1:00** / **mm:ss**}

specifies the time interval value for unit record input devices (CLASURI) and unit record output devices (CLASURO). You can specify a maximum value of 99 for mm (minutes) and a maximum value of 59 for ss (seconds). If you do not specify a value for this class, the time interval is set to the default value, one minute.

Note: 3800 and 3289 printers, formerly classified as unit record devices, are now classed as miscellaneous devices.

MISC = {**12:00** / **mm:ss**}

specifies the time interval value for miscellaneous devices. Miscellaneous devices include MSS devices, Clasgraf TYP3228A and TYP1053, and Clasuro TYP3800 and TYP3289E printers. MSS devices include Classpec TYP3851 and Clasdasd Feature = Sysvirt or Feature = Virtual. You can specify a maximum value of 99 for mm(minute) and a maximum value of 59 for SS(second). If you do not specify a value for this class, the time interval is set to the default value, 12 minutes.

Notes:

1. *If you do not specify a value for a device class, CP uses the default time interval.*
2. *If you specify zero for a device group, monitoring is set off for that group. Specify zero for any device group that you do not use.*
3. *Use the SET MITIME command to change the time intervals of device classes. Use the QUERY MITIME command to determine the time intervals in effect.*
4. *If you specify a time interval for a device class below its default value, be careful not to shorten the time interval too much since this may cause unnecessary missing interruption handler prcessing for devices that are functioning properly.*
5. *If you specify more than one time interval for a device class, the last value coded is used.*
6. *If you remove module DMKDID from the load list, and later issue the SET MITIME or QUERY MITIME command, CP issues a message that missing interruption monitoring is not available.*

7. *If you specify an invalid time value in the SYSMIH statement, the time value is set to the default, return code 4 is generated, and the following MNOTE is issued:*

   ```
   INVALID TIME VALUE SPECIFIED FOR class -TIME SET TO time
   ```

   Here, class *indicates the device class that has the invalid time value and time indicates the default value for this class.*

8. *Users of the 3800 printer should specify a minimum of 5 minutes in the MISC operand, to accommodate printer warm-up.*

Example

This example illustrates the use of the SYSMIH macro instruction to:

- Set a 15-second time interval for direct access storage devices
- Set a 20-second time interval for graphic devices
- Disable I/O monitoring for tape devices
- Set a one minute time interval for unit record miscellaneous devices
- Disable I/O monitoring for Mass Storage System devices

```
SYSMIH GRAF=00:20,TAPE=00:00,MISC=00:00
```

SYSFCN Macro

The SYSFCN macro is used to change user privilege classes. Both IBM-defined or installation-defined classes having access to restricted internal control program functions can be altered.

Whenever you redefine user privilege classes, be sure to change the SYSFCN macro if necessary.

The format of the SYSFCN macro is:

Name	Operation	Operands
	SYSFCN	[PRIV={ABCDEF / cl}] [,OPER={A / cl}] [,CPRD={CE / cl}] [,CPWT={C / cl}] [,SERV={F / cl}] [,DFLT={G / cl}]

where:

PRIV = {ABCDEF / cl}

specifies the user privilege classes that are authorized to access a dedicated tape drive. The default value is ABCDEF.

OPER = {A / cl}

specifies the user privilege classes that are authorized to LOGON during initialization. The default value is A.

CPRD = {CE / cl}

specifies the user privilege classes that are authorized to issue IOCP READ instructions. The default value is CE.

CPWT = {C / cl}

specifies the user privilege classes that are authorized to issue IOCP WRITE instructions. The default value is C.

SERV = {F / cl}

specifies the user privilege classes that are authorized to issue Diagnostic Load/Write and Sense/Read commands. The default value is F.

DFLT = {G / cl}

specifies the user privilege classes that will be assigned to a user by default if neither the class field in the USER control statement nor the CLASS control statement is coded. The default value is G.

cl

is, in all cases, a string of 1 to 32 characters identifying user privilege classes. Each character identifies a single class. Valid class identification characters are the letters A to Z and the digits 1 to 6. The characters may be entered in any order, without duplication, and must not be separated by blanks.

SYSLOCS Macro

The SYSLOCS macro instruction is a required macro used to generate internal pointer variables. This must be the last macro in the DMKSYS deck.

The name field must not be specified for the SYSLOCS macro instruction. No operands are required for the SYSLOCS macro. If one is specified, it is ignored.

The format of the SYSLOCS macro is:

Name	Operation	Operands
	SYSLOCS	

The following MNOTE message is generated by the SYSLOCS macro and has a severity code of 4:

```
===> ERROR.  THE SYSORD MACRO SHOULD NOT BE SPECIFIED. <===
```

The functions of the SYSORD macro are now performed by the SYSPAG macro. SYSORD is no longer a valid specification in module DMKSYS.

Chapter 24. Additional System Definition Files

Altering the Forms Control Buffer Load (DMKFCB)

The DMKFCB module is supplied with the starter system. This module defines a 3211, 3203-4, 3203-5, 3262-1, 3262-5, 3262-11, 3263-1, 3289 Model 4, 4245, or 4248-1 Printer forms control buffer image. There are two names provided for an FCB image.

FCB1 controls printing at 6 lines per inch, 66 lines per page, and has the following channel skip specifications:

Line Represented	Channel Skip Specification
1	1
3	2
5	3
7	4
9	5
11	6
13	7
15	8
19	10
21	11
23	12
64	9

FCB8 controls printing at 8 lines per inch, 68 lines per page, and has the following channel skip specifications:

Line Represented	Channel Skip Specification
1	1
4	2
8	3
12	4
16	5
20	6
24	7
28	8
32	10
36	11
63	12
66	9

If you wish to alter the supplied buffer load, see the *VM/SP HPO System Programmer's Guide* for directions.

Altering the Universal Character Set or Font Offset Buffer

The DMKUCS, DMKUCB, DMKUCC, DMKPIA, and DMKPIB modules are supplied with the starter system. These modules correspond to the following printer types:

Printer Type	Module Name
1403	DMKUCS
3211	DMKUCB
3203	DMKUCC
3289	DMKPIA
3262	DMKPIB

If you wish to change the supplied buffer load, see the module prologue.

Chapter 25. Defining Saved Systems and Discontiguous Saved Segments

Saved systems are systems that have been saved on a disk along with all the information you need to resume execution. Saved systems provide an efficient means of IPLing systems by bypassing many system initialization steps.

Discontiguous Saved Segments (DCSS) are areas of virtual storage outside the address range of a virtual machine. These segments can contain read only data or reentrant code that many users can share. Discontiguous saved segments provide an efficient means of fetching programs by merely connecting discontiguous segments to a virtual machine.

Shared Segments are segments within a saved system or DCSS. These segments can contain read only data or reentrant code that many users can share. This reduces the demand for real storage for the overall system.

Saved Systems

Saved systems provide an efficient means of IPLing systems by bypassing many system initialization steps. A saved system contains a copy of virtual machine storage, register contents, PSW, and storage keys. The system is not loaded at IPL time. Instead, the system initializes page tables, and brings pages into storage as needed during execution.

Since a saved system can share segments of reenterable code, real storage is used more efficiently.

Note: You can IPL a saved system in a virtual=real machine only in nonshared mode.

Discontiguous Saved Segments

Discontiguous Saved Segments (DCSS) are areas of virtual storage outside the address range of a virtual machine that:

- Have a name associated with them
- Contain read only data or reenterable code
- Were previously loaded and saved
- Can be shared by multiple users
- Can be loaded by a particular virtual machine in nonshared mode for testing and debugging

Discontiguous Saved Segments provide an efficient means of fetching programs by connecting discontiguous segments to a virtual machine.

Discontiguous saved segment support allows a virtual machine to logically attach and detach segments of storage. Many users can share these segments.

The load address for the discontiguous saved segment should be slightly higher than the highest storage address of any virtual machine using it. The saved segment should not overlay any other named segment that may be attached at the same time.

Installing the Newly Defined DCSS or Saved System

1. Be sure that the volume containing the segments is CP owned (listed in the SYSOWN macro for the DMKSYS file).

2. Reserve the allocated DASD space with a dummy minidisk in the $SAVSYS$ userid of the CP directory. Run the CP Format/Allocate program to format the cylinders and to allocate them as PERM.

   At this point you can use the SNTMAP EXEC to determine if the defined $SAVSYS$ areas contain gaps, which can be used for new or expanded DCSS DASD areas. See "SNTMAP EXEC" on page 370 for a description of this exec.

3. Update the DMKSNT assemble file with the appropriate NAMESYS, NAMENCP, or NAME3800 macro and operands.

   The NAMESYS macro creates an entry in the system name table for a virtual machine or saved segment. The NAMENCP macro creates an entry in the system name table for a 3704/3705 control program. The NAME3800 macro creates and entry in the system name table for a 3800 named system. Refer to Chapter 22, "Preparing the System Name Table File (DMKSNT)" on page 301 for more details about these macros.

4. Assemble the DMKSNT file.

   If you get a clean assembly, you should run the SNTMAP EXEC again to verify there are no DASD overlays in your macro specifications.

5. Build and test the new CP nucleus which contains the new definition.

6. Load the information to be saved into the proper location in virtual memory.

7. Save the information in the defined DASD area by issuing the CP SAVESYS command.

   For information on saving IBM supplied saved systems and DCSSs, refer to the *VM/SP HPO Installation Guide*.

Considerations

The following might cause errors while saving a DCSS:

- Formatting DASD improperly. This can be corrected by re-running the CP Format/Allocate program to format the cylinders to receive the DCSS.
- A mismatch between the SYSPGNM specification and the load address used when saving the system.
- Failure to account for the "extra" CP information pages when allocating DASD space for the DCSS.
- Forgetting to save the system before attempting to use it (some software will attempt to use the DCSS as soon as it is defined).
- Programming errors in the software saved into the DCSS.

Shared Segments

Shared segments are segments within a saved system or a DCSS that users can share. They contain read only data and reentrant code. If you designate one or more segments of a saved system as "shared", a virtual machine that loads the saved system by name can use these segments in real storage. This reduces the real storage demand for the overall system.

A segment of storage is 64K bytes long on a 64K byte boundary.

Notes:

1. *A shared segment must be reenterable.*
2. *You must include the segment number in the SYSHRSG operand of the NAMESYS macro for the saved system.*
3. *You cannot use shared segments in a virtual=real machine.*
4. *The maximum number of segments you can define in a saved system or DCSS is 78.*
5. *You must be careful when you resave a saved system that contains one or more shared segments.*

   If the previous system is loaded by name and is still in use, problems can occur. If you use the "old" system and continue to reference pages already in paging storage, then there is no problem. However, if you use the old system and reference pages not previously referenced (in the newly saved system), then you receive the **new** *versions of pages you reference.*
6. *When you IPL the newly saved system, you share only the new copy of the shared segment.*
7. *Using the SAVESYS command saves the entire segment, not just that portion the program occupies (for example, CMS). Thus, if there is any unwanted data it is contained in that segment.*

SNTMAP EXEC

Introduction

The SNTMAP EXEC, or "SNTMAP", processes SNT macro definitions and produces two CMS files. One file, DASD SNTMAP, is a saved segment DASD map, and the other, MEMORY SNTMAP, is a virtual memory map.

Using SNTMAP, you can group NAMESYS macros into "families". SNTMAP places each family into a separate memory map. You can then check for overlapping segments. This is particularly useful for GCS environments which do not allow segments to overlap within a GCS group.

Input to SNTMAP

User Input

The DMKSNT source file is the main input from the user. It contains the NAMESYS, NAMENCP, and NAME3800 macros. SNTMAP assumes that you have assembled the file and found no syntax errors.

Note: If you want to group NAMESYS macros into families, you must insert comment lines into the DMKSNT file. This is explained in more detail later. *See "Grouping Macros into Families" on page 371.*

The CP Directory file is another user input to SNTMAP. The directory contains a USER $SAVSYS$ entry that describes the DASD areas allocated for SNT data. After you invoke SNTMAP, the exec prompts you to enter the name of the directory. If SNTMAP cannot find a USER $SAVSYS$ entry or if one or more MDISK statements define the DASD type as FB-512, SNTMAP will prompt you to enter the DASD types. In this case, SNTMAP will not check for $SAVSYS$ extent exceeded conditions.

Other Input

SNTMAP also uses the file $DASD$ CONSTS, which is supplied with VM/SP. This file contains information about DASD that SNTMAP needs for calculations and conversions. SNTMAP cannot do any processing without this file.

Output from SNTMAP

SNTMAP creates two CMS files that you can edit (using XEDIT,) print, and display at a terminal. When the two output files are created, any old files by the same name are erased. Samples of these files are shown in Appendix F, "Sample SNTMAP Output" on page 441.

The output files are:

1. DASD SNTMAP

   This file contains a list of the DCSS names and DASD allocations for each pack that has DCSS resources defined. The list is arranged in order of occurrence on a given pack. Any gaps are noted within the list and summarized in a table at the end of the file.

2. MEMORY SNTMAP

   This file is a list of DCSS names and segment allocations arranged in order by memory location. Overlapping segments are noted within the list. If you have grouped NAMESYS macros into families, this file will show a separate memory map for each family. This file maps only the shared segments defined by a NAMESYS macro. Unshared pages are listed at the end of the file.

How to Use SNTMAP

Grouping Macros into Families

To make a NAMESYS macro a member of a family, insert a comment line in the DMKSNT file just before the NAMESYS macro. The format of the comment line is:

```
*  Family = f(f(...))
```

"f" is any single character representing a particular family. Members of the same family will use the same character. You can include a macro in more than one family. If, for example, you want to include a macro in families "A," "C," and "7," insert this line just before the NAMESYS macro:

```
*  Family = AC7
```

As a result of adding this comment line, the macro will appear in the main memory map along with all the other NAMESYS macros, and it will appear in the memory maps for families A, C, and 7.

Invoking the Exec

To invoke SNTMAP, enter the following:

```
SNTMAP [fn [ft [fm]]]
```

where fn, ft, and fm refer to the file name, file type, and file mode, respectively, of the DMKSNT file. The file name must be entered. If you don't enter the file type or file mode, SNTMAP will assume the file type is "ASSEMBLE" and the file mode is "*".

If you want help invoking SNTMAP, enter `SNTMAP ?` or `SNTMAP HELP`.

Prompts and Messages

After you invoke SNTMAP, the exec may ask you for more information.

1. If you did not supply a valid file name for the DMKSNT file, SNTMAP will ask for the file name with the following prompt:

   Enter name of SNT definition file or press ENTER key to quit.

2. SNTMAP will ask for the name of the CP directory file with this message:

   Enter the name of your current CP directory file. Default is VMUSERS DIRECT.

3. If one of the following is true:

   - SNTMAP cannot find a CP directory
   - There is no USER $SAVSYS$ entry in the CP directory
   - MDISK statements in the directory define the DASD type as FB-512

   SNTMAP will prompt you to enter the DASD types. In this case, SNTMAP will not check for $SAVSYS$ extent exceeded errors.

4. When SNTMAP has finished running, it will give you the following message:

   ```
   Results of mapping are in two CMS files:
       DASD related information is in DASD SNTMAP
       Memory related information is in MEMORY SNTMAP
   ```

Errors

SNTMAP assigns a return code of zero if there were no errors in processing. A non-zero return code means there were errors. SNTMAP issues messages for the following errors:

- SNTMAP can't find the $DASD$ CONSTS file
- There is no read/write disk accessed as an "A disk"
- You entered an invalid DASD type
- The page number on the SYSSTRT parameter exceeds the limit for the device type
- The SYSSTRT parameter information is not compatible with the DASD type.

For a detailed explanation of each message, see *VM/SP HPO System Messages and Codes*.

Security Considerations

Consider the following when planning to install saved systems and discontiguous saved segments:

- The SAVESYS command (used to write information into the DCSS) is a privileged CP command—class E. Privilege classes provide a means for restricting access to CP commands. Thus, giving Class E privilege to product installers gives them access to all Class E commands and the ability to save any DCSS defined in the system.
- If PROTECT = NO is specified on the NAMESYS macro, users of a DCSS may find their address space corrupted by others.

Appendixes

A. Program Products, Installed User Programs, Field Developed Programs, and Emulators

B. Configuration Aid

C. Compatible Devices

D. VM/SP HPO Restrictions

E. Worksheet to Aid in Coding the NAMESYS Macro

F. Sample SNTMAP Output

Appendix A. Program Products, Installed User Programs, Field-Developed Programs, and Emulators

VM/370 Assembler

The VM/370 Assembler is distributed as a part of the VM/SP HPO system and is required for installation and further support of the system. All necessary installation and support macros are provided in CMS libraries.

The Conversational Monitor System (CMS), and the Control Program (CP) are distributed as components of VM/SP HPO. Certain other facilities mentioned in this publication are not part of VM/SP HPO, but can be separately ordered from IBM. These include: IBM System/360 and System/370 operating systems, IBM language processors and other IBM program products, IBM Installed User Programs, and IBM Field-Developed Programs. For more information, contact your IBM representative.

Program Products

The following is a list of IBM program products and their respective program numbers that VM/SP HPO users have found useful.

Program products are listed first in each area, followed by FDP/IUP/PRPQs (Field-Developed Programs, Installed User Programs, Programming Requests for Price Quotation).

General Business Applications:

Interactive Instructional Authoring System (IIAS)	5668-001
Interactive Instructional Presentation System (IIPS)	5668-012
Alpha Search Inquiry System (ASIS)	5736-N14
Planning Control and Decision Evaluation (PLANCODE/I)	5740-XX8
A Departmental Reporting System II (ADRS-II)	5796-PLN
APL Data Interface - Version II (APL/DI-II)	5796-PNG
Alpha Search Inquiry System Online Update	5798-CFJ
Braille Utilities	5798-CRZ
VS APL Financial Planning System II (FPS-II)	5798-DCN

Scientific and Engineering Applications:

MATH/BASIC	5734-XM8
General Purpose Simulation System V OS (GPSS-V)	5734-XS2
Continuous System Modeling Program III (CSMP-III)	5734-XS9
Storage and Information Retrieval System (STAIRS/CMS)	5785-CAH
APL Forecasting and Time Series Analysis	5796-PFX
APL Statistical Library	5796-PHW
General Purpose Simulation System VSAPL (GPSS-APL)	5796-PJG
VSAPL Advanced Statistics Library (STATLIB2)	5796-PJT
APL Multivariate Time Series Analysis	5796-PLX
APL Workspace Structure Analyzer	5796-PNB
SOFTCOPY - A CADAM Drawing Viewer	5796-PNP
Pascal/VS Compiler	5796-PNQ
3277 Graphics Attachment Plotter Tablet	5798-DCE

Project Management:

Project Evaluation and Control System (PEACS)	5785-EAE
Automated Project Planning and Evaluation (APPLES)	5796-AZR

Application Development:

COBOL Interactive Debug	5734-CB4
Interactive Productivity Facility (IPF)	5748-MS1
Display Management System/CMS (DMS/CMS)	5748-XXB
Entry Level Interactive Application System I/VM (ELIAS-I/VM)	5748-XXK
Virtual Spooled Reader Display/CMS	5796-AYK
Automated Project Planning and Evaluation (APPLES)	5796-AZR
Query By Example (QBE)	5796-PKT
Structured Programming Macros	5798-CLF
VM/CMS Library and SPMOL-II Simulation	5798-CYA
Application Enabling Facility (AEF)	5798-DBF

Systems and Installation Management:

VM Monitor Analysis Program (VMMAP)	5664-191
Automated Project Planning and Evaluation (APPLES)	5796-AZR

Auditor and Security Support Aids:

Resource Access Control Facility (RACF)	5740-XXH
VM Directory Maintenance (DIRMAINT)	5748-XE4
Display Management System/CMS (DMS/CMS)	5748-XXB
VM Interactive File Sharing (VM/IFS)	5748-XXC
Resource Access Control Facility/Virtual Machine Support PRPQ (RACF/VM)	5767-002

Text and Office Systems Applications:

3277 Graphics Attachment Support	5799-AXX
Document Composition Facility (DCF)	5748-XX9
Storage and Informational Retrieval System (STAIRS/CMS)	5785-CAH

Graphics Applications:

Digital Interactive Graphics Interpretive Mapping	5668-959
Graphics Access Method (GAM/SP)	5668-978
Graphical Data Display Manager (GDDM)	5748-XXH
Interactive Circuit Design	5796-PLR
SOFTCOPY - A CADAM Drawing Viewer	5796-PNP
3277 Graphics Attachment Storage Tube	5798-DAG
3277 Graphics Attachment Plotter Tablet	5798-DCE
3279 Executive Presentation System Graphics	5798-DEB
3277 APL Graphics Attachment Support	5799-AXW
3277 Graphics Attachment Support	5799-AXX
Interactive Geo-Facilities Graphics (IGGS)	5799-AYB

Interactive and Personal Computing:

Interactive Instructional Authoring System (IIAS)	5668-011
Interactive Instructional Presentation System (IIPS)	5668-012
STAT/BASIC	5734-XA3
Business Analysis/BASIC	5734-XMB
MATH/BASIC	5734-XM8
Planning Control and Decision Evaluation (PLANCODE/I)	5740-XX8
VS/BASIC	5748-XX1

