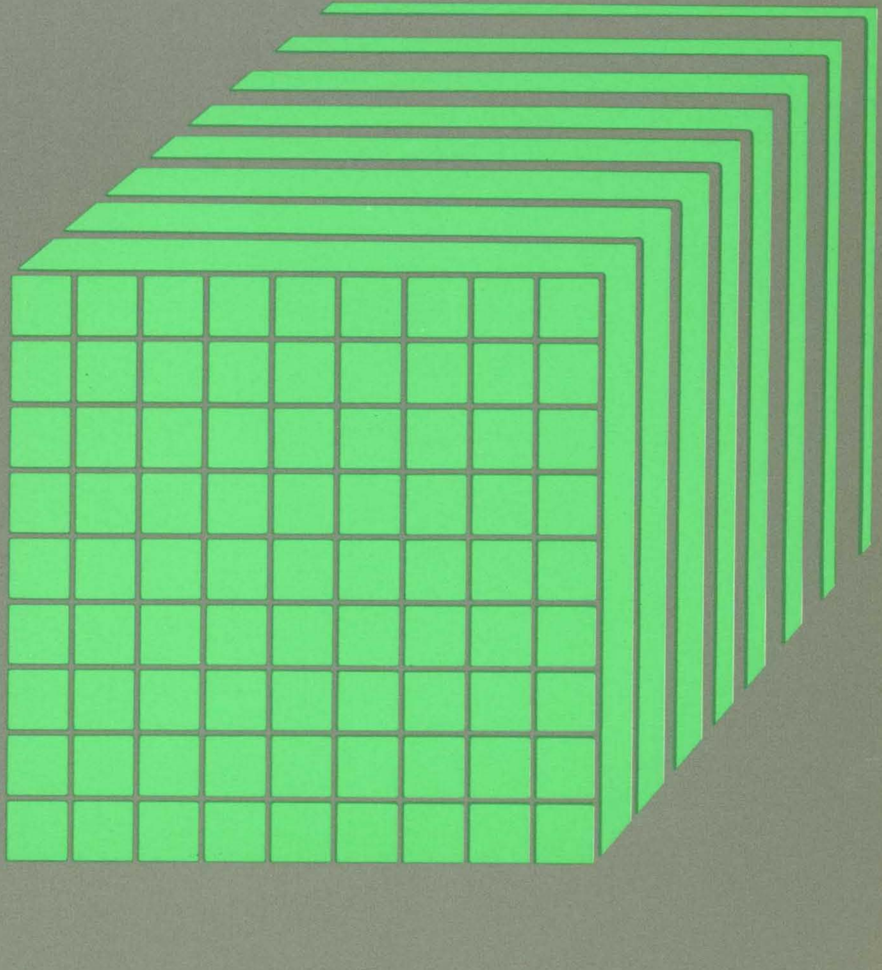
Virtual Machine/
System Product

Planning Guide and Reference

Release 3

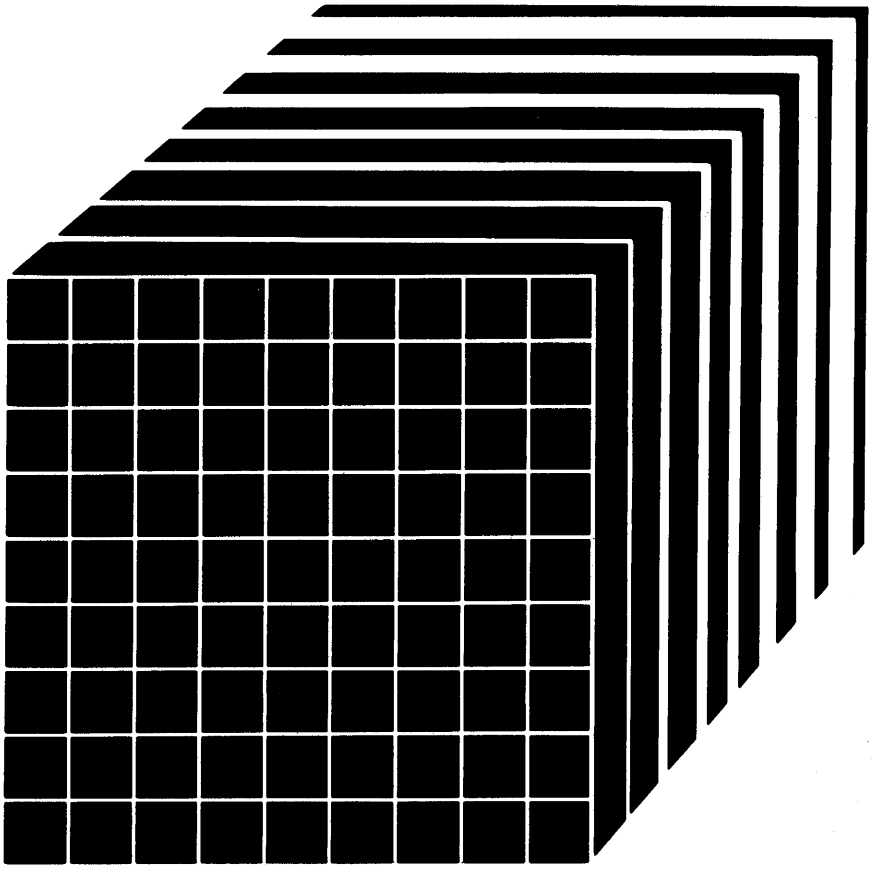


IBM

Virtual Machine/ System Product

Planning Guide and Reference

Release 3



Fourth Edition (September 1983)