APL Support and Applications:

A Programming Language (VSAPL)	5748-AP1
APL Continuous System Modeling (CSMP)	5785-KAE
APL Forecasting and Time Series Analysis	5796-PFX
APL Function Editor	5796-PGY
APL Statistical Library	5796-PHW
APL Decision Table Processor	5796-PJB
General Purpose Simulation System VSAPL (GPSS-APL)	5796-PJG
VS APL Advanced Statistics Library (STATLIB2)	5796-PJT
APL Computer-Aided Instruction Course	5796-PJW
PL/I File Creation for APL Data Interface	5796-PKP
A Departmental Reporting System II (ADRS-II)	5796-PLN
APL Handbook of Techniques Workspace	5796-PLP
Interactive Circuit Design	5796-PLR
APL Multivariate Time Series Analysis	5796-PLX
VSAPL Extended Editor and Full Screen Manager	5796-PLY
APL Workspace Structure Analyzer	5796-PNB
APL Data Interface Version II (APL/DI-II)	5796-PNG
3277 Graphics Attachment Plotter Tablet	5798-DCE
VS APL Financial Planning System II (FPS-II)	5798-DCN
3277 APL Graphics Attachment Support	5799-AXW

Query Programs and Support:

Query Management Facility (QMF)	5668-972
Structured Query Language/Data System (SQL/DS)	5748-XXJ
Query by Example (QBE)	5796-PKT
A Departmental Reporting System II (ADRS-II)	5796-PLN
APL Data Interface Version II (APLDI-II)	5796-PNG

DOS/VS, DOS/VSE SCP: Compilers, Utilities, and Aids:

DOS PL/I Optimizing Compiler and Library	5736-PL3
DOS RPG II	5746-RG1
DOS/VS Sort/Merge Version 2	5746-SM2

OS/VS SCP: Compilers, Utilities, and Aids:

VS FORTRAN Version 2 Compiler, Library, and Interactive Debug	5668-806
VS FORTRAN Version 2 Library and Interactive Debug	5668-805
OS Assembler H	5734-AS1
COBOL Interactive Debug	5734-CB4
OS FORTRAN IV (G1)	5734-FO2
OS FORTRAN IV (H Extended)	5734-FO3
OS FORTRAN IV Library Mod II	5734-LM3
OS PL/I Checkout Compiler	5734-PL2
OS PL/I Optimizing Compiler and Libraries	5734-PL3
OS/VS COBOL Compiler and Library	5740-CB1
VS FORTRAN Compiler and Library	5748-FO3
FORTRAN Conversion Aid	5796-PFG
PL/I Language Construction Processor	5796-PLL
PASCAL/VS	5796-PNQ
Structured Programming Macros	5798-CLF

MVS SCP: Compilers, Utilities, and Aids:

FORTRAN Interactive Debug	5734-FO5
System Productivity Facility	5668-009

VM SCP: Compilers, Utilities, and Aids:

Virtual Machine Monitor Analysis Program (VMMAP)	5664-191
System Productivity Facility	5668-009
VS FORTRAN Version 2 Compiler, Library, and Interactive Debug	5668-806
VS FORTRAN Version 2 Library and Interactive Debug	5668-805
Graphics Access Method (GAM/SP)	5668-978
FORTRAN Interactive Debug	5734-FO5
VSE/Virtual Storage Access Method (VSE/VSAM)	5746-AM2
DOS/VS Sort/Merge Version 2	5746-SM2
Device Support Facility (DSF)	5747-DS1
VS FORTRAN Compiler and Library	5748-FO3
Interactive Productivity Facility	5748-MS1
VM Pass Through Facility (PVM)	5748-RC1
VM/Directory Maintenance (DIRMAINT)	5748-XE4

Remote Spooling Communication System (RSCS)	5748-XP1
Display Management System/CMS (DMS/CMS)	5748-XXB
VM Interactive File Sharing (VM/IFS)	5748-XXC
Graphic Data Display Manager (GDDM)	5748-XXH
Entry Level Interactive Application System I/VM (ELIAS-I/VM)	5748-XXK
Document Composition Facility (SCRIPT/VS)	5748-XX9
EP/VS for OS/VS and VM/370	5744-AN1
Storage and Information Retrieval System (STAIRS/CMS)	5785-CAH
CMS Host Development, Testing 8100 COBOL	5785-DCG
JES2 Information Retrieval System for CMS	5796-AYD
Virtual Spooled Reader Display/CMS	5796-AYK
FORTRAN Conversion Aid	5796-PFG
Batch Monitor for VM/CMS	5796-PGZ
Query by Example (QBE)	5796-PKT
Source Compare Audit Utility (SUPER-C)	5796-PLZ
VM Real Time Monitor (SMART)	5796-PNA
Teleprocessing Virtual Machine (TPVM)	5796-PNC
Pascal/VS Compiler	5796-PNQ
CMS Sort for VM/CMS plus extensions	5798-BDW
VM/CMS Library and SPMOL-II Simulator	5798-CYA
Application Enabling Facility	5798-DBF
Professional Office System (PROFS)	5799-BEX

Integrated Emulators

Emulator-dependent programs (except for DOS emulation under OS or OS/VS) that execute on a particular System/370 equipped with the appropriate compatibility features can execute on that System/370 in DOS or OS virtual machines under VM/SP HPO.

Figure 46 shows, by System/370 model number, which integrated emulators can execute under VM/SP HPO and the compatibility feature numbers (#xxxx) that are required.

No changes are required to the emulators, to DOS or OS, or to VM/SP HPO to allow emulator-dependent programs to execute in virtual machines.

On the System/370 Model 158 only, the virtual machine assist feature cannot operate concurrently with the 7070/7074 compatibility feature (#7117).

In an Attached Processor (AP) system, a virtual machine can use the SET AFFINITY command to make use of an emulator installed on only one of the processors. The Directory option for Affinity may be used instead, with similar results.

System/370 Model	1401 1410 1440 1460 7010	7070 7074	7080	709 7090 7094 7094II
155 II, 158	#3950	#7117		
165 II		#7117	#7118	#7119
168		#7127	#7128	#7129

Figure 46. Integrated Emulators that Execute under VM/SP HPO

Appendix B. Configuration Aid

The Configuration Aid in this appendix consists of two tables:

- Matching Hardware to Control Units
- Matching Hardware to Shared or Nonshared Virtual Control Units

The first lists hardware devices and control units (grouped by use) needed for VM/SP HPO system generation. The second matches the shared or nonshared device properties with each device that can be created as a stand-alone virtual device.

Matching Hardware to Control Units

The following table shows:

- Devices that can be attached to each control unit
- Operands that can be specified for each device in the RDEVICE macro
- Maximum number of devices specified in the FEATURE= operand of the RDEVICE macro
- Shared subchannel operation (if any)

The control units have been are placed in subgroups according to the ways they can be configured. For example, the chart of tape devices indicates that a 2401, 2402, or 2420 can be attached to a 2803 or 2804 control unit.

Type of Device	RCTLUNIT			RDEVICE	
	CUTYPE=	**Maximum FEATURE=**	**Shared Subchannel**	**DEVTYPE=**	**Other Operands**
System Consoles	1052	—	—	1052	—
	3210	—	—	3210	—
	3215	—	—	3215	—
	2150	—	—	2150	—
	3066	—	—	3066	—
	3138	—	—	3138	—
	3148	—	—	3148	—
	3158	—	—	3158	—
	3036	—	—	3036	—
	3272	—	—	3278	MODEL=2A
Transmission Control Units	2701	—	—	2701	ADAPTER=BSCA, IBM1, or TELE2
	2702	32-DEVICE	—	2702	ADAPTER=BSCA, IBM1, or TELE2 SETADDR=0, 1, 2, or 3
	2703	176-DEVICE	—	2703	ADAPTER=BSCA, IBM1, or TELE2
	3704 3705 3725	16-DEVICE 256-DEVICE 256-DEVICE	— Yes Yes	3704 3705 3725	ADAPTER=BSCA, IBM1, TELE2, TYPE1, TYPE2, TYPE3, TYPE4 , or TYPE5 MODEL=A1 through H8 (1 or 2 for 3275) SETADDR=0, 1, 2, or 3 CPTYPE=EP CPNAME=ncpname (omitted for 3275 BASEADD=cuu
	2955	—	—	2955	—
Display Devices	2848	32-DEVICE	Yes	2260 1052	—
(Local Attach.)	2845	—	Yes	2265	—
	2840	—	YES	2250	—
	2250	—	—	2250	—
	3272[1]	32-DEVICE	Yes	3277 3284 3286 3288	FEATURE=OPRDR

[1]If a 3287 is attached to a 3272, the 3287 is specified as a 3284 or 3286.

Type of Device	RCTLUNIT			RDEVICE	
	CUTYPE=	**Maximum FEATURE=**	**Shared Subchannel**	**DEVTYPE=**	**Other Operands**
Display Devices (Local Attach.) (cont.)	3274	32-DEVICE	Yes	3277 3278 3279 3284 3286 3287 3288 3289 3289E 3290	FEATURE=OPRDR MODEL=2, 3, 4, or 5 MODEL=2, or 3
	DPA	—	—	3230 3278 3279 3284 3286 3287 3288 3289 4250	
	HFCU	32-DEVICE	—	HFGD	
Remote 3270 Display Devices	2701	—	—	2701	ADDRESS=cuu (line address) ADAPTER=BSCA CLUSTER=label
	2703	—	—	2703	ADDRESS=cuu (line address) ADAPTER=BSCA or SDLC CLUSTER=label
	3704 3705 3725	—	—	3704 3705 3725	ADDRESS=cuu (line address) ADAPTER=BSCA CPTYPE=EP BASEADD=cuu CLUSTER=label
Direct Access Storage Devices	2841	—	Yes	2311 2321 2303	
	2314 2319 1FA	— — 16-DEVICE	Yes — —	2314 2319 —	
	3830 3830 3880 3880 ISC	32-DEVICE 160-DEVICE 32-DEVICE 32-DEVICE 64-DEVICE	— — — —	3330 3330 3330 3333	MODEL=1, 2, or 11 FEATURE=SYSVIRT, FEATURE=VIRTUAL MODEL=1, 2, or 11 MODEL=1 or 11
	3830 3880 ISC IFA 3830 3880 3880 3380 ISC IFA	64-DEVICE 64-DEVICE 160-DEVICE 160-DEVICE 64-DEVICE 64-DEVICE 64-DEVICE 64-DEVICE 64-DEVICE 16-DEVICE	— — — — — —	3340 3340 3350 3350 3375 3380	 FEATURE=FH FEATURE=FH

Type of Device	RCTLUNIT			RDEVICE	
	CUTYPE=	Maximum FEATURE=	Shared Subchannel	DEVTYPE=	Other Operands
Direct Access Storage Devices (cont.)		3880	16-DEVICE	—	FB-512
			32-DEVICE	—	3350P[1]
			32-DEVICE	—	3351P[2]
			16-DEVICE	—	3380P[3]
		32-DEVICE	—	3380P[4]	
		64-DEVICE	—	3310	
				3370	
	2820	—	Yes	2310	
	2835	16-DEVICE		2305	MODEL=1 or 2
	FTA	16-DEVICE	—	FB-512	
Tape Devices	2803	16-DEVICE	Yes	2401	MODEL=1, 2, 3, 4, 5, 6, or 8
	2804	16-DEVICE			FEATURE=7-TRACK, DUALDENS
				2402	MODEL=1, 2, 3, 4, 5 or 6
					FEATURE=7-TRACK, CONV, DUALDENS
				2420	MODEL=5 or 7
	3411	—	Yes	3410	MODEL=1, 2, or 3
				3411	FEATURE=7-TRACK, DUALDENS
	2403	16-DEVICE	Yes	2403	MODEL=1, 2, 3, 4, 5 or 6
	2404			2404	FEATURE=7-TRACK, CONV, DUALDENS
	2415	—	Yes	2415	MODEL=1, 2, 3, 4, 5 or 6
					FEATURE=7-TRACK, CONV
	3411	—	Yes	3410	MODEL=1, 2, or 3
				3411	FEATURE=7-TRACK, DUALDENS
	3430	—	Yes	3430	FEATURE=DUALDENS
	3480	16-DEVICE[5]	—	3480	
	3803	16-DEVICE[6]	Yes	3420	MODEL=3, 4, 5, 6, 7 or 8
					FEATURE=7-TRACK, DUALDENS

[1]When coding for the 3880 Model 11 Storage Subsystem.
[2]When coding for the 3880 Model 21 Storage Subsystem.
[3]When coding for the 3880 Model 13/23 Storage Subsystem with one string of DASD.
[4]When coding for the 3880 Model 13/23 Storage Subsystem with two strings of DASD.
[5]When two 3480 Control Units are attached via Dual Control Unit Communications Feature.
[6]FEATURE=16-DEVICE should be specified for 3803 when the communicator feature is used, allowing access to a second tape control unit and eight more tape drives.

Type of Device	RCTLUNIT			RDEVICE	
	CUTYPE=	**Maximum FEATURE=**	**Shared Subchannel**	**DEVTYPE=**	**Other Operands**
Unit Record Output Devices	2821	—	—	1403 2540P	CLASS=(class[,class . . .]) FEATURE=UNVCHSET CLASS=(class[,class . . .])
	1442 1443	—	—	1442P 1443	 CLASS=(class[,class . . .])
	3811	—	—	3211	CLASS=(class[,class . . .])
	2826	—	—	1018	
	2520	—	—	2520P	CLASS=(class[,class . . .])
	SVPC	—	—	3262	MODEL=1 or 11
	SVPC	—	—	3289	MODEL=4
	SVPC	—	—	3289E	MODEL=4
	3203	—	—	3203	MODEL=4 or 5
	3262	—	—	3262[1]	MODEL=5 CLASS=(class[,class . . .])
	3505	—	—	3525	CLASS=(class[,class . . .])
	3800	—	—	3800	FEATURE=4WCGMS, IMAGE=imagelib, CHARS=ffff,FCB=lpi, DPMSIZE=n
	4245	—	—	4245	MODEL=1 CLASS=(class[,class . . .])
	4248 2821	— —	— —	4248 2540R	MODEL=1 CLASS=(class[,class . . .])
	2520	—	—	2520R	
	3505	—	—	3505	
	1442	—	—	1442R	
	2495 2822	— —	— —	2495 2671	
	2826	—	—	1017	
	2501	—	—	2501	
Special Devices	CTCA 3088	— 64 DEVICES	Yes —	CTCA 3088	
	7443	—	—	7443	

[1]The RCTLUNIT, when coded as 3262, is valid for DEVTYPE=3262, Model=5.

Matching Hardware to Shared or Nonshared Virtual Control Units

The virtual control unit (VCU) is an internal model of the control unit. Each VCU in either a SHARED or a NONSHARED mode controls sixteen adjacent device addresses. One of the two modes must be selected when the device is attached via the three hex-symbol address, cuu, where:

- **c** indicates the channel
- The first **u** indicates the control unit
- The second **u** indicates the device.

Only devices using the same VCU mode may be added to the VCU. Also, the attached or dedicated devices assume the properties of the real control unit to which the real device is attached.

- If the attached devices are SHARED, the VCU represents a single SHARED subchannel.[12]
- If the attached devices are NONSHARED, the VCU represents up to sixteen NONSHARED subchannels connected through the same control unit.
- If the attached devices are removed from the VCU, either mode is available once again.

The following table matches the shared or nonshared device properties with each device that can be created as a stand-alone virtual device. The table shows the command or directory statement (COMMAND), the proper device specification (DEVICE SPEC), and the VCU mode (VCU MODE).

---

12 The RCUSHRD bit in RCUBLOK contains the status, shared or nonshared.

Temporary Disk Space (TDISK)

Command	Device Spec	VCU Mode
DEFINE	T2305	Nonshared
DEFINE	T2314	Shared
DEFINE	T2319	Nonshared
DEFINE	T3310	Nonshared
DEFINE	T3330	Nonshared
DEFINE	T3340	Nonshared
DEFINE	T3350	Nonshared
DEFINE	T3370	Nonshared
DEFINE	T3375	Nonshared
DEFINE	T3380	Nonshared
DEFINE	TFB-512	Nonshared

Unit Record Devices

Command	Device Spec	VCU Mode
DEFINE	READER	Nonshared
DEFINE	RDR	Nonshared
DEFINE	2540R	Nonshared
DEFINE	2501	Nonshared
DEFINE	3505	Nonshared
DEFINE	PUNCH	Nonshared
DEFINE	PCH	Nonshared
DEFINE	2540P	Nonshared
DEFINE	3525	Nonshared
DEFINE	PRINTER	Nonshared
DEFINE	PRT	Nonshared
DEFINE	PTR	Nonshared
DEFINE	1403	Nonshared
DEFINE	1443	Nonshared
DEFINE	3211	Nonshared
DEFINE	3203	Nonshared
DEFINE	3262	Nonshared
DEFINE	4245	Nonshared
DEFINE	4248	Nonshared
SPOOL	2540 READER	Nonshared
SPOOL	3505	Nonshared
SPOOL	3505	Nonshared
SPOOL	2540 PUNCH	Nonshared
SPOOL	3525	Nonshared
SPOOL	1403	Nonshared
SPOOL	1443	Nonshared
SPOOL	3211	Nonshared
SPOOL	3203	Nonshared
SPOOL	3262	Nonshared
SPOOL	4245	Nonshared
SPOOL	4248	Nonshared

Terminals and Displays

Command	Device Spec	VCU Mode
CONSOLE	1052	Nonshared
CONSOLE	3210	Nonshared
CONSOLE	3215	Nonshared
CONSOLE	3270	Nonshared
DEFINE	CONSOLE	Nonshared
DEFINE	GRAF	Shared
DEFINE	GRAF 3270	Shared
DEFINE	GRAF 3138	Nonshared
DEFINE	GRAF 3148	Nonshared
DEFINE	GRAF 3158	Nonshared
DEFINE	GRAF 3036	Nonshared
DEFINE	GRAF 3066	Nonshared
SPECIAL	3270	Shared
SPECIAL	3138	Nonshared
SPECIAL	3148	Nonshared
SPECIAL	3158	Nonshared

Miscellaneous Virtual Devices

Command	Device Spec	VCU Mode
DEFINE	LINE	Nonshared
DEFINE	TIMER	Nonshared
DEFINE	CTCA	Shared
DEFINE	3088	Nonshared
DEFINE	3289E	Nonshared
DEFINE	3800	Nonshared
SPECIAL	2701 IBM	Nonshared
SPECIAL	2701 TELE	Nonshared
SPECIAL	2702 IBM	Nonshared
SPECIAL	2702 TELE	Nonshared
SPECIAL	2703 TELE	Nonshared
SPECIAL	TIMER	Nonshared
SPECIAL	CTCA	Shared
SPECIAL	3088	Nonshared
SPOOL	3289E	Nonshared
SPOOL	3800	Nonshared

Appendix C. Compatible Devices

The devices listed below are functionally equivalent to the 2770 Communication System operating on a VM/370 Remote Spooling Communication Subsystem. Details on the feature requirements for VM/370 RSCS use and operational control of such devices are not contained in VM/SP HPO publications but in the programming and operating publications that support these devices.

IBM 6640 Document Printer

- *Programming Guide for Communicating with the IBM 6640 Document Printer*, Form No. G544-1001
- *IBM 6640 Document Printer—Communicating User's Guide*, Form No. S544-0507
- *IBM 6640 Document Printer—Communicating Operating Instructions*, Form No. S544-0506

IBM Office System 6 Information Processors (6/650, 6/440, 6/430)

- *Programming Guide for Communicating with the IBM Office System 6 Information Processors*, Form No. G544-1003
- *IBM 6/450, 6/440, and 6/430 Information Processors—Communicating User's Guide*, Form No. S544-0521
- *IBM 6/450, 6/440, and 6/430 Information Processors—Communicating Operating Instructions*, Form No. S544-0522

IBM Mag Card II Typewriter and IBM 6240 Mag Card Typewriter

- *Programming Guide for Communicating with the IBM Mag Card II Typewriter and the IBM 6240 Mag Card Typewriter*, Form No. G544-1005
- *IBM Mag Card II Typewriter—Communicating and IBM 6240 Mag Card Typewriter—Communicating Reference Guide*, Form No. S544-0549
- *IBM Mag Card II Typewriter—Communicating and IBM 6240 Mag Card Typewriter—Communicating Operating Instructions*, Form No. S544-1005

Appendix D. VM/SP HPO Restrictions

A virtual machine created by VM/SP HPO (used in conjunction with VM/SP) is capable of running a System/370 operating system as long as the virtual machine operation does not violate certain restrictions. Virtual machine restrictions and certain execution characteristics follow.

Channel Program Restrictions

1. Avoid using channel programs that loop. Execution of a Backward Transfer In Channel CCW followed by an I/O CCW that present concurrent Channel End and Device End may result in locking out the device as well as the channel. Users attempting to access devices on the channel will also be locked out. To recover from this situation, issue the CP HALT command to the device or have the operator issue a SYSTEM RESET.