This edition, SC19-6201-3 is a major revision of SC19-6201-2, and applies to Release 3 Virtual Machine/System Product (VM/SP), program number 5664-167, unless otherwise indicated in new editions or Technical Newsletters. Changes are made periodically to the information herein; before using this publication in connection with the operation of IBM systems, consult the latest *IBM System/370 and 4300 Processors Bibliography,* GC20-0001, for the editions that are applicable and current.

Changes or additions to the text and examples are indicated by a vertical line to the left of the change.

Summary of Changes

For a detailed list of changes, see page iii.

A form for reader's comments is supplied at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, Programming Publications, Dept. G60, P.O. Box 6, Endicott, NY, U.S.A. 13760. IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Summary of Changes

Summary of Changes
for SC19-6201-3
for VM/SP Release 3

- The title of this book used to be *VM/SP: Planning and System Generation Guide*, SC19-6201. While the order number (SC19-6201) still applies, the title has been changed to the *VM/SP: Planning Guide and Reference*. This title change is the result of moving the following sections to the new *VM/SP: Installation Guide*, SC24-5237:
  - "Part 3. Generating VM/SP (CP and CMS)"
  - "Part 4. Generating the 3704/3705 Control Program"
  - "Part 5. Updating VM/SP"
  - "Appendix C. CP/CMS Nucleus/Module Regeneration Requirements"
  - "Appendix F. A Sample EXEC Procedure for Copying VSE Macros into a CMS MACLIB"
  - "Appendix G. Generating VM/SP Without a VM/SP Starter System or the Merged Product Tapes"
  - The list of DMKSYS, Directory, and DMKSNT files supplied with the Product Tape
- CMS now supports the 512-byte block size for minidisks.
- The CMSSEG Saved Segment has been removed. Code from CMSSEG has been incorporated into the CMS Nucleus.
- The CMSXGEN Procedure that formerly generated CMSSEG has been deleted.
- The Small CP option has been updated to remove support modules for the 3066, 3800 printers, and the Missing Interrupt Handler.
- The VSE/VSAM macros and their options and a subset of the OS/VSAM macros are now supported for use in CMS.
- Support has been added for the following hardware devices:
  - The IBM 3088 Multisystem Communication Unit interconnects multiple systems using block multiplexer channels. The 3088 uses an unshared subchannel for each unique address and is fully compatible with existing channel-to-channel adapter protocol.
  - The IBM 3262 printer Model 5.
  - The IBM 3430 Tape Unit and Control.
  - The IBM 3800 printer Models 3 and 8.

  - The IBM 4245 Printer.
  - The IBM 4250 Printer.
- GAM/SP Release 2 is now supported under VM/SP. This support is provided for High Function Graphics Devices (HFGD).

Preface

This publication is intended for system programmers and those responsible for planning a VM/SP system.

You should have a general understanding of data processing methods and be familiar with teleprocessing techniques.

This publication has two parts, plus appendixes.

"Part 1. Planning for System Generation" describes the components, features, and options of VM/SP, and tells you what you must do during system generation to support them. Part 1 includes information about CMS and other operating systems in a virtual machine. It also discusses performance options, remote 3270s, the 3704/3705 control program, saved systems, discontiguous saved segments, CMS/DOS, VSAM under CMS, Access Method Services, attached processor and multiprocessor systems, storage requirements, and minidisks; Part 1 also lists the devices supported by VM/SP.

"Part 2. Defining Your VM/SP System" tells you how to create the files that define your system; the real I/O configuration (DMKRIO), CP system control (DMKSYS), VM/SP directory (DMKDIR), system name table (DMKSNT), forms control buffer load (DMKFCB) files, and the macros and control statements needed to create them.

The appendixes include information about:

- Licensed Programs and Integrated Emulators
- Configuring VM/SP
- Compatible devices
- VM/SP restrictions

In this publication, the following terms have extended meanings:

- The term "3330 series" refers to the IBM 3330 Disk Storage Models 1, 2, and 11; and the IBM 3333 Disk Storage and Control, Models 1 and 11.
- The term "2305 series" refers to the IBM 2305 Disk Storage, Models 1 and 2.
- The term "3262" refers to the IBM 3262 Printer, Models 1, 3, 5, 11, and 13.
- The term "3289-4" refers to the IBM 3289, Model 4 Printer.
- The term "3340 series" refers to the IBM 3340 Disk Storage, Models A2, B1 and B2, and the 3344 Direct Access Storage Model B2.
- The term "3350 series" refers to the IBM 3350 Direct Access Storage Models A2 and B2 in native mode.
- The term "3375" refers to the IBM 3375 Direct Access Storage Device.
- The term "3380" refers to the IBM 3380 Direct Access Storage Device.
- The term "FB-512" refers to the Fixed Block Architecture of the IBM 3310 and 3370 Direct Access Storage Devices.
- The term "3705" refers to the IBM 3705-I and 3705-II Communications Controllers, unless otherwise specified.
- The term "3270" is used in this publication to refer to all VM/SP supported virtual machine display consoles unless otherwise noted. A specific device type is used only when a distinction is required between device types.
- Information about display terminal use also applies to the IBM 3138, 3148, and 3158 Display Consoles, when used in display mode, unless otherwise noted.
- Any information pertaining to the IBM 3284 or 3286 printer also pertains to the IBM 3287, 3288, and 3289 printers, unless otherwise noted.
- The term "typewriter terminal" refers to printer-keyboard devices that produce hard-copy output only (such as the IBM 2741 Communication Terminal, the IBM 3215 Console Printer-Keyboard, or the IBM 3767 Communication Terminal, Model 1 or 2, operating as a 2741).
- The term "2741" refers to the IBM 2741 Communication Terminal, and also the 3767 Communication Terminal (unless otherwise noted).
- The term "display device" refers to any VM/SP supported system console terminal that displays data on a screen.
- Unless otherwise noted, where the term "Attention key" is used in this publication, the phrase "(or equivalent)" is implied. The equivalent key on the 1050 terminal is the RESET LINE key; on the 3276, 3277, 3278, and 3279 terminal, the Enter key. Each of the terminals that can be used with the VM/SP system has a key that is the equivalent of the Attention key on the 2741 (with which you can signal an attention interrupt).

- Unless otherwise noted, the term VSE refers to the combination of the DOS/VSE system control program and the VSE/Advanced Functions program product.

  In certain cases, the term DOS is still used as a generic term. For example, disk packs prepared for use with VSE or any predecessor DOS or DOS/VS system may be referred to as DOS disks.

  The DOS-like simulation environment provided under the CMS component of the VM/System Product continues to be referred to as CMS/DOS.

- CMS/DOS is part of the CMS system and is not a separate system. The term "CMS/DOS" is used in this publication as a concise way of stating that the VSE simulation mode of CMS is now active; that is, that the CMS command

  ```
  set dos on
  ```

  has been previously invoked.

- The phrase "the CMS file system" refers to disk files that are in CMS's 512, 800, 1K, 2K, or 4K fixed physical block format; CMS's VSAM data sets are not included.

- Unless stated otherwise, reference to the System/370 Models 138 and 148 also apply to Models 135-3 and 145-3, respectively.

- The term "3330V" is used in this publication for both volumes and device addresses. When used with volumes, it refers to a Mass Storage System volume that has been mounted and that is directly accessible from the processor. When used with device addresses, 3330V refers to a device on which 3330V volumes may be mounted by the Mass Storage System.

- If you have an IBM 3850 Mass Storage System attached to your processor, references to 3330 can be thought of as meaning 3330Vs unless the reference is to VM/SP system residence, paging or spooling devices.

Corequisite Publications

Virtual Machine/System Product:

General Information, GC20-1838

Introduction, GC19-6200

Release 3 Guide, SC24-5240

CMS Command and Macro Reference, SC19-6209

CMS User's Guide, SC19-6210

CP Command Reference for General Users, SC19-6211

Command Summary (General User), SX20-4401

Command Summary (Other than General User), SX20-4402

Quick Guide for Users, SX20-4400

System Programmer's Guide, SC19-6203

Installation Guide, SC24-5237

System Messages and Codes, SC19-6204

OLTSEP and Error Recording Guide, SC19-6204

Terminal Reference, GC19-6206

Library Guide and Master Index. GC19-6207

Operating Systems in a Virtual Machine, GC19-6212

Operator's Guide, SC19-6202

EXEC 2 Reference, SC24-5219

EXEC 2 Language Reference Summary, SX24-5124

System Product Editor User's Guide, SC24-5220

System Product Editor Command and Macro Reference, SC24-5221

System Product Editor Command Language Reference Summary, SX24-5122

System Product Interpreter User's Guide, SC24-5238

System Product Interpreter Reference, SC24-5239

System Product Interpreter Reference Summary, SX24-5126

Data Areas and Control Block Logic, Volume 1 (CP), LY24-5220

Data Areas and Control Block Logic, Volume 2 (CMS), LY24-5221

Service Routines Program Logic, LY20-0890

System Logic and Problem Determination Guide Volume 1 (CP), LY20-0892

System Logic and Problem Determination Guide Volume 2 (CMS), LY20-0893

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

Remote Spooling Communications Subsystem Networking General Information, GH24-5004

Remote Spooling Communications Subsystem Networking Program Reference and Operations Manual, SH24-5005

Remote Spooling Communications Subsystem Networking Reference Summary, SX24-5119

Remote Spooling Communications Subsystem Networking Logic, LY24-5203

Non-VM/SP Titles:

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

IBM 3850 Mass Storage System (MSS) Principles of Operation: Theory, GA32-0035

IBM 3850 Mass Storage System (MSS) Principles of Operation: Reference, GA32-0036

IBM 3850 Mass Storage System (MSS) Introduction and Preinstallation Planning, GA23-0038

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

Introduction to the IBM 3704 and 3705 Communications Controllers, GA27-3051

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, GC30-3007

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/AJ2, GC30-3008

IBM 3704 Control Panel Guide, GA27-3086

IBM 3705 Control Panel Guide, GA27-3087

IBM OS/VS Linkage Editor and Loader, GC26-3813

VM/VTAM Communication Network Application General Information Manual, GC27-0501

Virtual Machine/VTAM Communication Network Application: Installation, Operation, and Terminal Use, SC27-0502

Virtual Machine/VTAM Communication Network Application Messages, GC27-0510

Virtual Machine/VTAM Communication Network Application Logic, LY38-3033

Environmental Recording, Editing, and Printing (EREP) Program, GC28-1178

EREP Messages, GC28-1179

IPCS Extension User's Guide and Reference, SC34-2020

References in the text to titles of prerequisite and corequisite VM/SP publications are given in abbreviated form.

Figure 1 on page xii is an overview of the VM/SP library, with the publications grouped according to their probable users.