2. When issuing CCW's in which a data address is specified, that address must be within the virtual machine size regardless of whether data transfer is involved or not. The use of an address above the virtual machine size will result in VM/SP forcing a channel program check.

Dynamically Modified Channel Program Restrictions

1. In general, virtual machines may not execute channel programs that are dynamically modified (that is, channel programs that are changed between the time the START I/O (SIO) is issued and the time the I/O ends, either by the channel program itself or by the processor).

   Exceptions (that is, dynamically modified channel programs given special consideration by CP) are:

   - Those generated by the Indexed Sequential Access Method (ISAM) running under OS/PCP, OS/MFT, and OS/MVT
   - Those generated by ISAM running in an OS/VS virtual=real partition
   - Those generated by the OS/VS Telecommunications Access Method (TCAM) Level 5, with the VM/370 option
   - Those containing polling sequences

Notes:

a. *The self-modifying channel programs that ISAM generates for some of its operations receive special handling if the virtual machine using ISAM has that option specified in its directory entry. If ISAM is to run in an OS/VS virtual=real partition, you must specify the ISAM option in the directory entry for the OS/VS virtual machine. You do not have to specify the ISAM option in the directory entry for an OS/VS virtual machine running DOS/ISAM or ISAM in a virtual=virtual partition.*

b. *Virtual machines using OS/VS TCAM (Level 5, generated or invoked with the VM/370 option) modify the channel program with a DIAGNOSE instruction. This instruction causes CP to forward the change made in the virtual CCW string to the real CCW string being executed by the channel. CP is then able to execute the dynamically modified channel program properly.*

c. *When a virtual machine starts a channel program containing a polling sequence, the CCW translation sets the PCI bit in the real CCW string. Each time the real CCW string is executed, the resulting PCI interruption causes CP to examine the corresponding virtual CCW string for changes. Any changes to the virtual CCW string are also made to the real CCW string while it is executing.*

The restrictions for using dynamically modified channel programs do not apply if the virtual machine is using the virtual=real performance option of the SET command with the NOTRANS option set ON.

Minidisk Restrictions

The following restrictions exist for minidisks:

1. In the case of Read Home Address with the Skip bit off, CP modifies the home address data in user storage at the completion of the channel program because the addresses must be converted for minidisks. Therefore, the data buffer area may not be dynamically modified during the input/output operation.

2. In the case of read device characteristics to an FBA device with the Skip bit off, CP modifies the data in user storage at completion of the channel program so the data reflects the true minidisk size and characteristics. Therefore, the data buffer area cannot be dynamically modified during the input/output operation.

   Note: You should not attempt to use this data during the I/O operation.

3. On a minidisk, if a CCW string uses multitrack search on input/output operations, subsequent operations to that disk must have preceding seeks or continue to use multitrack operations. There is no restriction for dedicated disks.

4. OS/PCP, MFT, and MVT ISAM or OS/VS ISAM running virtual=real may be used with a minidisk only if the minidisk is located at the beginning of the physical disk (that is, at cylinder 0). There is no such restriction for DOS ISAM or OS/VS ISAM running virtual=virtual.

   Note: Because the VS1 system does no paging, CP treats any ISAM programs run under VS1 as though they are running in an ADDRSPC=REAL partition.

5. CP does not return an end-of-cylinder condition to a virtual machine that has a virtual 2311 mapped to the top half (that is, tracks 0 through 9) of 2314 or 2319 cylinders.

6. If the user's channel program for a count-key-data minidisk does not perform a Seek operation, then to prevent accidental accessing, CP inserts a positioning Seek operation into the user's channel program. Thus, certain channel programs may generate a condition code (CC) of 0 on a SIO instead of an expected CC of 1, which is returned to the virtual machine. The final status is returned to the virtual machine as an interruption.

7. A DASD channel program directed to a 3330, 3340, 3350, 3375, or 3380 device may give results on dedicated drives that differ from results on minidisks having nonzero relocation factors. This occurs if the channel program includes multiple-track operations and depends on a Search Id High or Search Id Equal or High I/O command code to terminate the program. This is because the record 0 count fields on the 3330, 3340, 3350, 3375, and 3380 devices must contain the real cylinder number of the track on which they reside. Therefore, a Search Id High, for example, based on a low virtual cylinder number may terminate prematurely if a real record 0 is encountered.

   Notes:

   a. *Minidisks with nonzero relocation factors on 3330, 3340, 3350, 3375, and 3380 devices are not usable under OS and OS/VS systems. This is because the locate catalog management function employs a Search Id Equal or Search Id High I/O command code to find the end of the VTOC.*

   b. *On a 3330, 3340, or 3350 DASD with a VTOC of more than one track, an OS or OS/VS minidisk must start at real cylinder 0.*

   c. *OS or OS/VS minidisks with a VTOC of more than one track must not be relocated.*

   d. *If the R byte field of CCHHR is equal to zero when a virtual SIO is issued, but the CCHHR field is read in dynamically by the channel program before the Search ID CCW is executed, the real Search ID CCW will use the relocated CCHHR field that was dynamically read in, causing incorrect results. To avoid this problem, the R byte of CCHHR should not be defaulted to binary zero by the virtual machine if the search arguments are to be read in dynamically and a Search ID on Record R0 is not desired.*

8. The Device Support Facility cannot assign alternate tracks for a 3330, 3340, 3350, 3375, 3380 or FBA minidisk.

9. If the DASD channel programs directed to the 3330, 3340, 3350, 3375 and 3380 devices include a write record R(0), then results differ depending on whether the direct access storage device is dedicated or nondedicated. A full-pack minidisk is treated as a nondedicated device. For dedicated devices of the listed device types, a write R(0) is allowed, but you must be aware that the track descriptor record may not be valid from one DASD to another. For a nondedicated device, a write record R(0) is replaced by a read record R(0) and the skip flag is set on. This could result in a command reject condition due to an invalid command sequence.

10. When performing DASD I/O to a 3330, 3340, 3350, 3375, or 3380, the real Search Id command uses the relocated search argument instead of the argument that was read dynamically if the record field of the search argument is zero. To avoid this problem, the record field of a Search Id I/O command should not be set to binary zero if the search argument is to be dynamically read or if a Search Id on record 0 is not intended.

11. When performing DASD I/O to 2305, 2311, 2314, or 2319 devices, VM will not relocate the search argument in the Search Id command. This is because the Write Record R(0) is allowed to pass through to the hardware. For minidisks, therefore, the cylinder address in record zero will contain the virtual cylinder address after a Write R(0) has been issued.

12. On FBA devices, the use of the CE area is different for dedicated devices and minidisks. Any user with a dedicated device can use the CE area. However, only class F users can use the CE area for minidisks.

13. FBA diagnostic commands are also handled differently for dedicated devices and minidisks. Any user with a dedicated device can issue diagnostic CCWs. For minidisks, however, only users with a minidisk equal to the size of the entire pack can issue a diagnostic control command. Because Diagnostic Sense commands must be chained from a Diagnostic Control command, this restriction indirectly applies to those commands also.

14. Diagnostic Read Home Address and Diagnostic Write Home Address I/O commands are supported only for:

    - Dedicated devices
    - Minidisks that start at real cylinder 0

    Only privilege class F users may use the I/O command Diagnostic Write Home Address.

15. Refer to *Device Support Facilities*, GC35-0033, for procedures to format 3375/3380 DASD for use in an OS/VS operating system running in a virtual machine.

16. Any minidisk that is used by an MVS/SP virtual machine operating with preferred machine assist, must be allocated as a full-extent virtual disk (the entire pack). Because a temporary disk (TDISK) is never full-extent, do not specify a TDISK for a virtual machine using preferred machine assist.

17. Do not use duplicate volid's for SYSRES or CP-owned volumes, else results are unpredictable.

18. To ensure that the saved copy of the S-STAT or Y-STAT is current, a validity check is performed when a saved system is IPLed. This check is performed only for S-DISKS and Y-DISKS formatted in 512-, 1024-, 2048-, or 4096-byte CMS blocks. For 800-byte blocks disks, the saved copy of the S-STAT or Y-STAT is used. The validity checking consists of comparing the date when the saved directory was last updated with the date when the current disk was last updated. If the dates for the S-STAT are different, then the S-STAT is built in user storage. If the dates for the Y-STAT are different, then the Y-disk is accessed using the CMS ACCESS command: ACCESS 19E Y/S* * Y2[13]. This means that even when the S- and Y-disks are accessed in read/write mode and then RELEASED, the message DMSINS100W S-STAT and/or Y-STAT NOT AVAILABLE will result.

19. A guest operating system which receives a unit check in response to a START I/O command without the VIRTUAL RESERVE option should issue a SENSE I/O command without VIRTUAL RESERVE. If another guest operating system issued START I/O RESERVE at the same time as the first user issued START I/O, the second user has now reserved the minidisk. Issuing SENSE I/O RESERVE will cause the sense data to be passed to the first user; but the first user will not be able to reserve the minidisk. There will be no indication that the RESERVE has failed.

20. A minidisk should not be located where its track 0, cylinder 0 falls on a defective track of a 3340 or 3344 storage device, because it will be impossible for the CP IPL command to function for that minidisk.

21. Any operating system doing SIO to a disk with a defective 3340 or 3344 track must be capable of doing normal alternate track error recovery.

22. VM/SP HPO does not perform alternate track recovery for overflow record processing on 3340 or 3344 storage devices. If a defective track is encountered during overflow record processing, a fatal I/O error results.

23. When a virtual 3330 Model 1 is mapped to a real 3330 Model 11 and a virtual machine references sense information during error recovery, incorrect results will occur. Since sense byte 6, bits 1 and 2, is refernced differently by 3330 Models 1 and 11, the results will be unexpected.

---

[13] The DASD address of the Y-DISK will be whatever was specified when CMS was generated. For the standard system this will be 19E.

Timing Dependencies

Time dependent applications in input/output devices or programming do not function consistently under VM/SP HPO:

1. The following telecommunication access methods (or the designated option) violate the restriction on timing dependency by using program-controlled interruption techniques and/or the restriction on dynamically modified channel programs:

   - OS Basic Telecommunications Access Method (BTAM) with the dynamic buffering option
   - OS Queued Telecommunications Access Method (QTAM)
   - DOS Queued Telecommunications Access Method (QTAM)
   - OS Telecommunications Access Method (TCAM)
   - OS/VS Telecommunications Access Method (TCAM) Level 4 or earlier, and Level 5 if VM is not generated or invoked with the VM option

   These access methods may run in a virtual=real machine with CCW translation suppressed by the SET NOTRANS ON command. Even if SET NOTRANS ON is issued, CCW translation will take place if one of the following conditions is in effect:

   - The channel program is directed at a nondedicated device (such as a spooled unit record device, a virtual CTCA, a minidisk, or a console).
   - The channel program starts with a Sense operation code.
   - The channel program is for a terminal invoked by the DIAL command.
   - Start I/O tracing is in effect.
   - The CAW is in page zero or beyond the end of the virtual=real area.

   (OS BTAM can be generated without dynamic buffering, in which case no virtual machine execution violations occur. However, the BTAM reset poll macro will not execute under VM/SP HPO if issued from third level storage. For example, a reset poll macro has no effect if executed from a virtual=virtual storage under a VS1 system that is running under VM/SP HPO.)

2. Programming that makes use of the PCI channel interruption for channel program or processor signalling must be written to allow processing to continue normally if the PCI interruption is not recognized until I/O completion or if the modifications performed are not executed by the channel.

3. Devices that expect a response to an interruption within a fixed period of time may not function correctly because of execution delays caused by normal VM/SP HPO system processing. An example of such a device is the IBM 1419 Magnetic Character Reader.

4. The operation of a virtual block multiplexer channel is time dependent. For this reason, the channel appears available to the virtual machine operating system, and channel interruptions are not observed. However, operations on virtual block-multiplexing devices should use the available features like Rotational Position Sensing to enhance utilization of the real channels.

5. Multiple errors on the same channel may occur before CP's channel check handler has finished processing the first error. Depending on the timing sequence, various abends may occur.

Processor Model-Dependent Functions

On the System/370 Model 158 and the I/O processor of the 158AP, the Virtual Machine Assist feature and the System/370 Extended Feature are mutually exclusive; however, installation of RPQ (#MK3272) allows coexistence of Virtual Machine Assist and the System/370 Extended Feature on a 158-3.

Programs written for processor-model dependent functions may not execute properly in the virtual machine under VM/SP High Performance Option. The following points should be noted:

1. Programs written to examine the machine logout area do not have meaningful data since CP does not reflect the machine logout data to a virtual machine.

2. Programs written to obtain processor identification (via the CP command STIDP) receive the real processor value. When the STIDP instruction is issued by a virtual machine, the version code contains the value 255 in hexadecimal ("FF") to represent a virtual machine.

3. No simulation of other processor models is attempted by VM/SP HPO.

4. Since an operating system's channel error recovery procedures may be model-dependent for processors and channels, operating systems that will run in a virtual machine may have to be generated for the same model of processor that VM/SP HPO will be running on.

5. Shared segments and named systems created on processors utilizing 2K storage protection keys may not execute correctly on processors that utilize 4K storage protection keys.

6. User application programs that rely on 2K key virtual storage may not execute correctly when loaded on a 4K key processor or when loaded on a processor that contains both 4K key and 2K key real storage frames.

7. Virtual machines saved on 2K key processors using the CP command VMSAVE or SAVESYS may not execute correctly when loaded on 4K key processors.

Channel Model-Dependent Functions

1. Channel checks (channel data check, channel control check, and interface control check) are returned to the virtual machine as are other I/O errors. This provides the operating system or other programs in the virtual machine with the opportunity to attempt recovery or close out its operation in an orderly manner. To take full advantage of this, the virtual machine should comply with the following requirement:

   - Each virtual channel should map to real channels of a single type. In other words, the virtual devices on a virtual channel should all map to real devices on real channels of a single type and model. These real channels should all be the same as each other, but not necessarily the same as the virtual channel.

   If the I/O configuration of a virtual machine does not meet the above requirement, no warning message is issued and the virtual machine will run successfully until a channel check occurs. In this case, when a channel check occurs, there is a possibility that the channel extended logout data area may be inconsistent with the data provided by the store channel id (STIDC) instruction.

   Note: Virtual machines running CMS need not comply with these requirements. Here, only unit record spooling and diagnose I/O are performed. For unit record spooling, there are no channel checks and for diagnose I/O, CP attempts to perform the error recovery itself.

   When the store channel id instruction (STIDC) is executed in a virtual machine, it returns information from an arbitrary channel. The type, model, and logout length data returned by the STIDC are the same as the real channel except that when a real channel is a block multiplexer and the virtual channel is a selector, the type field returned by STIDC indicates a selector channel.

   Since the STIDC returns identifying data from the real channel, channel type-dependent error recovery procedures can use STIDC to identify the channel.

2. Channel extended logouts are reflected to the virtual machine in a manner that is model-dependent for processors and channels and consistent with the data returned by STIDC (provided that the virtual-to-real channel mapping requirement stated previously is met).

   A deviation in handling channel extended logouts occurs if the virtual machine uses the control bit in register 14 to mask out channel extended logouts. Channel extended logouts will not be lost when they are kept pending along with their associated I/O interruptions. Regardless of whether or not the setting causes the virtual machine to lose the channel extended logout, CP will still successfully record the logout in its own error recording areas.

Virtual Machine Characteristics

Other characteristics that exist for a virtual machine under VM/SP HPO are as follows:

1. The minimum virtual machine size is 8K. The maximum size for a virtual machine is 16 megabytes. The size is defined in the user directory and may be temporarily altered by the user with the CP command DEFINE.

2. If the virtual=real option is selected for a virtual machine, input/output operations specifying data transfer into or out of the virtual machine's page zero, or into or out of storage locations whose addresses are greater than the storage allocated by the virtual=real option, must not occur. The storage protection key mechanism of the processor and channels operates in these situations but is unable to provide predictable protection to other virtual machines. In addition, violation of this restriction may compromise the integrity of the system. The results are unpredictable.

3. The DIAGNOSE instruction cannot be issued by the virtual machine for its normal function. CP uses this instruction to allow the virtual machine to communicate system services requests. The diagnose interface requires the operand storage addresses passed from the virtual machine issuing the DIAGNOSE instruction to be real. For more information about the DIAGNOSE instruction in a virtual machine, see the *VM/SP HPO System Programmer's Guide.*

4. A control unit normally does not appear busy to a virtual machine. An exception exists when a forward space file or backward space file command is executed for a tape drive. Subsequent I/O operations to the same virtual control unit result in a control unit busy condition until the forward space file or backward space file command completes. If the real tape control unit is shared by more than one virtual machine, a control unit busy condition is reflected only to the virtual machine executing the forward space file or backward space file command. When a virtual machine attempts an I/O operation to a device for which its real control unit is busy, the virtual machine is placed in I/O wait (nondispatchable) until the real control unit is available. If the virtual machine executed a SIOF instruction (rather than SIO) and was enabled for block-multiplexing, it is not placed in I/O wait for the above condition.

5. The CP command IPL cannot simulate self-modifying IPL sequences from dedicated unit record devices or certain self-modifying IPL sequences from tape devices.

6. The VM/SP HPO spooling facilities do not support punch-feed-read, stacker selection, or column binary operations. Detection of carriage control channels is supported for a virtual 3211 only.

7. VM/SP HPO does not support count control on the virtual 1052 operator's console.

8. Programs that use the integrated emulators function only if the real computing system has the appropriate compatibility feature. VM/SP HPO does not attempt simulation. The DOS emulator running under OS or OS/VS is not supported under VM/SP High Performance Option.

9. The Read Direct and Write Direct instructions are not supported for a virtual machine.

10. The SET CLOCK instruction cannot be simulated, and hence, is ignored if issued by a virtual machine. The STORE CLOCK instruction is a nonprivileged instruction and cannot be trapped by the control program; it provides the true TOD clock value from the real processor.

11. The 1050/1052 Model 2 Data Communication System is supported only as a keyboard operator's console. Card reading, paper tape I/O, and other modes of operation are not recognized as unique, and hence may not work properly. This restriction applies only when the 1050 system is used as a virtual machine operator's console. It does not apply when the 1050 system is attached to a virtual machine via a virtual 2701, 2702, or 2703 line.

12. The pseudo-timer (usually device address 0FF, device type TIMER) does not return an interruption from a Start I/O; therefore, do not use EXCP to read this device.

13. A virtual machine device IPL with the NOCLEAR option overlays one page of virtual machine storage. The IPL simulator uses one page of the virtual machine to initiate the IPL function. The starting address of the overlaid page is either the result of the formula

$$\frac{\text{virtual machine size}}{2}$$

or the hexadecimal value 20000, whichever is smaller.

14. When CMS is IPLed by name and the virtual storage size is too small to contain all of the saved CMS pages (as specified by SYSPGNM in DMKSNT), the CMS saved system is treated by CP as a discontiguous saved segment. The virtual machine's segment table is expanded to address all of the virtual storage needed for CMS. If there are shared segments in CMS beyond the VMSIZE (the field in VMBLOCK representing the storage size of the virtual machine), any subsequent IPL causes the saved CMS to be purged in its entirety. The segment table is adjusted back to the VMSIZE, and all virtual pages associated with the CMS saved system within the VMSIZE are cleared. This means that if data is stored into CMS saved pages in low storage, the pages will be cleared, even if the NOCLEAR option is specified for the IPL. To make the data available for the IPL, either define storage large enough to accommodate all of CMS or use the STOP option on the second IPL, store the data required for the IPL, and continue with the BEGIN command.

15. To maintain system integrity, data transfer sequences to and from a virtual system console are limited to a maximum of 2032 bytes. Channel programs containing data transfer sequences that violate this restriction are terminated with an interruption whose CSW status indicates incorrect length and a channel program check.

    Notes:

    a. *A data transfer sequence is defined as one or more connected read or write CCWs. The introduction of command chaining defines the start of a new data transfer sequence.*

    b. *Data chained Seek CCWs with counts of less than four are inconsistent with VM/SP HPO data security and therefore will give an error when attempting to use.*

16. When an I/O error occurs on a device, the processor maintains a contingent connection for that device until a SENSE CHANNEL command is executed and sense data is recorded. No other I/O activity can take place on the device during this time. Under VM/SP HPO, the contingent connection is maintained until the SENSE command is executed, but I/O activity from other virtual machines or another processor can begin on the device while the sense data is being reflected to the virtual machine. The user should therefore be aware that, on a shared disk, the access mechanism may have moved during this time.

17. The mode setting for 7-track tape devices is maintained by the control unit. Therefore, when a virtual machine issues the Set Mode channel command to a 7-track tape device, it changes the mode setting of all 7-track tape devices attached to that control unit.

    This has no effect on virtual machines (such as OS or DOS) that issue Set Mode each time a CCW string is to be executed. However, it can cause a problem if a virtual machine fails to issue a Set Mode with each CCW string executed. Another virtual machine may change the mode setting for another device on the same control unit, thereby changing the mode setting for all 7-track tape devices attached to that control unit.

18. A shared system or one that uses discontiguous saved segments cannot be loaded (via IPL) into a virtual machine running in the virtual=real area.

19. The DUMMY feature for VSAM data sets is not supported and should not be used at program execution time. Specifying this option on the DLBL command will cause an execution-time OPEN error.

20. The 3066 is supported as a 3215. It is not supported as a graphics terminal; therefore, it is recommended that the NODISP option of the EDIT command be used when editing in a 3066.

21. The Program Controlled Interruption (PCI) FETCH option for load module retrieval is not supported for OS/MFT or VS1.

22. 308x and 3090 processors do not permit use of one megabyte segments for virtual machines. Any attempt by a relocatable virtual machine using 1 Mb segments to invoke the DAT facility for address translation, results in a translation exception.

23. MVS subsystems and application programs that issue the store processor id (STIDP) instruction to determine whether they are executing under VM/SP HPO, should consider the following:

    a. If the MVS/SP virtual machine is using preferred machine assist and the STIDP instruction gives a return code X'FE', this indicates that CP will not intercept any DIAGNOSE instructions. The MVS/SP virtual machine with preferred machine assist runs in supervisor state.

    b. If the MVS/SP virtual machine is using preferred machine assist and the STIDP instruction gives a return code X'FE', the MVS subsystem or application program must use the TEST BLOCK instruction to determine available storage locations. Do not use the MVCL instruction. Use of the MVCL instruction for determining available storage locations produces unpredictable results.

    c. If the MVS/SP virtual machine is using preferred machine assist with control switch assist, the STIDP instruction gives a return code of X'FF'. See the *VM/SP HPO System Programmer's Guide* for the diagnose instructions that are supported.

24. VM/SP HPO executes in a virtual machine under VM/System Product Release 2; however, such a virtual machine cannot use preferred machine assist.

25. If you are running MVS version 1 release 3 or a later release in a V=R virtual machine on a 3033 processor and want to use SPMODE and/or QVM commands, either

    - Have preferred machine assist active,

      or

    - Use cross memory assist.

    If you are running MVS version 1 release 3 or a later release in a V=R virtual machine on a 308x or 4381 processor and want to use SPMODE and/or QVM commands, have preferred machine assist active.

26. When MVS/SP is executing with preferred machine assist active, the Diagnose instruction cannot be used to communicate between MVS/SP and VM/SP HPO. This restriction does not apply to preferred machine assist with control switch assist.

27. An MVS guest operating system using

    - Single processor mode
    - Nondisruptive transition

    on the 308x or 3090 processor must use preferred machine assist or preferred machine assist with control switch assist.

28. Some processors require certain levels of MVS/SP in order to use certain functions:

Processor	Function	MVS/SP Release Required
3090	• Preferred machine assist • Preferred machine assist with the control switch assist • Use of storage above 16 Mb by MVS/SP preferred guest • Use of single processor mode by MVS/SP preferred guest • Use of dynamic SCP transition by MVS/SP preferred guest	MVS/SP Version 1 Release 3 Modification 5
4381 308x	• Preferred machine assist • Preferred machine assist with the control switch assist • Single processor mode • Dynamic SCP transition	MVS/SP Release 1 Enhancement
	• Use of storage above 16 Mb by MVS/SP preferred guest • Use of single processor mode by MVS/SP preferred guest • Use of dynamic SCP transition by MVS/SP preferred guest	MVS/SP Release 3
3033	• Preferred machine assist	MVS/SP Release 1 Enhancement
	• Use of storage above 16 Mb by MVS/SP guest • Single processor mode • Dynamic SCP transition	MVS/SP Release 3

29. Since MVS/SP does not support the 3090 Vector Facility, the MVS preferred machine assist guest cannot use it.

30. Because MVS/SP does not support the use of the Vector Facility in 370 mode, V=R recovery does not save vector status or vector registers.

31. Extended control mode (ECMODE) virtual machines executing with DAT on in a processor with DAT on where available real storage exceeds 16Mb must use 4K pages. The ECMODE virtual machine receives a program check if the page size is not 4K.

32. The V=R virtual machine will not be recovered following a CP abend if a different CP nucleus was loaded onto the system disk after the system became operational. Do not load a new CP nucleus onto the system disk after VM/SP HPO is IPLed. If a new nucleus is loaded, MVS/SP V=R recovery results are unpredictable.

33. When you issue CCW's containing an address, that address must be within the virtual machine size whether or not data transfer is involved. Using an address above the virtual machine size will result in VM/SP HPO forcing a channel program check.

34. VM users may access the 3800 Model 3 Printing Subsystem. Existing programs designed to produce 3800 Model 1 printer output may produce output for the 3800 Model 3 printer with little or no program change. Customized 3800 Model 1 library character sets (LCSs) and graphic character modification modules (GRAPHMODs) are not interchangeable with 3800 Model 3 character sets. Users may recode customized 3800 Model 1 character sets and build new modules through the use of the IEBIMAGE utility program. The MVS Character Conversion Aid may also be used to convert customized Model 1 LCSs and GRAPHMODs for use in the 3800 Model 3 environment. The converted LCS and GRAPHMOD sources are written to an MVS partitioned data set which is then used as input to the IEBIMAGE utility program. Communication with IEBIMAGE is provided through the GENIMAGE, IMAGELIB, and IMAGEMOD commands.

3340/3344 Direct Access Storage Device Restrictions

1. A minidisk should not be located where its track 0, cylinder 0 falls on a defective track of a 3340 or 3344 storage device, because it will be impossible for the CP IPL command to function for that minidisk.

2. Any operating system doing SIO to a disk with a defective 3340 or 3344 track must be capable of doing normal alternate track error recovery.

3. VM/SP HPO does not perform alternate track recovery for overflow record processing on 3340 or 3344 storage devices. If a defective track is encountered during overflow record processing, a fatal I/O error results.

Mass Storage System Restrictions

1. There are two OS/VS system data sets associated with Mass Storage System: the mass storage volume inventory and the mass storage volume control journal. There is one copy of each data set per Mass Storage System, not necessarily one per operating system. If more than one OS/VS system (running either in native mode or in a virtual machine) is connected to a common Mass Storage System, then the OS/VS systems must share a common inventory and journal.

2. When a real 3330V device is dedicated to a virtual machine as a virtual 3330V, the programming support in the virtual machine must recognize and access the virtual device as a 3330V.

3. The following must be compatible: the definition of 3330V addresses in the MCS tables; the DMKRIO module; and the IOGEN for any OS/VS system running in a virtual machine with a dedicated MSC port.

4. Each active volume in the MSS must have a unique volume number. If you wish to have two or more user volumes having the same volume serial (such as different versions of an OS/VS2 system residence volume both having a volume serial of VS2037), then create two MSS volumes having different volume serials and allocate the user volumes as minidisks.

5. Mass Storage System volumes may not be used for system residence, paging, spooling, or temporary disk space.

6. You must not change the volume serial of a real 3330V volume (the volume serial as known by the MSC) except by using the OS/VS access method services utilities. If, for example, cylinder 0 of a 3330V is dedicated to a virtual machine and that virtual machine alters the volume serial using DDR, then the volume cannot be mounted.

7. The MSS Central Service Application Program is not supported by MVS/SP when MVS is executing with preferred machine assist. This restriction does not apply when MVS is executing with preferred machine assist with control switch assist.

8. CP commands must not be issued from the central server VIRTSYS that require action from the central server.

9. If virtual volumes are to be shared between processors, the virtual machine must handle cylinder faulting.

3880 Storage Subsystem Restrictions

1. Storage devices attached to a 3880 Model 11 or Model 21 Storage Subsystem cannot be used for spooling, minidisks, temporary disks, directories, as system residence volumes, or to receive VMSAVE data. This restriction does not apply to the 3880 Model 13 or Model 23 Storage Subsystem.

2. The cache of a 3880 Model 11 or 21 Storage Subsystem cannot be shared between virtual machines. It must be dedicated to a single virtual machine or owned by the system. This restriction does not apply to the 3880 Model 13 or Model 23 Storage Subsystem.

3. Users of 308x or 3090 processors must execute the Input/Output Configuration Program to define the 3880 Storage Subsystem to the system.

4. Because 3880 Storage Subsystem addresses span multiple control units, you must code the FEATURE=32-DEVICE operand of the RCTLUNIT macro for the Model 11 and Model 21, and for the Model 13 and Model 23 if they control two strings of DASD. If the Model 13 and 23 control only one string of DASD, code FEATURE=16-DEVICE.

5. For the 3880 Model 11 and Model 21 Storage Subsystem, you must supply the direct mode addresses when defining devices in the RDEVICE macro or when referring to devices in a CP command such as VARY, ATTACH, or DETACH.

6. When you code the control unit address of the 3880 Model 11 Storage Subsystem, the control unit portion of the address must be an even number. The device portion of the address must range from 0 to 7.

7. When you code the control unit addresses of the 3880 Model 21 Storage Subsystem, the device portion of each address must range from 0 to 3.

8. When you define a virtual storage device for a 3880 Model 11 Storage Subsystem, the control unit portion of the address must be an even number, and the device portion of the address must be the same as the device portion of the real address.

9. When you define a virtual storage device for a 3880 Model 21 Storage Subsystem, the device portion of the address must be the same as the device portion of the real address.

10. When you code RDEVICE macros for the Model 21, Model 13, or Model 23, you must include the ALTCU=ccuu operand if there is a path to the string of DASDs from the second storage director.

11. Allocate all storage attached to a 3880 Model 11 or 21 Storage Subsystem as PAGE space. If you allocate storage space any other way, it will be reallocated as PERM space. This restriction does not apply to the 3880 Model 13 or Model 23 Storage Subsystem.

12. The 3380 DASD devices attached to a 3880 Model 13 or 23 Storage Subsystem must have the same address from the point of view of either storage director. In other words, the device portion of the ADDR and ALTCU operands of the RDEVICE macro must be the same.

13. Do not use the 3880 Model 13 or Model 23 Storage Subsystem for paging, spooling, or swapping unless you have disabled the cache by issuing the CACHE OFF command.

Conversational Monitor System Restrictions

The following restrictions apply to CMS, the conversational monitor system of VM/SP HPO.

Note: VM/SP HPO adds no additional functions to the CMS component of VM/SP.

1. CMS executes only on a virtual machine controlled by VM/SP or VM/SP HPO.

2. When you install VM/SP HPO Release 4.0 and later releases, you must take special care with any CP commands you have changed. If you have

   - Defined new CP commands or defined new subcommands for existing CP commands
   - Changed the abbreviations for CP commands
   - Redefined user privilege classes

   consult the section "How to Add a Command to CP" in the *VM/SP HPO System Programmer's Guide.*

3. CMS employs the spooling facilities of VM/SP to perform unit record I/O. However, a program running under CMS can issue its own SIOs to attached dedicated unit record devices.

4. Only those OS and DOS facilities that are simulated by CMS can be used to execute OS and DOS programs produced by language processors under CMS.

5. Programs that operate under CMS are encouraged to use documented interfaces. Those programs which modify DMSNUC or other CMS control blocks in order to accomplish their interfaces with the CMS system, may hamper the performance and reliability of the system.

6. CMS can read sequential and partitioned OS data sets and sequential VSE files.

   The following restrictions apply when CMS reads OS data sets that reside on OS disks:

   - Except for VSAM, read-password-protected data sets are not read.
   - BDAM and ISAM data sets are not read.
   - Multivolume data sets are read as single-volume data sets. End-of-volume is treated as end-of-file and there is no end-of-volume switching.
   - Except for VSAM, keys are ignored and only the data is read.
   - User labels in user-labeled data sets are bypassed. The following restrictions apply when CMS reads VSE files that reside on DOS disks:
     - Only VSE sequential files can be read. CMS options and operands that do not apply to OS sequential data sets (such as the MEMBER and CONTACT options of FILEDEF and the PDS option of MOVEFILE) also do not apply to VSE sequential files.
     - The following types of DOS files cannot be read:
       - VSE DAM and ISAM files
       - Files with the input security indicator on
       - VSE files that contain more than 16 extents

       Note: User labels occupy the first extent; therefore, the file can hold only 15 additional data extents.

   - Multivolume files are read as single-volume files. End-of-volume is treated as end-of-file. There is no end-of-volume switching.
   - User labels in user-labeled files are bypassed.
   - Since VSE files do not contain BLKSIZE, RECFM, or LRECL parameters, these parameters must be specified via FILEDEF or DCB parameters; otherwise, defaults of BLOCKSIZE = 32760 and RECFM = U are assigned. LRECL is not used for RECFM = U files.
   - CMS does not support the use of OS/VS DUMMY VSAM data sets at program execution time. Specifying the DUMMY option with the DLBL command causes an error.

7. Assembler program usage of the ISAM Interface Program (IIP) is not supported.

8. If CMS cannot calculate a valid time, it will display *.** in place of n.nn or x.xx.

9. Only those OS and DOS facilities that are simulated by CMS can be used to execute OS and VSE programs produced by language processors under CMS.

10. Many types of object programs produced by CMS (and OS) languages can be executed under CMS using CMS's simulation of OS supervisory functions. Although supported in OS and VSE virtual machines under VM/SP the writing and updating of non-VSAM OS data sets and VSE files are not supported under CMS.

11. Setting the PSW EC mode bit on is not recommended because CMS handles interrupts in BC mode only.

12. CMS/DOS support is based on the VSE/Advanced Functions program product. With VSE, prior releases of VSAM are not supported under CMS/DOS.

13. System logical units (SYSIN, SYSRDR, SYSIPT, SYSLST, and SYSPCH), are not supported for VSE formatted FBA devices because the SYSFIL function (SVC 103) of VSE is not supported under CMS/DOS.

14. Although programs compiled using CMS/DOS can be transferred directly to a VSE machine, this is not recommended for the following reasons:

    - The CMS/DOS VSE linkage editor is designed to link edit VSE programs under CMS/DOS only.
    - Programs compiled using the CMS/DOS assembler may have incorrect external symbol dictionaries. In this situation, the OS assembler is used. The OS assembler is *not* compatible with VSE.
    - Some VSE macros and SVC's are simulated. The code generated is not complete under CMS/DOS.
    - CMS handles interruptions in BC mode only, therefore, setting the PSW EC mode bit on is not recommended.

15. The maximum sizes (in cylinders or blocks) of CMS minidisks are as follows:

Device Type	Model(s)	CMS/VSAM	CMS 800-Byte Format	CMS 1K, 2K, or 4K Format
2314/2319	--	200 cyl	203 cyl	203 cyl
3310	--	126,016 blk	Not Supported	126,016 blk
3330	1 or 2	404 cyl	246 cyl	404 cyl
3330	11	808 cyl	246 cyl	808 cyl
3333	1	404 cyl	246 cyl	404 cyl
3333	11	808 cyl	246 cyl	808 cyl
3348	35	349 cyl	348 cyl	348 cyl
3348	70	696 cyl	682 cyl	696 cyl
3350	Native Mode	555 cyl	115 cyl	555 cyl
3370	A1 or B1	558,000 blk	Not Supported	558,000 blk
3370	A2 or B2	712,752 blk	Not Supported	712,752 blk
3375	--	959 cyl	182 cyl	959 cyl
3380	--	885 cyl	121 cyl	885 cyl
3380	AE4 or BE4	1770 cyl	121 cyl	1770 cyl

16. To ensure that the saved copy of the S-STAT or Y-STAT is current, a validity check is performed when a saved system is IPLed. This check is performed only for S-DISKS and Y-DISKS formatted in 512-, 1024-, 2048-, or 4096-byte CMS blocks. For 800-byte blocks disks, the saved copy of the S-STAT or Y-STAT is used. The validity checking consists of comparing the date when the saved directory was last updated with the date when the current disk was last updated. If the dates for the S-STAT are different, then the S-STAT is built in user storage. If the dates for the Y-STAT are different, then the Y-disk is accessed using the CMS ACCESS command: ACCESS 19E Y/S* * Y2.[14] This means that even when the S- and Y-disks are accessed in read/write mode and then RELEASED, the message DMSINS100W S-STAT and/or Y-STAT NOT AVAILABLE will result.

17. Programs that operate under CMS should use documented interfaces. Those programs which modify DMSNUC or other CMS control blocks in order to accomplish their interfaces with the CMS system may hamper the performance and reliability of the system.

18. Programs that modify the file id of an FST can destroy the integrity of the file system. CMS does not support such programs, which may cause a "file not found" condition for the file until the disk is released.

Virtual Device Restrictions

1. When you define virtual devices for a virtual machine, you must consider free storage requirements for virtual device tables. Virtual device tables must reside in contiguous free storage, and sometimes enough is not available.

   For Release 4.2, if you define more than 51 devices for a virtual machine, you may receive an error message because you need more than one page of contiguous free storage. The number "51" results from the size of the VDEVBLOK as it relates to a full page (512 doublewords) of free storage. Because the size of the VDEVBLOK varies from one release to the next, the device number will be different for each release. Consider the following to avoid the free storage problem:

   - You need 512 doublewords of contiguous free storage for every (512/VDEVBLOK size) virtual devices. (For Release 4.2, the size of the VDEVBLOK is 10 doublewords. 512 divided by 10 is 51.2. Thus, if you define more than 51 devices you need more than one page of contiguous free storage.

   - Virtual machines with more than 51 devices should be the first to log on after IPL. Contiguous pages of free storage are sure to be available only immediately after IPL, before other virtual machines have logged on.

---

[14] The DASD address of the Y-DISK will be whatever was specified when CMS was generated. For the standard system this will be 19E.

- If you want to define more than 410 virtual devices, see *VM/SP HPO Data Areas and Control Block Logic—CP* for the current size of the VDEVBLOK (virtual device block).

2. The MAXDEV operand of the OPTION control statement sets the maximum number of virtual devices that can be defined for a virtual machine. The greatest possible value for MAXDEV is

```
(32767/VDEVBLOK size in doublewords) + 1
```

with the result of the division rounded down to the next lowest whole number. The default value is 410. For a virtual machine running in BC mode, the maximum value of MAXDEV is 1536. If this limit is exceeded, unpredictable results may occur.

Notes:

a. 32767 is equal to X'7FFF'.

b. Since the current size of the VDEVBLOK is 10 doublewords, the maximum value of MAXDEV is 3277. For changes in the size of the virtual device block (VDEVBLOK), see VM/SP HPO Data Areas and Control Block Logic—CP.

Extended Channel Support Restrictions

1. If you use extended channel support, define devices to accommodate all SYSOWN volumes on channels 0 through F, so that your installation can run on hardware with or without extended channel support as well as in a virtual machine.

2. If your installation uses the extended channel support option in combination with directories from previous releases, make sure that all directory statements referring to real addresses use channel addresses between 0 and F.

3. When you specify multiple real device addresses or a range of addresses on the ATTACH command, the virtual device addresses must be the same as the real device addresses. Since virtual device addresses can be only three hexadecimal digits long, devices attached in this manner must be on channels 0 to F. Devices on real channels 10 to 1F must be attached one at a time.

4. Since the ATTACH CHANNEL command requires the virtual and real channel addresses to be the same; and since virtual addresses allow only one digit for the channel address, channels 10 through 1F cannot be attached to a virtual machine.

5. Channels 10 through 1F cannot be preferred channels.

Preferred Machine Assist Restrictions

Note: For restrictions on certain CP commands when preferred machine assist is active, see the VM/SP HPO System Programmer's Guide.

1. Preferred machine assist supports *only* MVS/SP Version 1 Release 1 with Release 1 Enhancement or later releases. MVS/SP Version 1 Release 3 or later releases are needed to support "extended storage" above 16 megabytes. Any attempt to run another system using the preferred machine assist feature may result in CP abending.

2. The MSS Central Server Application Program, a VM/370 Release 6 Program that runs under MVS, cannot run when preferred machine assist is active. This restriction does not apply to preferred machine assist with control switch assist.

3. Devices on channels that belong to the preferred virtual machine are not defined to VM/SP HPO at system generation. Because these devices are unknown to VM/SP HPO, the VM/SP HPO system operator cannot vary them online to VM/SP HPO without reinitializing the system and including the devices in DMKRIO.

4. Do not issue DIAGNOSE instructions when running a guest MVS system with preferred machine assist. The results of using a DIAGNOSE in this environment are unpredictable. For example, if a guest MVS system running with preferred machine assist issues a DIAGNOSE X'80' to vary a channel offline, the DIAGNOSE instruction will go directly to the hardware and the channel will be varied offline. VM/SP HPO is unable to intercept the DIAGNOSE when the MVS guest is running with preferred machine assist.

   You may issue DIAGNOSE instructions when the MVS guest is running with preferred machine assist with control switch assist.

5. You must not have CP-owned volumes and MVS volumes on the same strings and control units when you are:

   - Running an MVS guest when VM is in Single Processor Mode(SPMODE), on multi-processing systems

   - Running an MVS guest with Preferred Machine Assist(PMA) on uni-processors, or multi-processors

   In either of these two environments, a path must exist to the DASD volumes from a VM channel, and from a channel that VM does not know about. This can be either a channel from the MVS native processor (when running Single Processor Mode) or a preferred channel on the VM processor.