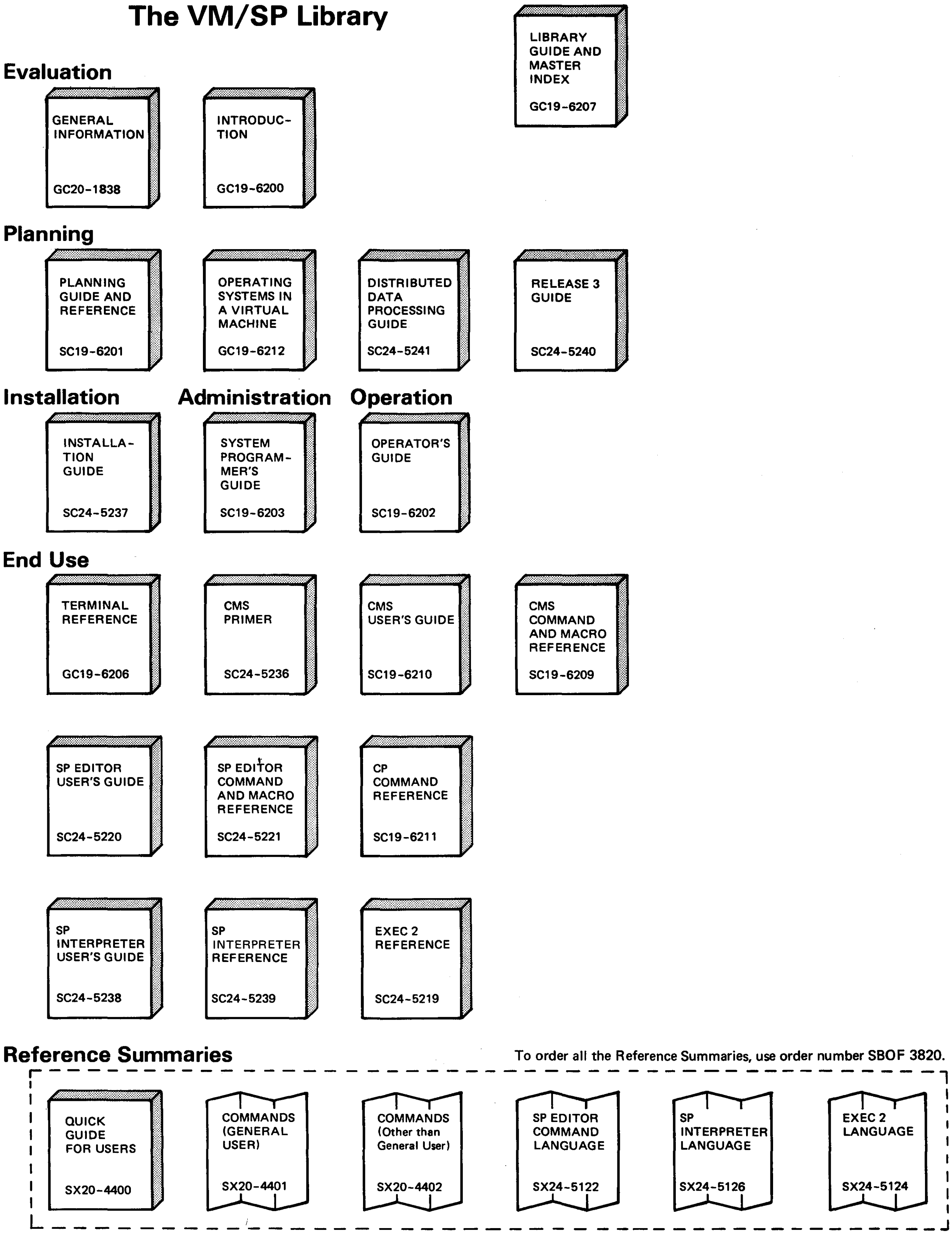


Figure 1 (Part 1 of 2). Virtual Machine/System Product Library

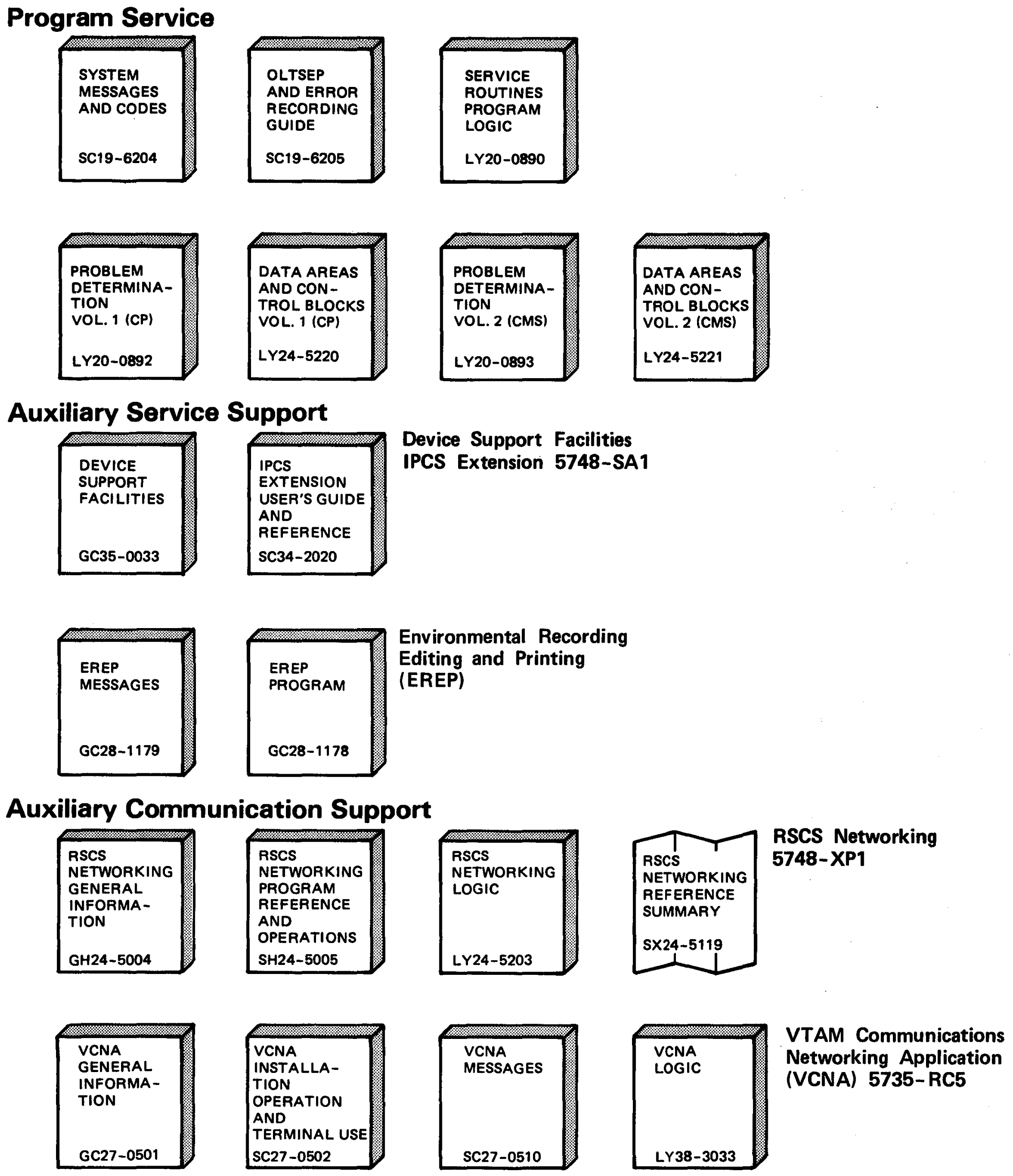


Figure 1 (Part 2 of 2). Virtual Machine/System Product Library

Contents

Figures

Part 1. Planning for System Generation

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

- Introduction to Planning
- Configurations
- Estimating VM/SP Storage Requirements
- Planning for CMS
- Minidisks
- Planning for CMS VSAM and Access Method Services
- Planning for CMS/DOS
- Planning for Virtual Machine Operating Systems (Other than CMS)
- Saved Systems and Discontiguous Segments
- Performance Guidelines
- Attached Processor and Multiprocessor Systems
- Planning for 3270s
- Planning for the 3704/3705 Control Program
- Planning for SNA Console Communications Services
- Planning for the 3800 Image Library
- Planning for the 3850 Mass Storage System

Chapter 1. Introduction to Planning

The IBM VM/System Product is a program product that manages a real system so that all its resources -- main (IPL) processor, attached (non-IPL) processor, storage, and input/output devices -- are available to many users at the same time. Each user has at their disposal the functional equivalent of a real, dedicated system. Because this functional equivalent is simulated by VM/SP and does not really exist, it is called a "virtual" machine.

VM/SP has two components:

- The control program (CP), which controls the resources of the real processor to provide multiple virtual machines.
- The Conversational Monitor System (CMS), which provides a wide range of conversational and time-sharing facilities. Using CMS, you can create and manage files; and compile, test, and run problem programs.

When you install VM/SP in conjunction with the VM/370 Release 6 System Control Program, it becomes a functional operating system that provides extended features to the System Control Program and Conversational Monitor System components of VM/370 Release 6.

The processors that VM/SP supports are listed under the heading "Processors" in Chapter 2. The real processor 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 processor, including dynamic address translation, are operational.

Virtual Machine Operating Systems

While the control program of VM/SP manages the concurrent execution of virtual machines, it is also necessary to have an operating system managing the work flow within each virtual machine. Because each virtual machine runs 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:

Batch or Single User Interactive

DOS
DOS/VS
VSE
OS/ASP
OS/PCP
OS/MFT
OS/MVT
OS/VS1
OS/VS2
RSCS

Multiple-Access

VM/370
VM/SP
Time Sharing Option of OS
MVS/SP

Conversational

CMS

CP provides each of these with virtual device support and virtual storage. The operating systems themselves run as though they are controlling real devices and real storage, but they must not violate any of the restrictions listed in "Appendix D. VM/SP Restrictions."

Introduction to VM/SP System Generation

The reason for the system generation is to create a VM/SP system that meets your particular needs.

The first step in the system generation procedure is to restore the starter system, a small working copy of a basic VM/SP system. Using the starter system, you tailor a VM/SP 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 VM/SP starter system can be ordered:

- 3330 Starter System
- 3340 Starter System
- 3350 Starter System
- 3375 Starter System
- 3380 Starter System
- FB-512 Starter System

All starter systems must be restored to a corresponding disk (that is, 3330 starter system to a 3330 disk), but all starter systems can then be used to build any supported system residence volume type (3330, 3340, 3350, 3375, 3380, or FB-512).

Before you begin the system generation procedure, you should:

- Know which devices to include in your VM/SP 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 VM/SP, you must coordinate the real I/O configuration with the Mass Storage Control tables.
- Decide how many virtual machines to define.
- Create the VM/SP 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, 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 product tape.

Once you have defined your VM/SP 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. Part 2 has the information you need to code the macro statements that define your system. The *VM/SP Installation Guide* describes the generation procedure step-by-step.

Chapter 2. Configurations

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

VM/SP Minimum Configuration

The minimum configuration supported by VM/SP is:

One	Processor (524,288 bytes of storage)
One	System console device
One	Printer
One	Card reader[1]
One	Card punch[1]
Two	Spindles of direct access storage[2]
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 VM/SP Storage Requirements."

A representative VM/SP configuration is:

IBM 4341 2Mb/4Mb Storage

IBM 3278 Display Console Model 2A

IBM 3203 Printer Model 5 -- Two

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 3705 Communications Controller

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

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

---

[1] This device is not needed for a cardless system.

[2] Three spindles are required if you are using 3310 or 3340-70 DASDs.

Configurations Supported by CMS

CMS supports:

- Virtual storage size: minimum of 256K bytes, up to 16 million bytes in multiples of 4K.

- Virtual console: any terminal supported by VM/SP as a virtual machine operator's console.

- The same unit record devices (card readers, punches, and printers) supported by VM/SP as spooling devices, except the 2520 Punch. See "Unit Record Devices" later in this chapter.

- 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 512, 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	-	558,000 blocks	not supported	558,000 blocks
3375	-	959 cyls.	182 cyls.	959 cyls.
3380	-	885 cyls.	121 cyls.	885 cyls.

- Up to four 2400, 2415, 2420, 3410 (9 track only), 3420, 3430, or 8809 (7 or 9 track) Magnetic Tape Units.

Devices Supported by VM/SP

The following devices are supported by VM/SP 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
- 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

VM/SP supports the following processors:

- IBM System/370 Model 135 Submodel 3
- IBM System/370 Model 138
- IBM System/370 Model 145
- IBM System/370 Model 145 Submodel 3
- IBM System/370 Model 148
- IBM System/370 Model 155 II
- IBM System/370 Model 158 UP/AP/MP
- IBM System/370 Model 158 Submodel 3
- IBM System/370 Model 165 II
- IBM System/370 Model 168 UP/AP/MP
- IBM System/370 Model 168 Submodel 3
- IBM 4321
- IBM 4331 All models
- IBM 4341 All models
- IBM 3031 UP/AP
- IBM 3032 UP
- IBM 3033 UP/AP/MP
- IBM 3042 AP Model 2
- IBM 3033 Model Group N
- IBM 3033 Model Group S
- IBM 3081 Model D16

Processor Required Features and Facilities

Processor features and facilities required by VM/SP are listed below. Only features and facilities that are not standard on a particular processor are described. For example, the Word Buffer feature is standard only on the Model 148; therefore, the feature number and requirements are described only for the Models 145 and 145-3.

- The System Timing facility (#2001), which includes the clock comparator and the processor timer, on the Models 135 and 145.