   When the above conditions are met, a deadlock situation can occur that will prevent the MVS guest from being dispatched if MVS and CP-owned volumes exist on the same strings and control units. This can happen when an I/O error occurs on the channel that VM does not know about. When this happens, a contingent connection is established

between the device, head of string, control unit, and the channel until MVS can issue a sense command to gather the error data. If, before the sense can be issued, VM must perform some service for the MVS guest, the lockout can occur. If CP must do some I/O to the CP owned volumes that share the same string or control unit that are busy because of the I/O error, MVS may be placed in a wait state, until the resulting busy condition has been cleared. Since only the MVS guest can clear the busy, by issuing the sense, a deadlock has occurred, and the MVS guest will never be dispatched.

The two types of CP services that can cause I/O to DASD devices are

- Use of VM spooling. This can be unit record spooling or console spooling.
- The issuing of CP commands at the MVS virtual machine console.

If you cannot avoid either of these activities, you must not have VM system volumes and MVS volumes sharing the same strings and control units in the configurations mentioned above.

There are some special considerations when using strings of 3380s that start with AA4s which are connected to two storage directors of the 3880 control units. Since these devices have multiple internal paths, and can have two concurrent data transfer operations taking place, these strings can be properly configured to avoid the deadlock situation, and still have MVS and VM volumes on the same string. To do this you must segregate the MVS and VM volumes on different internal paths. In a string of 16 devices the addresses are assigned to internal paths as follows:

PATH A - xx0,xx1,xx8,xx9

PATH B - xx2,xx3,xxA,xxB

PATH C - xx4,xx5,xxC,xxD

PATH D - xx6,xx7,xxE,xxF

If you can arrange your MVS and CP owned volumes so that they do not share an internal path, you can avoid these lockout situations.

6. If you started single processor mode on a 3033, 308x, 3090, or 4381-3 UP-generated system to IPL with the PMA or PMAV option, the MVS guest must vary off the native side before issuing SPMODE off.

7. If you are running single processor mode on a 308x, 3090, or 4381-3 processor and you IPL with the PMAV option, the MVS guest must issue DIAGNOSE instructions from the CP side (not the native side). In order to accomplish this, the MVS system programmer must establish affinity for the program issuing the DIAGNOSE instructions.

8. An MVS guest operating system using

   - Single Processor mode
   - Non-disruptive transition

   on the 308x or 3090 processor must use preferred machine assist or preferred machine assist with control switch assist.

9. An MVS guest operating system using

   - Preferred machine assist
   - Single processor mode
   - Nondisruptive transition

   on the 3090 processor must use the release of MVS currently supported on the 3090 processor (Release 1.3.5).

10. Since MVS/SP does not support the 3090 Vector Facility, the MVS preferred machine assist guest cannot use it.

11. V=R recovery, the SAVESYS command, and the VMSAVE option on the OPTION directory control statement do not save vector status or vector registers for virtual machines.

Miscellaneous Restrictions

Restrictions on VM in General

1. CP checks any attempts to construct a virtual device configuration that would mix SHARED and NONSHARED device types on the same virtual control unit. If a violation is encountered, CP rejects the offending command and sends an error message to the user who issued it. This restriction affects the following situations:

   - CP directory updates
   - Logon processing
   - Certain CP console functions

   As the CP directory source file is processed, those statements that describe virtual devices are checked for a subchannel protocol conflict. If a conflict is detected, CP sends an error message. Directory processing continues, but the directory cylinders are not updated. Only the effects of the T-DISK options of the MDISK statement and the CONSOLE, SPECIAL, and SPOOL statements are considered during the CP directory update.

When the user logs on to the system, CP checks the devices represented by each CONSOLE, DEDICATE, LINK, MDISK, SPECIAL, and SPOOL statement. If CP finds a violation, it rejects the statement and sends an error message.

When the CP console commands ATTACH, DEFINE, LINK, or NETWORK ATTACH are issued to create or move a virtual device, CP checks to see if a conflict could occur on the virtual control unit that would support the new device. If a conflict is detected, the operation is not performed and CP sends the user an error message.

In the following example, a user encounters the subchannel protocol restriction.

```
USER: cp define graf as 4A0 3270
CP:   GRAF 4A0 DEFINED
USER: cp define printer as 4A8 1403
CP:   DMKDEF331E  4A8 NOT DEFINED - USE A NONSHARED VCU INSTEAD
```

In the preceding example, the 3270 is a shared device and the 1403 printer is nonshared. CP found a protocol conflict and sent the user an error message.

Refer to Appendix B, "Configuration Aid" on page 391 to become familiar with the different virtual devices that are available.

2. Logical device support is not designed to simulate all aspects of real device support. Some instances are:

   - Logical device support always presents the channel status words, "Channel End" and "Device End," to the virtual machine at the same time.
   - Logical device support does not manipulate the PCI bit in the CCW.
   - Only the I/O ending status is returned to the virtual machine, never the initial status.

   CP does not support a write for position CCW chained to another write CCW. If this situation does occur, it will be treated as two chained write CCWs.

3. VM/SP HPO does not support the VM/System Product Small CP Option.

4. If a terminal has an inhibited (nondisplay) read up and either a delayed PF key or an undefined PF key is used, the input area will be rewritten without the inhibited attribute byte. Any data typed in at this point is displayed. To rewrite the inhibited read, use the clear key after using the PF key.

5. Do not use volume identifiers for SYSRES or CP-owned volumes that are the same as volume IDs specified for dedicated volumes or minidisk volumes.

6. The total number of area specifications in all the SYSPAG macros cannot exceed 255.

7. You can specify a given volid only once on the same SYSPAG macro. You can specify a given volid on different SYSPAG macros.

8. Areas that you specify as TYPE=PS on a SYSPAG macro must be areas allocated as TEMP by DMKFMT.

9. Areas that you specify as TYPE=DU on a SYSPAG macro must be areas allocated as DUMP by DMKFMT.

10. After you change the area specifications on the SYSPAG macro, you must do a cold start, not a warm start or CKPT start.

11. If a virtual machine issues a CLEAR CHANNEL to any dedicated channel, the results will be unpredictable.

12. When a real channel uses a hardware LIFO stacking mechanism, missing interrupts may occur at the control unit level. Because the hardware LIFO mechanism stacks the interrupts in the channel, the channel may present the control unit end interrupt to the host *before* it presents the control unit busy interrupt. Use the missing interrupt handler to relieve this situation.

13. When VM/SP HPO is installed, the system cannot be shutdown by an IPL of a new nucleus.

14. In order to defer error processing, VM uses the stacking of CPEXBLOKs to device-dependent error recovery modules. Subsequent interrupts may occur before error processing is completed for the first interrupt. As a result, two CPEXBLOKs may be stacked to the ERP module for the same IOBLOK.

15. When you generate a new nucleus and change unit record device configurations in DMKRIO, be sure to drain all unit record devices before shutting down the system. If you do not drain the unit record devices, you will get unpredictable results when you IPL the new system.

Restrictions Relating to I/O

1. The number of pages used for input/output must not exceed the total number of user pages available in real storage. Violation of this restriction causes the real computing system to be put into an enabled wait state.

2. If an I/O device (such as a disk or tape drive) drops ready status while it is processing virtual I/O activity, any virtual machine users performing I/O on that device are unable to continue processing or to log off. Also, the CP commands LOGOFF and FORCE are not effective because they do not complete until all outstanding I/O is finished. The system operator should determine which I/O device is involved and make that device ready once more.

3. If the system prematurely terminates the Input/Output Configuration Program (IOCP) while an input/output configuration data set write is in process, the CP IOCP write lock will remain set. To clear the lock, do one of the following:

   - Rerun the IOCP to completion
   - Perform a virtual system restart
   - Re-IPL the virtual machine

4. When sending a large graphics data stream, a user may incur a performance degradation.

5. Do not generate CP paging packs on Channel 0.

Restrictions on Specific Devices

1. For remote 3270's, VM/SP HPO supports a maximum of 256 binary synchronous lines, minus one (if there are any 3704/3705/3725 Communications Controllers in emulation mode).

2. If an installation is using an IBM 303x processor, it must dedicate the service record file (SRF) device to the control program. Thus, the channel on which the SRF is located cannot be dedicated to any virtual machine.

3. When using two channel-to-channel adapters dedicated to virtual machines and the CTCAs are operating on the same channels on each processor, the virtual machines should use the control CCW to prevent locking out the channel.

4. The SEREP program is not supported as a service strategy on the 4300 processors or the 308x or 3090 Processor Complexes.

5. Once a 3278 Model 4 has switched to alternate mode (43-line screen), the terminal then dials to a virtual machine, it must be reset to default mode (24-line screen). The 3278 Model 4 remains in alternate mode, which is initiated after the logo has been written and an erase write alternate has been issued to the screen.

6. Support of the 3800 as a nondedicated virtual spooling device does not save the loaded FCB between spool files. When a virtual spool file is closed, the loaded FCB is not kept. When a spool file is opened, a default FCB is set. A X'63' (LOAD FCB) CCW must be issued to set up any other FCB for the spool file. This support is inconsistent with VM/SP HPO support of other virtual spooling devices, such as the 3211.

7. When using the virtual channel-to-channel adapter it is possible to receive a spurious attention interrupt after receiving *attention* plus *busy* in response to a data transfer operation. The spurious attention may occur if both the X and Y sides of the *VCTCA* are doing the same data transfer operation (for example, both doing writes or both doing reads.).

8. The following devices cannot be used for swapping:

   - 3310 or 3370 FBA devices
   - 3375 or 3380 devices equipped with Extended Count-key-data support that are not attached to a high speed channel
   - 2314 or 3340 disk devices

9. If a printer is defined as a 4248 during system generation, it must also be defined as a 4248 during IML (Initial Microprocessor Load). If this consistency is not maintained, spool files may be lost.

10. CMS support of the 3290 Information Panel is limited to the XEDIT function.

    The CP function requires a minimum screen size of 3 rows by 50 columns. If you define a screen size smaller than this minimum, your screen will become the default size of 24 rows by 80 columns. The CMS full screen function requires a minimum screen size of 24 rows by 80 columns. Logical screen sizes ranging from the minimum up to 62 rows by 160 columns (the maximum for the 3290 Information Panel) are acceptable for CMS.

    Many program products which run under VM/SP require running in 3278 compatibility mode. For these program products, you must use the screen size associated with the model number defined during system generation. If you are using one of these program products and you use a different screen size, unpredictable results may occur. The 3278 model numbers and associated screen sizes are:

Model 2A or 2C	20 rows by 80 columns
Model 2	24 rows by 80 columns
Model 3	32 rows by 80 columns
Model 4	43 rows by 80 columns
Model 5	27 rows by 132 columns

    In the XEDIT environment, if the screen size is defined to be larger than 150 columns, the copy key will work only if the virtual printer is defined as 3800.

    Users of the 3290 Information Panel should be aware that there are maximum lengths for input and output in VM/SP. In some cases, the command line of the 3290 allows more input characters than the maximum length of a command. In order to avoid losing part of a command due to truncation, you should make sure that your commands do not exceed the maximum lengths. The following is a list of input and output maximum lengths in characters.

    CMS Environment:

Input command line	130
Output	130

XEDIT Environment:

XEDIT subcommand from command line	255
Input following the logical line end character when in TYPEWRITER mode	130
CP commands	240
RETRIEVE or ? (Redisplays the last input line or lines up to the maximum number of characters)	255

CP Environment:

CP commands	240
RETRIEVE or ? (Redisplays the last input line or lines up to the maximum number of characters)	240

11. The allocation of paging and spooling areas must be on pseudo-cylinder boundaries for FBA devices. When an allocation boundary is specified within a pseudo-cylinder boundary, the cylinder will not be used. A method of conversion for pages (blocks) to pseudo-cylinders for FBA devices is provided in "FBA Page/Pseudo-Cylinder Conversion" on page 37.

12. When you migrate from a VM/SP HPO Release 3.2 system or earlier to a Release 3.4 system or later, if you have spool files on FBA devices, a warmstart is not permitted. The recommended procedure for recovering spool files is to do a checkpoint start and issue the SPTAPE command.

13. The 3380 DASD Model AE4/BE4 may exceed the capacity of the current file system (having 1K, 2K, or 4K blocksizes). If this DASD is established as one minidisk and one-byte records are created, then the fullword limit for fields in the file system will be exceeded.

Restrictions on Commands or Features

1. Any modifications to local OPTIONS COPYFILE, unless otherwise specified in existing documentation, is not supported.

2. When TERMINAL CONMODE 3270 is invoked, tracing should not be performed at the same console; the results are unpredictable.

3. If you use CONMODE 3270 with a guest system control program such as MVS, you must specify SCRNSAVE ON. If you don't, the results are unpredictable.

4. If a NETWORK ENABLE is issued to a device with advanced features and a NETWORK ATTACH is issued prior to powering the device on, then the advanced features will be non-operational. The device must be powered on and enabled prior to the NETWORK ATTACH.

5. When you are running either in single processor mode or as a preferred machine assist guest, do not use the PER command, because MVS has control of real registers 9, 10, and 11.

6. Users of MVS Release 2 or 3 with the cross-memory feature (#6850) installed cannot use the single-processor mode (SPM) or nondisruptive transition (QVM) functions of VM/SP HPO.

7. You must not have CP-owned volumes and MVS volumes on the same strings and control units when you are:

   - Running an MVS guest when VM is in Single Processor Mode(SPMODE), on multi-processing systems

   - Running an MVS guest with Preferred Machine Assist(PMA) on uni-processors, or multi-processors

   In either of these two environments, a path must exist to the DASD volumes from a VM channel, and from a channel that VM does not know about. This can be either a channel from the MVS native processor (when running Single Processor Mode) or a preferred channel on the VM processor.

   When the above conditions are met, a deadlock situation can occur that will prevent the MVS guest from being dispatched if MVS and CP-owned volumes exist on the same strings and control units. This can happen when an I/O error occurs on the channel that VM does not know about. When this happens, a contingent connection is established between the device, head of string, control unit, and the channel until MVS can issue a sense command to gather the error data. If, before the sense can be issued, VM must perform some service for the MVS guest, the lockout can occur. If CP must do some I/O to the CP owned volumes that share the same string or control unit that are busy because of the I/O error, MVS may be placed in a wait state, until the resulting busy condition has been cleared. Since only the MVS guest can clear the busy, by issuing the sense, a deadlock has occurred, and the MVS guest will never be dispatched.

   The two types of CP services that can cause I/O to DASD devices are:

   - Use of VM spooling. This can be unit record spooling or console spooling.

   - The issuing of CP commands at the MVS virtual machine console.

   If you cannot avoid either of these activities, you must not have VM system volumes and MVS volumes sharing the same strings and control units in the configurations mentioned above.

There are some special considerations when using strings of 3380s that start with AA4s which are connected to two storage directors of the 3880 control units. Since these devices have multiple internal paths, and can have two concurrent data transfer operations taking place, these strings can be properly configured to avoid the deadlock situation, and still have MVS and VM volumes on the same string. To do this you must segregate the MVS and VM volumes on different internal paths. In a string of 16 devices the addresses are assigned to internal paths as follows:

PATH A - xx0,xx1,xx8,xx9

PATH B - xx2,xx3,xxA,xxB

PATH C - xx4,xx5,xxC,xxD

PATH D - xx6,xx7,xxE,xxF

If you can arrange your MVS and CP owned volumes so that they do not share an internal path, you can avoid these lockout situations.

8. Some processors require certain levels of MVS/SP in order to use certain functions:

Processor	Function	MVS/SP Release Required
3090	• Preferred machine assist • Preferred machine assist with the control switch assist • Use of storage above 16 Mb by MVS/SP preferred guest • Use of single processor mode by MVS/SP preferred guest • Use of dynamic SCP transition by MVS/SP preferred guest	MVS/SP Version 1 Release 3 Modification 5
4381 308x	• Preferred machine assist • Preferred machine assist with the control switch assist • Single processor mode • Dynamic SCP transition	MVS/SP Release 1 Enhancement
	• Use of storage above 16 Mb by MVS/SP preferred guest • Use of single processor mode by MVS/SP preferred guest • Use of dynamic SCP transition by MVS/SP preferred guest	MVS/SP Release 3
3033	• Preferred machine assist	MVS/SP Release 1 Enhancement
	• Use of storage above 16 Mb by MVS/SP guest • Single processor mode • Dynamic SCP transition	MVS/SP Release 3

Appendix E. Worksheet to Aid in Coding the NAMESYS Macro

Seg	Page	Hex Address	Name	Seg	Page	Hex Address	Name
0	0	0		36	576	240000	
1	16	10000		37	592	250000	
2	32	20000		38	608	260000	
3	48	30000		39	624	270000	
4	64	40000		40	640	280000	
5	80	50000		41	656	290000	
6	96	60000		42	672	2A0000	
7	112	70000		43	688	2B0000	
8	128	80000		44	704	2C0000	
9	144	90000		45	720	2D0000	
10	160	A0000		46	736	2E0000	
11	176	B0000		47	752	2F0000	
12	192	C0000		48	768	300000	
13	208	D0000		49	784	310000	
14	224	E0000		50	800	320000	
15	240	F0000		51	816	330000	
16	256	100000		52	832	340000	
17	272	110000		53	848	350000	
18	288	120000		54	864	360000	
19	304	130000		55	880	370000	
20	320	140000		56	896	380000	
21	336	150000		57	912	390000	
22	352	160000		58	928	3A0000	
23	368	170000		59	944	3B0000	
24	384	180000		60	960	3C0000	
25	400	190000		61	976	3D0000	
26	416	1A0000		62	992	3E0000	
27	432	1B0000		63	1008	3F0000	
28	448	1C0000		64	1024	400000	
29	464	1D0000		65	1040	410000	
30	480	1E0000		66	1056	420000	
31	496	1F0000		67	1072	430000	
32	512	200000		68	1088	440000	
33	528	210000		69	1104	450000	
34	544	220000		70	1120	460000	
35	560	230000		71	1136	470000	

Seg	Page	Hex Address	Name	Seg	Page	Hex Address	Name
72	1152	480000		108	1728	6C0000	
73	1168	490000		109	1744	6D0000	
74	1184	4A0000		110	1760	6E0000	
75	1200	4B0000		111	1776	6F0000	
76	1216	4C0000		112	1792	700000	
77	1232	4D0000		113	1808	710000	
78	1248	4E0000		114	1824	720000	
79	1264	4F0000		115	1840	730000	
80	1280	500000		116	1856	740000	
81	1296	510000		117	1872	750000	
82	1312	520000		118	1888	760000	
83	1328	530000		119	1904	770000	
84	1344	540000		120	1920	780000	
85	1360	550000		121	1936	790000	
86	1376	560000		122	1952	7A0000	
87	1392	570000		123	1968	7B0000	
88	1408	580000		124	1984	7C0000	
89	1424	590000		125	2000	7D0000	
90	1440	5A0000		126	2016	7E0000	
91	1456	5B0000		127	2032	7F0000	
92	1472	5C0000		128	2048	800000	
93	1488	5D0000		129	2064	810000	
94	1504	5E0000		130	2080	820000	
95	1520	5F0000		131	2096	830000	
96	1536	600000		132	2112	840000	
97	1552	610000		133	2128	850000	
98	1568	620000		134	2144	860000	
99	1584	630000		135	2160	870000	
100	1600	640000		136	2176	880000	
101	1616	650000		137	2192	890000	
102	1632	660000		138	2208	8A0000	
103	1648	670000		139	2224	8B0000	
104	1664	680000		140	2240	8C0000	
105	1680	690000		141	2256	8D0000	
106	1696	6A0000		142	2272	8E0000	
107	1712	6B0000		143	2288	8F0000	

Seg	Page	Hex Address	Name	Seg	Page	Hex Address	Name
144	2304	900000		180	2880	B40000	
145	2320	910000		181	2896	B50000	
146	2336	920000		182	2912	B60000	
147	2352	930000		183	2928	B70000	
148	2368	940000		184	2944	B80000	
149	2384	950000		185	2960	B90000	
150	2400	960000		186	2976	BA0000	
151	2416	970000		187	2992	BB0000	
152	2432	980000		188	3008	BC0000	
153	2448	990000		189	3024	BD0000	
154	2464	9A0000		190	3040	BE0000	
155	2480	9B0000		191	3056	BF0000	
156	2496	9C0000		192	3072	C00000	
157	2512	9D0000		193	3088	C10000	
158	2528	9E0000		194	3104	C20000	
159	2544	9F0000		195	3120	C30000	
160	2560	A00000		196	3136	C40000	
161	2576	A10000		197	3152	C50000	
162	2592	A20000		198	3168	C60000	
163	2608	A30000		199	3184	C70000	
164	2624	A40000		200	3200	C80000	
165	2640	A50000		201	3216	C90000	
166	2656	A60000		202	3232	CA0000	
167	2672	A70000		203	3248	CB0000	
168	2688	A80000		204	3264	CC0000	
169	2704	A90000		205	3280	CD0000	
170	2720	AA0000		206	3296	CE0000	
171	2736	AB0000		207	3312	CF0000	
172	2752	AC0000		208	3328	D00000	
173	2768	AD0000		209	3344	D10000	
174	2784	AE0000		210	3360	D20000	
175	2800	AF0000		211	3376	D30000	
176	2816	B00000		212	3392	D40000	
177	2832	B10000		213	3408	D50000	
178	2848	B20000		214	3424	D60000	
179	2864	B30000		215	3440	D70000	

Seg	Page	Hex Address	Name	Seg	Page	Hex Address	Name
216	3456	D80000		237	3792	ED0000	
217	3472	D90000		238	3808	EE0000	
218	3488	DA0000		239	3824	EF0000	
219	3504	DB0000		240	3840	F00000	
220	3520	DC0000		241	3856	F10000	
221	3536	DD0000		242	3872	F20000	
222	3552	DE0000		243	3888	F30000	
223	3568	DF0000		244	3904	F40000	
224	3584	E00000		245	3920	F50000	
225	3600	E10000		246	3936	F60000	
226	3616	E20000		247	3952	F70000	
227	3632	E30000		248	3968	F80000	
228	3648	E40000		249	3984	F90000	
229	3664	E50000		250	4000	FA0000	
230	3680	E60000		251	4016	FB0000	
231	3696	E70000		252	4032	FC0000	
232	3712	E80000		253	4048	FD0000	
233	3728	E90000		254	4064	FE0000	
234	3744	EA0000		255	4080	FF0000	
235	3760	EB0000		256	4096	1000000	
236	3776	EC0000					

Appendix F. Sample SNTMAP Output

DASD SNTMAP

```
 S A V E D   S E G M E N T   D A S D   L A Y O U T

Volume Segment   Segment   Start     End       Number
Label  Name      Location  Cyl/Page  Cyl/Page  Of Pages
------ --------  --------  --------  --------  --------

PROFPK - 3350    (*** No SAVSYS area defined ***)
       OFSSEG    083-092    0001,001 0002,073    193

VMPK01 - 3350    (*** No SAVSYS area defined ***)
       APLSYS1   058-063    0001,001 0001,097     97
       APLSYS2   074-079    0001,098 0002,074     97
       APLSYS3   107-112    0002,075 0003,051     97
       APLSYS4   234-239    0003,052 0004,028     97
       BASSEG    096-101    0004,029 0005,005     97
       BLISEG    102-104    0005,006 0005,054     49
     ** GAP **                                   241
       ADMASS30  146-173    0007,056 0011,024    449

VMSRES - 3350    (*** No SAVSYS area defined ***)
       CMS       025-031    0001,001 0002,018    138
       CMSL      240-246    0002,019 0003,036    138
       CMSM      242-246    0002,021 0003,038    138 DASD Overlap
       CMSDOS    049        0003,037 0003,053     17 DASD Overlap
       CMSBAM    050-052    0003,054 0003,102     49
       CMSVSAM   033-038    0003,103 0004,095    113
       CMSAMS    040-045    0004,096 0005,120    145
       INSTVSAM  250        0006,001 0006,017     17
       VMEP01    (NONE)     0006,018 0006,034     17
       VMEP02    (NONE)     0006,035 0006,051     17
       DSMSEG3   067-072    0006,052 0007,028     97
       DTRSEG    073        0007,029 0007,045     17
       ISPDCSS   128-133    0007,046 0008,022     97
       ISRDCSS   134-139    0008,023 0008,119     97
       IISDCSS   054-057    0008,120 0009,064     65
```

DASD Overlap messages show an ERROR condition.

```
FREE DASD SPACE - SAVSYS GAP SUMMARY

Volume DASD      Start      End        Number
Label  Type      Cyl/Page   Cyl/Page   Of Pages
------ --------  --------   --------   --------
VMPK01 3350      0005,055   0007,055     241
```

MEMORY SNTMAP

```
V I R T U A L   M E M O R Y   M A P

Volume Segment  Segment   Number of
Label  Name     Location  Segments
------ -------- --------  ---------
VMSRES CMS      025-031       7
VMSRES CMSVSAM  033-038       6
VMSRES CMSAMS   040-045       6
VMSRES CMSDOS   049           1
VMSRES CMSBAM   050-052       3
VMSRES IISDCSS  054-057       4
VMPK01 APLSYS1  058-063       6
VMSRES DSMSEG3  067-072       6
VMSRES DTRSEG   073           1
VMPK01 APLSYS2  074-079       6
PROFPK OFSSEG   083-092      10
VMPK01 BASSEG   096-101       6
VMPK01 BLISEG   102-104       3
VMPK01 APLSYS3  107-112       6
VMSRES ISPDCSS  128-133       6
VMSRES ISRDCSS  134-139       6
VMPK01 ADMASS30 146-173      28
VMPK01 APLSYS4  234-239       6
VMSRES CMSL     240-246       7
VMSRES CMSM     242-246       5 Segment Overlay
VMSRES INSTVSAM 250           1

-----------------------------------
--------- FAMILY 'A' ONLY ---------
-----------------------------------
VMSRES CMSVSAM  033-038       6
VMSRES CMSBAM   050-052       3
VMSRES CMSL     240-246       7
VMSRES CMSM     242-246       5 Segment Overlay

-----------------------------------
--------- FAMILY 'B' ONLY ---------
-----------------------------------
VMSRES CMSVSAM  033-038       6
VMSRES CMSDOS   049           1
VMSRES CMSL     240-246       7
VMSRES CMSM     242-246       5 Segment Overlay

-----------------------------------
--------- FAMILY 'C' ONLY ---------
-----------------------------------
VMSRES CMSVSAM  033-038       6
VMSRES CMSDOS   049           1
VMSRES CMSM     242-246       5

-------------------------------------------------------
```

Note 1: The following macros contain unshared pages:

Segment Name	Shared Segments	Actual Page Allocation
VMEP01	(NONE)	4000-4015
VMEP02	(NONE)	4000-4015
CMS	025-031	0-4,14-33,400-511
CMSVSAM	033-038	528-639
CMSAMS	040-045	640-783
OFSSEG	083-092	1328-1519
CMSL	240-246	0-4,14-33,3840-3951
CMSM	242-246	0-4,14-33,3840-3951

Note 2: Segment overlays do not work in the same virtual machine.

Index

A

A-disk
  CMS primary user disk 44
    accessing 51
ACCESS
  command (CMS)
    use in file sharing 51
  filemodes
    CMS files 51
  modes
    defining for minidisks 213
access control groups 175
  defining 198
Access Method Services
  planning considerations 75
  storage requirements 39
ACCOUNT
  directory control statement 197
accounting
  number
    defining in directory 197
ACCT option
  defining in directory 202
ACIGROUP
  directory control statement 198
ADAPTER operand
  of RDEVICE macro 261
ADDRESS operand
  RCHANNEL macro 282
  RCTLUNIT macro 278
  RDEVICE macro 254
AFFINITY option
  defining in directory 203
allocating
  DASD space for the directory 183
  DASD space on CP-owned volumes 313
ALTCH operand
  RCTLUNIT macro 278
ALTCONS operand
  RIOGEN macro 284
ALTCU operand
  RDEVICE macro 262
alternate blocks
  FB-512 disks 64
alternate console
  defining in RIOGEN macro 284
alternate path support
  restrictions 89
  supported switches 88
  two-channel switch 88
alternate tracks
  minidisks 61
  system residence devices 63
AP operand
  SYSCOR macro 343
assembler
  VM/370 377
assigning more than 8 privilege classes 196
attached processor system
  generating 135
  load list 31
  modules 135
  performance measurement 112
  specifying AP initialization
    SYSCOR macro 343
  system identification
    SYSID macro 356
  System/370 Extended Feature 125
attachments
  3270s
    local 145
    remote 137
AUTO operand
  SYSMON macro 337
auxiliary storage
  required by CMS 44
available real storage
  calculating 118
  Formula 1 118

B

BASEADD operand
  RDEVICE macro 264
blocks
  FB-512
    alternate 64
  minidisks
    alternate 62
BMX option
  defining in directory 202
BOTTOM operand
  SYSPCLAS macro 354
BSC lines
  CLUSTER macro 245
  coding
    DMKRIO 243
  coding RDEVICE macro 142
  3270 support 138
BUFFS operand
  SYSMON macro 337

C

cached devices
  general paging 317
  page migration 317
  preferred paging 317
cardless system
  required devices 7
CCW translation
  reserve/release 103
changing the class structure 234
channel
  alternate
    RCTLUNIT macro 281
  errors
    RCTLUNIT macro 281
  restrictions 401
channel index table
  generating 282
Channel Indirect Data Addressing feature 11
channel interface
  Mass Storage Control 160
  positions for Staging Adapter 160
channel model dependent functions
  restrictions 402
channel programs 395
channel switching
  between two processors 85
  on one processor 86
  system generation requirements 85
  tape 87
  two- or four-channel switch feature 85
CHAR operand
  RDEVICE macro 264
checkpoint cylinder
  number of cylinders required by device type 333
checkpoint start
  data
    calculating number of cylinders needed 333
    DASD storage requirements 36
    defining cylinders 333
CHTYPE operand
  RCHANNEL macro 282
class
  classifying printed output
    SYSPCLAS macro 354
CLASS control statement 196, 239
CLASS operand
  RDEVICE macro 260
  SYSACNT macro 348
  SYSMON macro 337
class-override file 236
  validating 235
class structure
  See user class structure
cluster
  remote attachments
    3270s 137
CLUSTER macro
  coding real I/O configuration 245
  CUTYPE operand 245
  DIAL operand 245
  examples 245
  format 245
  GPOLL operand 245
  label requirements 243
  LINE operand 245
CLUSTER operand
  RDEVICE macro 264
CMS (Conversational Monitor System)
  A-disk 44
  assembler
    use with CMS 47
  auxiliary storage requirements 44
  capacity of virtual disks 51
  CMS 53
  CMS/DOS environment restrictions 49
  command language 47
  communication intents for CMS 47
  default device addresses 44
  devices supported 44
  DIRECT command 186
  disk and file management
    CMS 52
    OS/DOS 51
    VSAM 51
  disks
    access 48
    capacity 53
    formatting 52
    labels 52
    linking 52
  executable program products 377
  FB-512 blocks 53
  file directory 53
  files
    format 52
    identification 54
    maximum usable number 53
    sharing 51
  formatted disks volume label contents 52
  introduction 3
  invoking the directory program 186
  limited support of OS and VSE 49
  minidisk labels 65
  minimum configurations 8
  nucleus
    storage requirements 43
  partitioned data sets 50
  planning considerations 44
  program languages supported under CMS 48
  records
    maximum usable number per file 53
  saving the CMS system 55
  sharing the system residence volume 82
  simulated partitioned data sets 45
  support of DL/I 50

symbolic names for devices 44
system disk
S-disk 44
system libraries
macro 47
text 47
tape support 54
unit record support 44, 55
virtual storage requirements 43
CMS/DOS
directory entries 230
label information area 77
planning considerations 75
tape handling 77
VSE system generation considerations 75
when VSE system must be online 76
COBOL
compiling programs under CMS 48
color
defining
SCREEN directory control statement 209
communication facility
virtual machine
IUCV 107
virtual machines
VMCF 109
compilers
supported for CMS/VSAM 72
components
CMS (Conversational Monitor System) 3
CP (Control Program) 3
introduction 3
configuration
aid 383
configurations
other considerations for planning 27
remote spooling 24
supported by VM/SP HPO 8
TCUs
3704/3705/3725 required features 23
CONNECT
authorizing
IUCV directory control statement 206
CONS operand
RIOGEN macro 284
console
alternate
defining in RIOGEN macro 284
defining real system console 284
directory control statement 211
supported by VM/SP HPO 14
control blocks
DMKRIO
defining 241
control files
attached processor system 135
multiprocessor system 136
control program functions
changing restricted access 361
control statements
See also directory control statements
CLASS 239
DESTINATION 236, 237
for UCR 239
OVERRIDE 236, 238
USER 239
control units
DASD
supported by VM/SP HPO 12
magnetic tape control units supported by VM/SP HPO 13
unit record control units supported by VM/SP HPO 14
CONVIPCS EXEC 68
CORTABLE
defining in DMKSYS 343
count-key-data devices
characteristics 33
CMS block 53
CP (Control Program)
checkpoint start data
DASD storage requirements 36
directory
DASD storage requirements 36
disk access 52
dump space
DASD requirements 37
error recording
DASD storage requirements 36
free storage requirements 29
introduction 3
minimum configurations 7
nucleus
DASD storage requirements 36
reducing its size 33
reducing its size, excluding SNA CCS 155
paging
DASD requirements 39
real control blocks storage requirements 29
real storage requirements
example 30
resident nucleus storage requirements 29
saved systems
DASD requirements 38
Special Message Facility 109
spooling
DASD requirements 39
trace table storage requirements 29
warm start data
DASD storage requirements 36
CP assist
description 122
CP commands
OVERRIDE 235
QUERY 231
CPNAME operand
NAMENCP macro 301
NAME3800 macro 303
RDEVICE macro 263
CPRD operand

SYSFCN macro 361
CPSIZE operand
NAMENCP macro 301
CPTYPE operand
NAMENCP macro 302
RDEVICE macro 262
CPUID option
defining in directory 202
CPWT operand
SYSFCN macro 361
cross memory services (MVS) 126
CTCA
coding RDEVICE macro 255
CUTYPE operand
CLUSTER macro 245
RCTLUNIT macro 278
cylinder assignments
3340 62

D

DASD (Direct Access Storage Device)
control units supported by VM/SP HPO 12
error recording space requirements 37
sharing
reserve/release support 100
space
allocating for the directory 183
allocating on FB-512 volumes 34
calculating for saved systems 121
formatting for the directory 183
needed by CMS 43
reserving for 3704/3705 control program image 150
storage
for CMS minidisks 41
required by CP nucleus 33
required for Access Method Services 39
required for CMS/VSAM 39
supported by VM/SP HPO 11
SYSOWN macro
allocating on CP-owned volumes 313
SYSRES macro
calculating cylinders for checkpoint start data 333
calculating cylinders needed for warm start data 335
DASD (Direct Access Storage Devices)
configuration aid 383
DASD operand
SYSMIH macro 358
DASTAP
data collection class 337
data collection
defining in SYSMON macro 337
performance measurement and analysis 112
data set compatibility
CMS/VSAM 72
data transfer
logical device support facility 106
DCSS
See discontiguous saved segments
DEDICATE
directory control statement
example 221
preferred machine assist 221
DEFAULT operand
SYSID macro 356
DEFCON operand
SYSFORM macro 352
defective tracks
2314/2319 disks 63
DEFINE
command (CP)
use in file sharing and access 51
defining minidisks 213
defining more than 8 privilege classes 196
defining your system
introduction 169
DEFPRT operand
SYSFORM macro 352
DEFPUN operand
SYSFORM macro 352
DESTINATION control statement 236, 237
Device Support Facilities
assigning alternate tracks for minidisks 61
use with OS and VSE minidisks 61
3380
minidisk allocation 60
devices
channel switching 85
coding RDEVICE macro
DEVTYPE 255
system console 255
unsupported devices 260
compatible with RSCS 393
configuration aid 383
configuring 383
DASD supported by VM/SP HPO 11
dedicating to virtual machines 221
default addresses for CMS 44
hardware
supported by CMS VSAM 71
linking at logon 225
magnetic tape supported by VM/SP HPO 13
processors supported by VM/SP HPO 10
real devices supported by VM/SP HPO 9
remote spooling 24
required for cardless system 7
sample configuration 286
simulated by programming 228
simulated I/O
specifying 228
special
configuration aid 383
subclass
defining unsupported devices 260

supported by CMS 44
terminals supported by VM/SP HPO 14
unit record devices supported by VM/SP HPO 14
unsupported
coding RDEVICE macro 257
unsupported devices dedicated to virtual machines 26
DEVTYPE operand
of RDEVICE macro 255
DFLT operand
SYSFCN macro 361
DIAGNOSE instruction
invoking logical device facility 106
DIAL operand
CLUSTER macro 245
DIRECT command
format 186
directory 234, 239
ACCOUNT control statement 197
ACCT option 202
ACIGROUP control statement 198
AFFINITY option 203
allocating DASD space 183
BMX option 202
CMS file directory 53
considerations for preparing entries 174, 177
CONSOLE control statement 211
control statements for creating 190
CPUID option 202
DASD storage requirements 36
DEDICATE control statement 221
defining access control groups 198
defining accounting number 197
defining distribution code 197
defining volume to contain 190
DIRECTORY control statement 190
ECMODE option 199
entries for CMS/DOS 230
example
a virtual machine for updating VMSYS 180
maintenance virtual machine, hardware 182
operating systems in a virtual machine, system dump 179
formatting DASD space 183
hardware support 178
INCLUDE control statement 195
IPL control statement 208
ISAM option 200
IUCV control statement 206
LINK control statement 225
MAXCONN option 204
MDISK control statement 213
MDISK statements 65
minidisks
overlapping extents 174
operator directory entry
example 179
OPTION control statement 199
other system virtual machines 180
permitted number of users 173
PMA option 200
PROFILE control statement 191
REALTIMER option 199
requirements for changing 186
RSCS requirements 83
SCREEN control statement 209
software support 178
SPECIAL control statement 228
SPOOL control statement 218
supplied with starter systems 182
SVCOFF option 202
system support virtual machines 177
USER control statement 192
VIRT=REAL option 200
XMEM option 204
370E option 204
directory capabilities, defined 170
directory control statements
ACCOUNT 197
ACIGROUP 198
CLASS 196
CONSOLE 211
DEDICATE 221
INCLUDE 195
IPL 208
IUCV 206
LINK 225
list of 188
MDISK 213
OPTION 199
PROFILE 191
SCREEN 209
SPECIAL 228
SPOOL 218
USER 190, 192
directory profile
creating 191
description 174
example 175
invoking 195
directory program
invoking under CMS 186
running standalone 187
discontiguous saved segments
defining 367-373
description 367
installing 368-369
SNTMAP EXEC 368
saving
errors 369
security 373
disks
CMS
access 48
formatting for CMS 52
labels
CMS 52
management
CMS 48

OS/DOS 51
VSAM 51
display devices
configuration aid 383
distribution code
defining in the directory 197
DL/I
support in CMS/DOS environment 50
DMKDIR
directory program
invoking under CMS 186
DMKOVR 235
DMKRIO
example of VM/SP HPO system 286
file definitions
introduction 4
preparing for system generation 241
RCHANNEL macro 282
RCTLUNIT macro 277
RDEVICE macro 253
RIOGEN macro 284
sample file 287
sequence of macros 241
TERMINAL macro 247
3270
example assemble file 144
DMKSNT
file definitions
introduction 5
NAMENCP macro 301
NAMESYS macro 296
NAME3800 macro 303
DMKSYS
file definitions
introduction 5
performance considerations for coding 311
preparing system control file 305
supplied with starter systems 306
SYSCOR macro 343
SYSFCN macro 361
SYSFORM macro 352
SYSID macro 356
SYSJRL macro 340
SYSLOCS macro 363
SYSMIH macro 106, 358
SYSMON macro 337
SYSOPR macro 347
SYSOWN macro 313
SYSPAG macro 315
SYSPCLAS macro 354
SYSRES macro 329
SYSTIME macro 350
DOS (Disk Operating System)
Initialize Disk utility program
alternate tracks 64
macro libraries for CMS 46
programs
assembling 46
support under CMS 50
DPMSIZE operand
RDEVICE macro 264
dual-frame configuration
3880 Storage Subsystem 90
dump space
DASD requirements 37
dumps
analysis
using IPCS 3
directory entry
example 179
routing 179
dyadic processor 10
dynamic address translation feature
System/370
introduction 3
dynamically modified channel program
restrictions 395

E

ECMODE option
defining in directory 199
Emulation Program (EP)
coding the RDEVICE macro 268
3704/3705
VM/SP HPO support 148
emulators
integrated emulators 377
ENABLE operand
SYSMON macro 337
error recording
cylinders
defining 332
DASD storage requirements 36
error recovery
3340 and 3344 62
example
operating systems in a virtual machine
system dump directory entry 179
remote 3270 addressing 247
SCREEN directory control statement 209
SYSFORM macro 352
SYSID macro 356
SYSMIH macro 358
SYSPCLAS macro 354
SYSRES macro 335
TERMINAL macro 247
Expanded Storage
See Paging Storage
expanded virtual machine assist 122
extended channel support 110
restrictions 415
extended-control mode
System/370
introduction 3
Extended Control-Program Support 122
description 122

processors supported 122
Extended Storage Support
general information 122
performance 311
processors supported 122
SYSCOR macro 343
system generation considerations 122
TEST BLOCK
validating storage 116

F

FB-512
allocating DASD space 34
alternate blocks
minidisks 64
coding RDEVICE macro 255
DASD space requirements
CP 36
disks 36
format defective block procedure 64
minidisk space allocation 58
specifying in SYSRES macro 331
FCB operand
RDEVICE macro 264
FEATURE operand
of TERMINAL macro 247
RCTLUNIT macro 278
RDEVICE macro 258
features
IX/370 handshaking 84
processor
Channel Indirect Data Addressing 11
RSCS 24
two-channel Switch 26
VM/VS handshaking 80
files
CMS
maximum number of records 53
directory 53
management
CMS 47
OS/DOS 51
VSAM 50
fixed head feature
RDEVICE macro 258
font offset buffer
altering 365
form width codes 218
format defective block procedure
FB-512 disks 64
Format/Allocate program
flagging defective tracks 64
formatting minidisks 61
forms control buffer
file definitions
introduction 5
supplied with starter systems 365
Formula 1 (calculating available real storage) 118
Formula 2 (calculating maximum size of virtual=real area) 119
FORTRAN IV
compiling programs under CMS 48
four-channel switch
RDEVICE macro 259
FREE operand
SYSCOR macro 343
free storage
permanently allocated for CP 29