- The clock comparator and processor timer (#2001) on the Model 145-3.

- The Floating-point feature

  - Model 135, feature #3900
  - Model 145, feature #3910

- The Extended Precision Floating feature (#3840) on the Model 135-3.

- 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.

- The Word Buffer feature (#8810) is required on System/370 Model 145-3. It is also required on Model 145 if:
  - A 2305 Model 2 Fixed Head Storage device is attached.
  - A 3340, 3344, or 3350 Direct Access Storage Facility is attached.
  - A 3330 configuration includes an integrated file adapter and two selector channels, or three or more selector channels.

    Note: This feature is recommended for selector channels if 2314, 3330, 3340, or 3350 devices are attached.

Desirable Features

The following processor features are desirable for VM/SP:

- Virtual machine assist improves performance of VM/SP systems that run virtual storage operating systems in virtual machines. The manner in which virtual machine assist and VM/370:ECPS (see below) are supported on the various VM/SP processors is detailed under "Using Performance Options" in Chapter 10.
- Extended Control Program Support improves performance of VM/SP through CP assist and expanded virtual machine assist capabilities.
- The Extended Floating Point feature improves execution of programs that use Extended Floating Point instructions under VM/SP on Models 135, 155 II, and 158.

  For Model 135, feature #3840
  For Model 155 II, feature #3700
  For Model 158, feature #3700
- The APL Assist feature provides performance assistance when used with the VS APL program product. It is available as hardware feature #1005 on System/370 Models 135 and 145.
- The Conditional Swapping feature provides additional instructions required for execution of VTAM programs. It is available as feature #1051 on System/370 Models 135 and 145.
- The Advanced Control Program Support feature is available only on System/370 Model 145 as feature #1001. It provides additional instructions required for the execution of MVS (OS/VS2 Release 2 and above) and/or VTAM.

  Note: The Conditional Swapping feature and the Advanced Control Program Support feature are mutually exclusive.

- The ECPS Expansion Feature (#1601) is available on the IBM 4341 Processor Model Group 2 and 12. 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) interconnects two channels (either S/360, S/370, or 4341 processor). Only one of the processors requires this feature.

- The IBM 3088 Multisystem Communication Unit is an input/output (I/O) device used to interconnect as many as eight processors using block multiplexer channels. The 4341, 303x, 3042 Attached Processor Model 2, and 308x processors will support the 3088.

- 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 higher data rate. This feature is standard on the 3081 and 4341 processors. Please refer to the appropriate processor manual to determine which channels support data streaming.

Direct Access Storage Devices

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

- IBM 2305 Fixed Head Storage, Models 1 and 2
- IBM 2314 Direct Access Storage Facility
- IBM 2319 Disk Storage
- IBM 3310 Direct Access 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 3370 Direct Access Storage
- IBM 3375 Direct Access Storage
- IBM 3380 Direct Access Storage.

All of these direct access devices are supported as VM/SP system residence, paging, and spooling devices and as virtual devices for use by virtual machines. All are supported as dedicated devices. All except the 2305 are supported by CMS.

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

- IBM 3345 Storage and Control Frame Models 3, 4, and 5 on the Models 145, 145-3, and 148 with the standard ISC for:
  - 3330 Models 1 and 2
  - 3333 Models 1 and 11
  - 3340 Model A2 and 3344 Model B2
  - 3350 Model A2
- IBM 2835 Storage Control Models 1 and 2 for 2305 Models 1 and 2.
- IBM 2844 Auxiliary Storage Control for 2314 and 2319.
- IBM 3830 Storage Control Model 1 for 3330 Models 1 and 2 only.
- IBM 3830 Storage Control Model 2 for 3333 Models 1 and 11, 3340 Model A2, and 3350 Model A2.
- IBM 3830 Storage Control Model 3 for 3330 Models 1 and 11, and 3333 Models 1 and 11.
- IBM 3880 Storage Control Model 1 for 3330 Models 1, 2, and 11, 3333 Models 1 and 11, 3340 Model A2, 3350 Models A2 and A2F, 3370 and 3375.
- IBM 3880 Storage Control Model 2 for 3330 Models 1, 2, and 11, 3333 Models 1 and 11, 3340 Model A2, 3350 Models A2 and A2F, 3370, 3375, and 3380s on the 303x processor with the Data Streaming Feature (#4850).
- IBM 3880 Storage Control Model 3 for 3380s on the 303x processor with the Data Streaming Feature (#4850).
- IBM Integrated File Adapter (#4650) on System/370 Models 135 and 145 for 2319s.
- IBM Integrated File Adapter (#4655) on the System/370 Models 135, 135-3, and 138 or the IBM Integrated Storage Control #4660) on the System/370 Model 145, 145-3, and 148 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 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 for:
  - 3330 Models 1 and 2
  - 3333 Models 1 and 11
  - 3340 Model A2 and 3344 Model B2

- 3350 Model A2

Special Features Required with the 3350

Expanded Control Store special feature (#2151) provides additional control storage for microprogramming use. It is a prerequisite for 3350 disks attached to the 3830 Model 2, or for the 3345 Integrated Storage control units Models 3, 4, and 5 attached to a System/370 Model 145, 145-3, 148, 158 or 168.