G

general polling characters 245
figure 247
GENIMAGE
updating a 3800 named system 157
GENLINK macro
use with RSCS 83
GPOLL operand
CLUSTER macro 245
GRAF operand
SYSMIH macro 358
guest operating system, defined 170

H

handshaking feature
IX/370 84
VM/VS 80
hardware
maintenance directory entry
example 182
remote
supported configurations 139
support
virtual machines 178
HELP files
with UCR 234
highlight
defining
SCREEN directory control statement 209

I

IBM
Field Developed Programs (FDPs) 377
program products
executable under CMS 377
IBM program numbers 377
integrated emulators 377

IBM-defined privilege classes 231
  list of 231
ID operand
  SYSTIME macro 350
identification
  of CMS files 54
IEHDASDR
  assigning alternate tracks 64
IMAGE operand
  RDEVICE macro 264
IMAGELIB
  updating a 3800 named system 157
IMAGEMOD
  updating a 3800 named system 157
INCLUDE
  directory control statement 195
INDICATE command
  performance measurement 112
Input/Output Configuration Program 97
Installation Verification Procedure (IVP)
  system test
    introduction 6
Installed User Programs (IUPs) 377
INTDK (DOS Initialize Disk utility program)
  assigning alternate tracks 64
Integrated Storage Control
  supported models 12
Interactive Problem Control System
  conversion considerations 67
    CONVIPCS EXEC 68
  machine requirements 67
  programming requirements 67
  storage requirements
    calculating DASD space 69
    external storage 68
internal control block
  generating 343
internal override file 235
internal pointer variables
  generating with SYSLOCS 363
IPCS (Interactive Problem Control System) 67
  considerations 40
  introduction 3
  storage usage by device type 40
IPL
  directory control statement 208
ISAM Interface Program (IIP) 73
ISAM option
  defining in directory 200
IUCV
  authorizing communication path 206
  connections
    specifying maximum number 206
  directory control statement 206
  structure of the SNA environment 153
  virtual machine communication facility 107
IX/370 handshaking feature 84

J

JOURNAL operand
  SYSJRL macro 340
journalizing facility
  SYSJRL macro 340

L

label information area
  CMS/DOS 77
labeling
  CMS disk labels 52
  minidisks 65
languages
  programming
    supported for CMS/VSAM 72
  supported by CMS 48
libraries
  system macro libraries for CMS 46
LIMIT operand
  SYSACNT macro 348
  SYSMON macro 337
line code
  determining for 3270s 143
LINE operand
  CLUSTER macro 245
LINK
  command
    sharing minidisks 65
  command (CP)
    use in file sharing and access 51
  directory control statement
    preferred machine assist 225
linking
  CMS disks 52
  command (CMS)
    usage 52
LNKLMT operand
  SYSJRL macro 340
LNKUID operand
  SYSJRL macro 340
load lists
  attached processor system 135
  multiprocessor system 136
loader tables
  CMS storage requirements 43
LOC operand
  SYSTIME macro 350
local attachments
  3270s 145
logical device facility
  subfunctions 107
logical device support facility
  introduction 106

LOGLMT operand
SYSJRL macro 340
LOGUID operand
SYSJRL macro 340

M

macros
DOS macro libraries under CMS 46
grouping into families 371
NAMENCP 301
NAMESYS 296
NAME3800 303
OS macro libraries under CMS 46
RCHANNEL 282
RCTLUNIT 277
RDEVICE
coding for remote 3270s 266
coding for 3704/3705/3725 communications controller 268
coding for 3800 image library 273
coding for 3880 storage subsystem 274
RIOGEN 284
SYSACNT 348
SYSCOR 343
SYSFCN 361
SYSFORM 352
SYSID 356
SYSLOCS 363
SYSMIH 358
SYSMON 337
SYSOPR 347
SYSOWN 313
SYSPAG 315
SYSPCLAS 354
SYSRES 329
SYSTIME 350
TSO macro libraries under CMS 46
VSE macro libraries under CMS 46
3704/3705/3725 communications controllers 268
magnetic tape
control units supported by VM/SP HPO 13
devices supported by VM/SP HPO 13
main storage protection, defined 170
maintenance
example
hardware maintenance directory entry 182
Mass Storage Control
channel interfaces 160
introduction 159
Mass Storage Facility 159
Mass Storage System
communication device
defining 162
creating MSS volumes 164
generating 159
generating CP to support 3850 159
hardware supported 159
mass storage control table 163
restrictions 409
MAXCONN option
defining in directory 204
MAXDEV operand
OPTION control statement 205
MDISK
control statement
sharing minidisks 65
directory control statement 213
example 213
messages
RCTLUNIT macro
channel errors 281
RDEVICE macro
control unit errors 265
unit record errors 265
minidisk
restrictions 396
minidisks
allocating for VTOC 60
alternate tracks
2314/2319 63
3330/3350 60
3340 62
defective tracks
2314/2319 64
defining 57
in directory 213
in directory, examples 213
primary access mode 213
directory
overlapping extents 174
error recovery support 62
example of use and definition 58
formatting 61
initialization
OS and VSE 61
labeling 65
linking 65
MDISK control statement 65
sharing
access modes 65
passwords 65
virtual machines 65
temporary 57
3340
using DDR utility 63
3344
using DDR utility 63
minimum configurations
CMS 8
RSCS 9
VM/SP HPO 7
missing interruption handler
devices supported 105
timer intervals 305, 358
missing interruption monitoring
SYSMIH macro 358
MODEL operand

RDEVICE macro 257
SYSID macro 356
TERMINAL macro 247
models
3704/3705 communications controllers 271
MONITOR command
performance measurement 112
monitoring and service support facility (MSSF)
MSSFCALL instruction 96
SCPINFO command 96
MP operand
SYSCOR macro 343
MSS operand
SYSMIH macro 358
multi-write access of minidisks 65
multiple
write access of minidisks 65
multiple service record files
3033AP 27
multiprocessor system
COPY OPTIONS 241
generating 136
load list 31
modules 136
performance measurement 112
specifying MP initialization
SYSCOR macro 343
system identification
SYSID macro 356
System/370 Extended Feature 125
MVS/System Extensions support
requirements 124
System/370 Extended Facility
processors supported 125
System/370 Extended Feature
processors supported 125
MVS/System Product support
cross memory 126
preferred machine assist 127
requirements 125
single processor mode 126
System/370 Extended Facility
processors supported 125
System/370 Extended Feature
processors supported 125
V=R recovery 127

N

named system
creating
3800 printing subsystem 303
NAMENCP macro
CPNAME operand 301
CPSIZE operand 301
CPTYPE operand 302
format 301
SYSPGCT operand 302
SYSSTRT operand 302
SYSVOL operand 302
NAMESYS macro
format 296
PROTECT operand 298
RCVRID operand 299
SAVESEQ operand 299
SYSCYL operand 297
SYSHRSG operand 298
SYSNAME operand 296
SYSPGCT operand 298
SYSPGNM operand 298
SYSSIZE operand 296
SYSSTRT operand 297
SYSVOL operand 297
VMGROUP operand 299
VSYSADR operand 296
VSYSRES operand 296
NAME3800 macro
CPNAME operand 303
format 303
SYSPGCT operand 303
SYSSTRT operand 304
SYSVOL operand 303
NARROW operand
SYSFORM macro 352
NCP
structure of the SNA environment 153
nucleus
CP
DASD requirements 36
real storage requirements 29
reducing its size 33

O

OPER operand
SYSFCN macro 361
operands 231
operating systems
performance characteristics 111
performance guidelines 112
planning considerations 79
sharing the system residence volume 82
storage keys 79
that execute under VM/SP HPO 3
using reserve/release 101
virtual machine
example, directory entry 180
example, hardware maintenance directory entry 182
example, system dump directory entry 179
virtual machine assist 121
operator directory entry
example 179
OPERFORM operand

SYSFORM macro 352
OPTION
directory control statement 199
OPTION control statement
MAXDEV operand 205
OS (Operating System)
initializing minidisks 61
macro libraries for CMS 46
support under CMS 49
output
spooling classes
defining with RDEVICE macro 258
OUTPUT operand
SYSACNT macro 348
OVERRIDE command 235
OVERRIDE control statement 236, 238
override file
See also class-override file
internal 235
override program 235
OVRD space 234

P

page migration
description 320
paging hierarchy 320
paging
DASD requirements 39
performance considerations 311
VM/VS handshaking feature 80
paging hierarchy 320
correction 322
evaluation 321
Paging Storage
description 133
partitioned data sets
CMS
limited support 49
simulated
CMS supported 45
Pass-Through Facility
use with logical device support 106
password
sharing minidisks 65
password protected resource, defined 170
password suppression facility
SYSJRL macro 340
passwords
restricted
list of 176
unauthorized 176
PEP
structure of the SNA environment 153
PERFORM
data collection class 337
performance
characteristics 111
considerations
coding DMKSYS macros 311
Extended Storage Support 311
heavy I/O production 311
installations with large number of CMS users 311
read-only minidisks 311
using automatic monitoring facilities 312
using fixed head devices for paging 311
data collection
SYSMON macro 337
Extended Control-Program Support 122
guidelines 112
measurement and analysis
MONITOR command 112
SYSMON macro 112
MVS cross memory services 126
MVS/System Extensions
support 124
MVS/System Extensions support
processors supported 125
MVS/System Product
support 124
virtual machine 124
options 113
preferred machine assist 127
shadow table bypass assist 123
single processor mode 127
virtual machine assist 121
physical record
CMS 53
PL/I
compiling programs under CMS 48
planning a class structure 232
PMA option
defining in directory 200
preferred machine assist
DEDICATE directory control statement 221
directory 200
DMKRIO 241
Extended Storage Support 127
LINK directory control statement 225
OPTION statement 200
performance 127
restrictions 416
RMSIZE operand 343
T-DISK in directory 213
use with Extended Storage Support 114
preferred system paging
3880 Storage Subsystem 91
PRIME operand
SYSCOR macro 343
printer forms
specifying
SYSFORM macro 352
PRIV operand
SYSFCN macro 361
privilege classes 231
assigning more than 8 196
defining more than 8 196

IBM-defined 231
list of 231
problem analysis
using IPCS 3
processor
dyadic 10
features
RSCS 25
required features 11
supported by VM/SP HPO 10
processors
restrictions 401
profile
directory
creating 191
description 174
example 175
invoking 195
directory control statement 191
program languages
supported by CMS 48
supported for CMS VSAM 72
program numbers
IBM program products 377
programmable remote stations
features
required for RSCS 25
PROTECT operand
NAMESYS macro 298
pseudo-cylinder
converting FBA pages to 35
PSUPRS operand
SYSJRL macro 340
PUT (Program Update Tape)
introduction 6
PUT Document
introduction 6

QUERY command 231
QUERY SRM PGMACT command 321

RCHANNEL macro
ADDRESS operand 282
CHTYPE operand 282
example 282
format 282
RCHBLOK
creating 282
generated label 282
RCTLUNIT macro
ADDRESS operand 278
ALTCH operand 278
channel errors 281
coding to support the 3880 Storage Subsystem 278
configuration aid 383
CUTYPE operand 278
example 281
FEATURE operand 278
FEATURE=xxx-DEVICE operand 277
format 278
using to define alternate paths 89
RCUBLOK
addressing 277
creating 277
label generated 277
RCVRID operand
NAMESYS macro 299
RDEVBLOK
creating 253
label generated 253
RDEVICE macro
ADAPTER operand 261
ADDRESS operand 254
ALTCU operand 262
BASEADD operand 264
CHAR operand 264
CLASS operand 260
CLUSTER operand 264
coding
CTCA 255
EP 3704/3705/3725 control program 268
example 264
summary of considerations for 3704/3705/3725 control program 271
system console 255
to support 3800 image library 273
TWX terminals 261
3270s 142
coding for remote 3270s 266
coding to support the 3880 Storage Subsystem 274
configuration aid 383
control unit error messages 265
CPNAME operand 263
CPTYPE operand 262
DEVTYPE operand 255
DPMSIZE operand 264
examples
3704/3705/3725 272
FCB operand 264
FEATURE coding considerations 258
FEATURE operand 258
format 254
four-channel switch feature 259
IMAGE operand 264
MODEL operand 257
output spooling classes
defining 258
SETADDR operand 262
subclass

defining for unsupported devices 260
two-channel switch feature 259
unit record error messages 265
3704/3705/3725 error messages 272
read-only
access of minidisks 65
READ SCP INFORMATION service call 96
read/write
access of minidisks 65
real control blocks
real storage requirements 29
real I/O configuration file
CLUSTER macro 245
coding
BSC lines 243
3270s 243
coding sample 287
example of VM/SP HPO system 286
preparing 241
RCHANNEL macro 282
RCTLUNIT macro 277
RDEVICE macro 253
RIOGEN macro 284
sequence of macros 241
TERMINAL macro 247
real I/O control block structure
for alternate channel specification 89
for alternate control unit specification 89
real storage
calculating maximum size virtual = real area 119
Extended Storage Support 29
requirements for CP 29
requirements for VM/SP HPO 29
saved systems
DASD requirements 121
validating 115
REALTIMER option
defining in directory 199
records
maximum usable number per CMS file 53
remote attachments
3270s
planning considerations 137
remote hardware
supported configurations
3270s 139
remote spooling devices
supported by VM/SP HPO 24
requirements
changing the directory 186
reserve/release
handling reserve CCW 103
hardware-supported 100
restrictions
device sharing between real processors 105
device/minidisk sharing 105
no dynamic path selection 105
shared DASD 100
summary of support 103
using with operating systems 101
virtual 100
virtual machine simulation 103
resource identification codes
3270s
codes 143
sample list 144
restrictions
channel
virtual and real channels 401
channel model dependent functions 402
channel programs 395
CMS
BC mode 411
CMS simulation of OS 411
CMS/DOS 411
displaying time 411
interfaces 411
OS and DOS 411
OS data sets 411
programs under CMS/DOS 411
spooling 411
system logical units 411
VSAM and ISAM Interface Program 411
commands
OPTIONS COPYFILE 423
dynamically modified channel program 395
Extended Storage Support 407
I/O
graphics data stream 420
I/O device drops ready status 420
pages for I/O 420
Mass Storage System
Central Service Application Program 409
common inventory and journal 409
compatibility of MSC/DMKRIO/IOGEN 409
prohibited use 409
unique volume number 409
volume serial 409
3330V 409
minidisk
cylinder 0 396
DASD channel program 396
DASD I/O real search id 396
data buffer area 396
data buffer area for FB-512 396
Device Support Facility 396
Diagnostic Read/Write Home Address 396
end-of-cylinder condition 396
FBA and CE area 396
FBA diagnostic commands 396
multitrack search 396
OS/VS and 3375/3380 396
preferred machine assist 396
seek 396
miscellaneous
channel-to-channel adapters 421
conmode 3270 423
conmode 3270 and MVS 423
directory control statements 418

logical device support 418
non-display read up 418
remote 3270's 421
SEREP program 421
service record file 421
small CP option 418
number of virtual devices 414
preferred machine assist 416
processor model dependent functions 401
processors
application programs and 2K keys 401
channel error recovery 401
machine layout area 401
processor identification 401
simulation of processor models 401
VMSAVE/SAVESYS and 2K/4K keys 401
158 401
2K and 4K keys 401
reserve/release
device sharing between real processors 105
device/minidisk sharing 105
no dynamic path selection 103
timing dependencies
devices that expect interruptions within a fixed time 400
PCI 400
telecommunication access methods 400
virtual block multiplexer channel 400
virtual device 414
virtual machines
control units 403
DIAGNOSE 403
discontiguous saved segments 403
DUMMY feature 403
I/O errors 403
integrated emulators 403
IPL command 403
IPL NOCLEAR 403
MVS subsystems and applications programs 403
PCI fetch 403
pseudo-timer 403
Read/Write Direct 403
SET CLOCK 403
size 403
spooling 403
tape mode setting 403
V=R 403
VM/SP release 2 403
1050/152 Model 2 Data Communication System 403
1052 console 403
3066 403
3081 403
RIOGEN macro
ALTCONS operand 284
CONS operand 284
example 284
format 284
SRF operand 284
RMSIZE operand
SYSCOR macro 343
RPQ
remote 3270 support 140
RPWLIST DATA file 176
RSCS (Remote Spooling Communications Subsystem)
compatible devices 393
defining
more than one RSCS virtual machine 83
example of use 82
introduction 3
minimum configuration 9
planning considerations 83
processor
required features 25
programmable terminals
required features 25
required features 24
RSCS Networking 24
2770 required features 24
2780 required features 24
required for RSCS 24
3770 required features 24
3780 required features 25
RSSIZE operand
SYSCOR macro 343

S

S-disk
CMS system disk
accessing 51
saved systems
calculating DASD space 121
CMS 55
DASD requirements 38
defining 296, 367-373
description 367
installing 368-369
SNTMAP EXEC 368
naming 296
security 373
SAVESEQ operand
NAMESYS macro 299
SCPINFO command
usage with monitoring and service support facility 96
SCREEN
directory control statement 209
secondary storage protection, defined 170
security 231, 233
access control groups 175
considerations for preparing directory 175
discontiguous saved segments 373
password deactivation 176
saved systems 373

SELECT operand
TERMINAL macro 247
SERIAL operand
SYSID macro 356
SERV operand
SYSFCN macro 361
service processor
READ SCP INFORMATION service call 96
service programs
directory program 186
service record file
capability 27
specifying
RIOGEN macro 284
SET MITIME command 106
SET SRM PGFULL command 320
SETADDR operand
RDEVICE macro 262
shadow table bypass assist
processors supported 122
SET STBYPASS command 123
shared DASD
reserve/release support 100
shared segments
description 369
usage notes 369
simulated devices
I/O
specifying 228
single-frame configuration
3880 Storage Subsystem 90
single processor mode 127
SMSG command
(CP)
use with VMCF 109
SNA CCS
excluding modules 155
NCP and PEP sharing 153
planning considerations 153
structure of the SNA environment 153
supported devices 9
tracing transactions
error trace 154
normal trace 154
usage with IUCV 108
SNTMAP EXEC 370, 373
errors 373
input 370
introduction 370
invoking 372
messages 372
output 371
prompts 372
use in installing discontiguous saved segments 368
use in installing saved systems 368
use of 371
software
support
virtual machines 178
SPECIAL
directory control statement 228
Special Message Facility 109
SPOOL
directory control statement 218
spooling
accounting records
SYSACNT macro 348
DASD requirements 39
defining virtual devices 218
output classes
defining with RDEVICE macro 258
performance considerations 311
SRF
devices
specifying, RIOGEN macro 284
SRF mode 27
SRF operand
RIOGEN macro 284
Staging Adapter
channel interface positions 160
starter systems
directories supplied 182
storage
Extended Storage Support 118, 122
for each model of the 3704/3705 271
storage director
3880 Storage Subsystem 90
storage keys
2K key operating systems 79
2K key support 79
4K key operating systems 79
4k key support 79
STQUERY operand
SYSJRL macro 340
summary
VMCF functions 109
support packages
3704/3705/3725 control program 148
SVCOFF option
defining in directory 202
symbolic names
CMS devices 44
SYSACNT macro
CLASS operand 348
format 348
LIMIT operand 348
OUTPUT operand 348
spooling accounting records 348
USERID operand 348
SYSCKP operand
SYSRES macro 333
SYSCLR operand
SYSRES macro 332
SYSCOR macro
AP operand 343
example 343
format 343
FREE operand 343
MP operand 343

PRIME operand 343
RMSIZE operand 343
RSSIZE operand 343
TRACE operand 343
SYSCYL operand
NAMESYS macro 297
SYSDUMP operand
SYSOPR macro 347
SYSERR operand
SYSRES macro 332
SYSFCN macro
CPRD operand 361
CPWT operand 361
DFLT operand 361
format 361
OPER operand 361
PRIV operand 361
SERV operand 361
SYSFORM macro
DEFCON operand 352
DEFPRT operand 352
DEFPUN operand 352
format 352
NARROW operand 352
OPERFORM operand 352
USERFORM operand 352
SYSHRSG operand
NAMESYS macro 298
SYSID macro
DEFAULT operand 356
format 356
MODEL operand 356
SERIAL operand 356
SYSTEMID operand 356
SYSJRL macro
format 340
JOURNAL operand 340
LNKLMT operand 340
LNKUID operand 340
LOGLMT operand 340
LOGUID operand 340
PSUPRS operand 340
STQUERY operand 340
SYSLOCS macro
example 363
format 363
SYSMIH macro
DASD operand 358
format 358
GRAF operand 358
MSS operand 358
samples 305
TYPE operand 358
UR operand 358
SYSMON macro
AUTO operand 337
BUFFS operand 337
CLASS operand 337
ENABLE operand 337
example 337
format 337
LIMIT operand 337
TIME operand 337
USERID operand 337
SYSNAME operand
NAMESYS macro 296
SYSNUC operand
SYSRES macro 331
SYSOPER operand
SYSOPR macro 347
SYSOPR macro
example 347
format 347
SYSDUMP operand 347
SYSOPER operand 347
SYSOWN macro
example 313
format 313
VOLID operand 313
SYSPAG macro
error messages 326
example 324
format 315
restrictions 420
SYSPCLAS macro
BOTTOM operand 354
format 354
TOP operand 354
SYSPGCT operand
NAMENCP macro 302
NAMESYS macro 298
NAME3800 macro 303
SYSPGNM operand
NAMESYS macro 298
SYSRES macro
example 335
format 329
special coding considerations 329
SYSCKP operand 333
SYSCLR operand 332
SYSERR operand 332
SYSNUC operand 331
SYSRES operand 330
SYSTYPE operand 331
SYSVOL operand 330
SYSWRM operand 334
SYSRES operand
SYSRES macro 330
SYSSIZE operand
NAMESYS macro 296
SYSSTRT operand
NAMENCP macro 302
NAMESYS macro 297
NAME3800 macro 304
system configuration
AP or MP
load list 31
UP
load list 31
system console

coding RDEVICE macro 255
configuration aid 383
defining 284
system control file
performance considerations 311
preparing 305
supplied with starter systems 306
SYSCOR macro 343
SYSFCN macro 361
SYSFORM macro 352
SYSID macro 356
SYSLOCS macro 363
SYSMIH macro 358
SYSOPR macro 347
SYSOWN macro 313
SYSPAG macro 315
SYSPCLAS macro 354
SYSRES macro 329
SYSTIME macro 350
system disk
CMS
planning considerations 45
system dump directory entry
operating systems
virtual machine example 179
system generation
considerations for VSE 75
corequisite publications 5
defining
your system 167
DMKRIO
preparation 241
Extended Storage Support 122
introduction 3, 169
options
virtual=real 113
RDEVICE macro
coding considerations for
3704/3705/3725 271
requirements
locally supported display systems 146
remotely-attached display systems 142
to support channel switching between two
processors 85
to support channel switching on one
processor 86
3704/3705/3725 requirements 149
3800 image library requirements 157
system identification
SYSID macro 356
system name table
creating an entry for 3704/3705 150
NAMENCP macro 301
NAMESYS macro 296
system operator
defining in SYSOPR macro 347
system residence devices
alternate tracks 63
system residence volume
sharing among the same operating system 82
system support
virtual machines 177
system text libraries
for CMS 47
system timing facility
System/370
introduction 3
System/370
requirements
dynamic address translation feature 3
system timing facility 3
System/370 Extended Facility
functions 125
processors supported 125
System/370 Extended Feature
attached processor restriction 125
processors supported 125
SYSTEMID operand
SYSID macro 356
SYSTIME macro
example 350
format 350
ID operand 350
LOC operand 350
ZONE operand 350
SYSTYPE operand
SYSRES macro 331
SYSVOL operand
NAMENCP macro 302
NAMESYS macro 297
NAME3800 macro 303
SYSRES macro 330
SYSWRM operand
SYSRES macro 334

T

T-DISK
defining 57, 213
defining in directory
preferred machine assist 213
tapes
channel switching 87
CMS restrictions 54
configuration aid 383
control units supported by VM/SP HPO 13
devices supported by VM/SP HPO 13
handling
CMS/DOS 77
support for CMS 54
TERM operand
TERMINAL macro 247
TERMINAL macro
coding 247
control unit and device addressing 247
examples 247
FEATURE operand 247

format 247
MODEL operand 247
remote 3270 addressing 247
SELECT operand 247
TERM operand 247
3270s 141
terminals
required features and special considerations 16
supported as virtual system consoles 14
supported by VM/SP HPO 14
supported by 3704/3705/3725 control program 147
system generation restrictions 14
TEST BLOCK
Extended Storage Support
validating storage 116
supported operating systems 115
308x and 3090 hardware instruction
validating storage 115
text libraries
CMS 47
Time-of-Day (TOD) clock
defining in SYSTIME macro 350
TIME operand
SYSMON macro 337
TOP operand
SYSPCLAS macro 354
trace entries
SNA CCS 154
TRACE operand
SYSCOR macro 343
trace table
CP real storage requirements 29
tracks
alternate
3330/3350 60
3340, cylinder assignments 62
transmission control units (TCUs)
configuration aid 383
remote 3270 support 139
supported by VM/SP HPO 21
2702 required features 22
2703 features 22
3704/3705/3725 features 23
TSO
macro libraries for CMS 46
two- or four-channel switch
3880 87
two-channel switch
alternate path support 88
channel switching
feature 26
RDEVICE macro 259
TWX terminals
coding RDEVICE macro 255, 261
type (command parameter) 232
TYPE operand
SYSMIH macro 358

U

UCR
See user class restructure
uniprocessor system
load list 31
unit record
control units supported by VM/SP HPO 14
devices
configuration aid 383
support for CMS 44
supported by VM/SP HPO 14
error messages for RDEVICE macro 265
support for CMS 55
unit record table
generating 284
universal character set
altering 365
UNLOCK command
releasing virtual=real area 117
unsupported devices
defining subclass 260
RDEVICE macro 255
updating
a 3800 named system 157
UR operand
SYSMIH macro 358
USER
data collection class 337
directory control statement 192
user class restructure 231
user class structure
changing 234
described 231
planning 232
USER control statement 239
user forms
specifying
SYSFORM macro 352
user privilege class structure
See user class structure
user privilege classes
See privilege classes
user program area
CMS storage requirements 43
USERFORM operand
SYSFORM macro 352
USERID operand
NAMESYS macro 299
SYSACNT macro 348
SYSMON macro 337

V

VARY command
  use with monitoring and service support facility 96
  use with service processor 96
Vector Facility
  description 28
  storage requirements 40
VIRT=REAL option
  defining in directory 200
virtual
  system console
    supported by VM/SP HPO 14
virtual disks
  capacity under CMS 51
  CMS
    maximum number of files 53
  defining in directory 213
virtual interval timer assist
  description 122
virtual machine assist
  expanded 122
  general information 121
  processors supported 122
virtual machines
  automatic system loading at user logon 208
  communication facility
    IUCV 107
    VMCF 109
  dedicating real devices 221
  defining
    in directory 177
    spooling devices 218
    the virtual console in the directory 213
  example
    RSCS 82
  hardware support 178
  linking devices at logon 225
  multiple
    using the same operating system 82
  naming systems 296
  sharing minidisks 65
  software support 178
  system support 177
  using reserve/release 103
  VMSAVE option 204
virtual storage
  required by CMS 43
virtual storage requirements 117
virtual=real
  bypassing CCW translation 113
  calculating maximum size of area 119
  determining the size of CP 36
  generating CP to support 114
  recovery 127
  releasing the area 114
  restrictions 114
  specifying a virtual=real machine 113
  specifying the amount of space 117
  storage 113
  use with multiport teleprocessing system 115
  virtual storage requirements 117
VM/SP HPO
  (EP) Emulation Program support 148
  DASD storage requirements for directory 36
  defining your system 167
  direct access storage devices supported 11
  dump space
    DASD requirements 37
  introduction 3
  minimum configurations 7
  paging
    DASD requirements 39
  real devices supported 9
  real storage requirements
    for CP 29
    reducing size of the CP nucleus 31
  spooling
    DASD requirements 39
  system definition
    directory 173
    forms control buffer load 365
    real I/O configuration file 241
    system control file 305
  system generation
    introduction 4
  terminals supported 14
  transmission control units supported 21
  two-channel switch feature 26
  unsupported devices in virtual machines 26
  VSAM and Access Method Services requirements 39
VM/VS handshaking feature 80
VM/370
  assembler 377
VMCF (Virtual Machine Communication Facility)
  summary of functions 109
VMGROUP operand
  NAMESYS macro 299
VMSAVE option
  defining in directory 204
VOLID operand
  SYSOWN macro 313
volume label contents
  CMS formatted disks 52
VS APL
  use with CMS 48
VSAM (Virtual Storage Access Method)
  CMS support for DOS users 50
  considerations for installing 73
  data set compatibility with CMS 72
  hardware devices supported 71
  ISAM Interface Program 73
  programming languages supported 72
  storage requirements 39
VSE
  macro libraries for CMS 46

support under CMS 50
VSYSADR operand
NAMESYS macro 296
VSYSRES operand
NAMESYS macro 296
VTAM
structure of the SNA environment 153
VTOC
minidisk allocation 60

W

warm start data
calculating number of cylinders needed 335
DASD storage requirements 36
defining cylinders 335

X

XMEM option
defining in directory 204
xxx-DEVICE feature of RCTLUNIT macro 277

Z

ZONE operand
SYSTIME macro 350

Numerics

1017
coding RDEVICE macro 255
1018
coding RDEVICE macro 255
1050
control units
models and features 17
1052
coding RDEVICE macro 255
1053
coding RDEVICE macro 255
1403
altering font offset buffer 365
altering universal character set 365
coding RDEVICE macro 255
supported models 14
1442P
coding RDEVICE macro 255
1442R
coding RDEVICE macro 255
1443
coding RDEVICE macro 255
supported models 14
2150
coding RDEVICE macro 255
supported models 14
2250
coding RDEVICE macro 255
2260
coding RDEVICE macro 255
2265
coding RDEVICE macro 255
2301
coding RDEVICE macro 255
2303
coding RDEVICE macro 255
2305
allocating DASD space for directory 183
coding RDEVICE macro 255
DASD space requirements
checkpoint start data 36
CP nucleus 36
directory 36
error recording 36
paging and spooling 36
saved systems 36
warm start data 36
DASD storage requirements
CP 36
specifying in SYSRES macro 331
supported models 11
2311
coding RDEVICE macro 255
2314
allocating DASD space for directory 183
capacity for CMS minidisks 41
coding RDEVICE macro 255
DASD space requirements
checkpoint start data 36
CP nucleus 36
directory 36
error recording 36
paging and spooling 36
saved systems 36
warm start data 36
DASD storage requirements
CP 36
specifying in SYSRES macro 331
2319
allocating DASD space for directory 183
alternate tracks
minidisks 63
coding RDEVICE macro 255
specifying in SYSRES macro 331
2321
coding RDEVICE macro 255
2401
coding RDEVICE macro 255
supported models 13
2402

coding RDEVICE macro 255
supported models 13
2403
coding RDEVICE macro 255
2404
coding RDEVICE macro 255
2415
coding RDEVICE macro 255
supported models 13
2420
coding RDEVICE macro 255
supported models 13
2495
coding RDEVICE macro 255
2501
coding RDEVICE macro 255
supported models 14
2520
supported models 14
2520P
coding RDEVICE macro 255
2520R
coding RDEVICE macro 255
2540
supported models 14
2540P
coding RDEVICE macro 255
2540R
coding RDEVICE macro 255
2671
coding RDEVICE macro 255
2701
coding RDEVICE macro 255
features
required and optional 21
remote 3270 support 139
2702
coding RDEVICE macro 255
features
required 22
2703
coding RDEVICE macro 255
2741
features
required and desirable 16
supported models 15
2770
features
required for RSCS 24
supported models 24
2780
features
required for RSCS 24
supported models 24
2803
supported models 13
2804
supported models 13
2816
supported switches 88
switching unit 87
2821
supported models 14
2835
supported models 12
2844
supported models 12
2955
coding RDEVICE macro 255
3031
specifying system console 255
3032
specifying system console 255
3033
AP
channel-set switching 95
multiple service record files 27
specifying SRF devices 284
specifying system console 255
3036
coding RDEVICE macro 255
supported models 15
3066
coding RDEVICE macro 255
supported models 14
308x
AP
channel-set switching 95
Input/Output Configuration Program 97
system monitoring and reconfiguration 96
validating real storage
TEST BLOCK instruction 115
3081
Input/Output Configuration Program 241
3082
processor controller 28
system monitoring and reconfiguration 96
3090
extended channel support 110
Input/Output Configuration Program 97
specifying system console 255
system monitoring and reconfiguration 96
validating real storage
TEST BLOCK instruction 115
Vector Facility 28
3101
RDEVICE macro
coding considerations 255
specifying an adapter 261
supported models 15
3138
coding RDEVICE macro 255
3148
coding RDEVICE macro 255
3158
coding RDEVICE macro 255
3203
altering font offset buffer 365
altering universal character set 365
coding RDEVICE macro 255

supported models 14
3210
coding RDEVICE macro 255
supported models 14
3211
altering font offset buffer 365
altering universal character set 365
coding RDEVICE macro 255
supported models 14
3215
coding RDEVICE macro 255
supported models 14
3262
altering font offset buffer 365
altering universal character set 365
coding RDEVICE macro 255
forms control buffer
altering 365
supported models 14
3270
coding CLUSTER macro 141
coding RDEVICE macro 142
coding real I/O configuration file 243
coding TERMINAL macro 141
components
control units, models and features 18
control unit and device addressing 247
determining line code 143
example of configuration 144
examples of addressing 247
hardware configuration supported 139
local attachments 145
planning considerations 137
RDEVICE macro
FEATURE coding considerations 258
resource identification codes 142
sample list 144
support on binary synchronous lines 138
system generation requirements 142
TERMINAL macro
control unit and device addressing 247
3270 information display
RDEVICE macro 266
3271
coding real I/O configuration macros 243
coding TERMINAL macro 141
remote 3270 support 139
3272
local 3270 support 145
3274
CLUSTER macro 243
coding CLUSTER macro 141
coding TERMINAL macro 141
local 3270 support 145
remote 3270 support 139
3275
coding CLUSTER macro 141
coding real I/O configuration macros 243
coding TERMINAL macro 247
remote 3270 support 139
supported models 15
3276
CLUSTER macro 243
coding CLUSTER macro 141
coding TERMINAL macro 247
supported models 15
3277
coding RDEVICE macro 255
coding TERMINAL macro 141, 247
local 3270 support 145
remote 3270 support 139
supported models 15
3278
coding RDEVICE macro 255
coding TERMINAL macro 141, 247
local 3270 support 145
remote 3270 support 140
supported models 15
3279
coding RDEVICE macro 255
coding TERMINAL macro 141, 247
defining color
SCREEN directory control statement 209
local 3270 support 145
remote 3270 support 140
supported models 16
3284
coding RDEVICE macro 255
coding TERMINAL macro 141, 247
local 3270 support 145
remote 3270 support 140
3286
coding RDEVICE macro 255
coding TERMINAL macro 141, 247
local 3270 support 145
remote 3270 support 140
3287
coding TERMINAL macro 141, 247
local 3270 support 145
remote 3270 support 140
3288
coding RDEVICE macro 255
coding TERMINAL macro 141, 247
local 3270 support 145
remote 3270 support 140
3289
altering font offset buffer 365
altering universal character set 365
coding RDEVICE macro 255
coding TERMINAL macro 141, 247
local 3270 support 145
remote 3270 support 140
supported models 14
3310
capacity for CMS minidisks 41
3330
allocating DASD space for directory 183
alternate tracks
minidisks 61
capacity for CMS minidisks 41

coding RDEVICE macro 255
DASD space requirements
checkpoint start data 36
CP nucleus 36
directory 36
error recording 36
saved systems 36
warm start data 36
DASD storage requirements
CP 36
specifying in SYSRES macro 331
starter system
forms control buffer supplied 365
introduction 4
supported models 11
3330-1
copying to 3330V volumes 164
3330V
copying 3330-1 volumes to 164
demounting 161
mounting 161
used as system volumes 161
using for VS system residence 165
3333
coding RDEVICE macro 255
supported models 11
3340
allocating DASD space for directory 183
alternate tracks
minidisks 62
capacity for CMS minidisks 41
coding RDEVICE macro 255
cylinder assignments 62
DASD space requirements
checkpoint start data 36
CP nucleus 36
directory 36
error recording 36
paging and spooling 36
saved systems 36
warm start data 36
DASD storage requirements
CP 36
DDR utility 63
error recovery 62
specifying in SYSRES macro 331
supported models 11
3344
DDR utility 63
error recovery 62
3350
allocating DASD space for directory 183
alternate tracks
minidisks 61
capacity for CMS minidisks 41
coding RDEVICE macro 255
DASD space requirements
checkpoint start data 36
CP nucleus 36
directory 36
error recording 36
paging and spooling 36
saved systems 36
warm start data 36
DASD storage requirements
CP 36
specifying in SYSRES macro 331
starter system
forms control buffer supplied 365
introduction 4
supported models 11
3370
capacity for CMS minidisks 41
supported models 11
3375
allocating DASD space for directory 183
DASD space requirements
CP nucleus 36
starter system
introduction 4
3380
allocating DASD space for directory 183
capacity for CMS minidisks 41
coding RDEVICE macro 255
DASD space requirements
checkpoint start data 36
CP nucleus 36
directory 36
error recording 36
paging and spooling 36
saved systems 36
warm start data 36
minidisk allocation
Device Support Facilities 60
specifying in SYSRES macro 331
starter system
forms control buffer supplied 365
introduction 4
supported models 11
3410
coding RDEVICE macro 255
supported models 13
3411
coding RDEVICE macro 255
supported models 13
3420
coding RDEVICE macro 255
supported models 13
3505
coding RDEVICE macro 255
supported models 14
3525
coding RDEVICE macro 255
supported models 14
370E option
defining in directory 204
3704
coding RDEVICE macro 268
remote 3270 support 139
3704/3705

coding considerations for RDEVICE macro 271
creating an entry in the system name table 150
eliminating support modules 33
models supported 271
naming the control program 301
nucleus
reducing its size 33
reserving DASD space for image of control program 150
storage sizes 271
3704/3705/3725
error messages for RDEVICE macro 272
examples of RDEVICE macro 272
features not supported 23
RDEVICE macro
summary of coding considerations 271
supported models 23
3704/3705/3725 control program
(EP) Emulation Program 148
generating
VM/SP to support 149
generation procedure
coding the RDEVICE macro 269
introduction 148
minimum storage required 148
planning considerations 148
related publications 150
support packages
documentation supplied 148
terminals supported 147
3705
coding RDEVICE macro 268
remote 3270 support 139
3725
coding RDEVICE macro 268
3767
features
required 17
supported models 15
3770
features
required for RSCS 24
supported models 24
3780
features
required for RSCS 25
supported models 24
3800
coding RDEVICE macro 273
hardware features supported 158
image library
coding RDEVICE macro to support 273
generating CP to support 157
named system
creating 157, 303
updating 157
related publications 158
specifying delayed purge queue 264
supported models 14
3803
supported models 13
3811
supported models 14
3830
specifying 64-device feature 277
supported models 12
3850
Mass Storage System (MSS) 159
copying 3330-1 volumes to 3330V volumes 164
creating MSS volumes 164
defining the MSS communication device 162
mass storage control tables 163
3851
coding RDEVICE macro 255
Mass Storage Facility 159
3880
specifying alternate channel 278
supported models 12
two- or four-channel switch feature 87
3880 Storage Subsystem
addressing 93
system generation considerations 93
configuration 90
dual-frame configuration 90
general information 90
preferred system paging 91
RCTLUNIT macro 278
RDEVICE macro 274
single-frame configuration 90
storage directors 90
supported models 11
system generation considerations 93
4341
specifying system console 255
7412
supported models 14
7443
coding RDEVICE macro 255

**Virtual Machine/
System Product
High Performance Option**

**READER'S
COMMENT
FORM**

**Planning Guide
and Reference**

Order No. SC19-6223-6

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you. Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate.

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

How did you use this publication?

[ ] As an introduction
[ ] As a reference manual
[ ] For another purpose (explain) _____

[ ] As a text (student)
[ ] As a text (instructor)

_____

Is there anything you especially like or dislike about the organization, presentation, or writing in this manual? Helpful comments include general usefulness of the book; possible additions, deletions, and clarifications; specific errors and omissions.

Page Number: Comment:

What is your occupation? _____

Newsletter number of latest Technical Newsletter (if any) concerning this publication: _____

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

_____

_____

IBM branch office serving you _____

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

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

SC19-6223-6

Reader's Comment Form

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

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY MAIL

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

POSTAGE WILL BE PAID BY ADDRESSEE

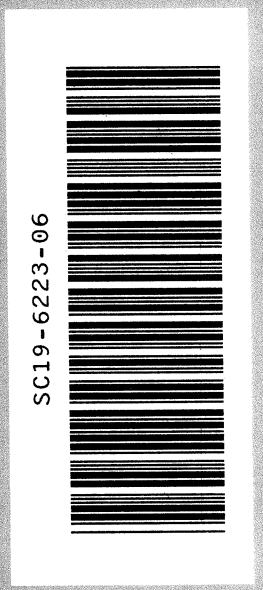
International Business Machines Corporation
Department 52Q MS 458
Neighborhood Road
Kingston, New York 12401

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

PRINTED IN U.S.A. SC19-6223-6

IBM®

VM/SP HPO Planning Guide and Reference File No. S370-34 Printed in U.S.A. SC19-6223-6



SC19-6223-6