The Control Store Extension feature (#2150) is a prerequisite for feature #2151.

Notes:

1. IBM 3330 Model 11 can be used as a system generation device in the same way as the 3330 Models 1 and 2, since the starter system does not use cylinders 404-807.

2. System/370 Models 145, 145-3, and 148 must have the Word Buffer feature (#8810) installed in order to attach a 3330, 3340, 3350 or 2305 Model 2. As noted earlier, this feature is standard on the Model 148.

Desirable Features

3880 Storage Control Unit Buffer Features:

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

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

  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, CP will permit execution of extended count-key-data channel programs.

These features modify the direct access data transfer path by adding a buffer feature 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 145, 148, 155-II, 158, 153-3, 165-II, 168, 168-3, 3031, 3032, 3033, 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

- 3.0 megabyte block multiplexer channels on 3031, 3032, 3033, and 3042 Model 2 when these processors are equipped with the Data Streaming Feature (#4850)

- 3.0 megabyte block multiplexer channels on 4341, 3081, and 3083 Processors

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.

Magnetic Tapes

The magnetic tape devices supported by VM/SP are:

- IBM 2401, 2402, 2403, and 2404 Magnetic Tape Units
- 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
- IBM 3430 Magnetic Tape Unit and Control
- IBM 8809 Magnetic Tape Unit

The magnetic tape control units supported by VM/SP 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 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 4 (available on processor Models 138 and 148 only) and Model 5
- IBM 3211 Printer (Right Indexing only)
- IBM 3213 Printer (in 3215 Emulator Mode)
- IBM 3262 Printer Models 1, 3, 5 (in 3262 Model 1 Emulator Mode), 11, and 13
- IBM 3287 Printer Models 1, 1C, 2, and 2C
- IBM 3289 Model 4 Printer
- IBM 3800 Printer Model 1
- IBM 3800 Printer Models 3 and 8 (dedicated only)
- IBM 4245 Printer

- IBM 4250 Printer (dedicated only).

VM/SP 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 supports the following unit record control units:

- IBM 2821 Control Unit
- IBM 3811 Printer Control Unit
- IBM Integrated Printer Adapter (IPA) on the System/370 Model 135
- IBM Integrated Printer Adapter Basic Control (#4670), and one of the following on the models 135-3 and 138:
  - 1403 Printer Models 2 or N1 Attachment (#4672)
  - 1403 Printer Model 7 Attachment (#4677)
- IBM Integrated 3203 Model 4 Printer Attachment, first printer (#8075) and optionally, second printer (#8076) on the Model 138 and 148.

Terminals

The following system consoles are supported by VM/SP 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 Models 138 and 148 in printer-keyboard mode (3286 printer required)
- 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 as virtual machine consoles (in 3215 emulation mode or in 3270 mode):

- IBM System Console for the System/370 Models 138 and 148 in display mode
- IBM System Console for the System/370 Model 158 in display mode

- IBM 3036 Console with the 3031, 3032 or 3033 processor
- IBM 3278 Model 2A Console (in display mode) with the 4300 and 3081 processors (3287 printer optional)
- IBM 3279 Model 2C Console (in display mode) 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 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 3232 Keyboard-Printer Terminal Model 51
- 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.

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

- 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, 3A, and 3B via 3274 Control Unit Models 1C and 51C
- IBM 3279 Color Display Station Models 2A and 2B via 3276 Control Unit Models 2, 3, and 4
- IBM 3279 Color Display Station Models 3A and 3B via 3276 Control Unit Models 3 and 4
- IBM 3767 Communications Terminal Models 1 and 2 (operating as a 2741).

The following system consoles are supported by VM/SP as virtual system consoles (supported as 3270 consoles):

- IBM System Console for the System/370 Models 138 and 148 in display mode
- IBM System Console for the System/370 Model 158 in display mode
- IBM 3036 Console with the 3031, 3032, or 3033 processor
- IBM 3278 Model 2A Console (in display mode) with the 4300 processors.

The following terminals are supported by VM/SP as virtual system consoles (supported as 3270 consoles):

- IBM 3277 Display Station, Model 2, via 3272 Control Unit, Model 2 (local attachment only)
- IBM 3277 Display Station, Model 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 3279 Color Display Station Models 2A, 2B, 3A, and 3B via 3274 Control Unit Models 1B and 1D (local attachment).

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 that 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 uses a new line for each one.

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.

Prior availability of an RPQ does not guarantee or imply current or future availability. Contact your IBM branch office for ordering information concerning the RPQs mentioned in this book.

2741 Features: Required and Desirable 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:

- 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)
- For switched lines, the Data Set Attachment (#9114) and Dialup feature (#3255) are required.
- For point-to-point nonswitched lines, one of the following features is required:
  - Data Set Attachment (#9115 duplexed for facility D1)
  - Data Set Attachment (#9116 duplexed for facility B2)
  - Data Set Attachment (#9120 duplexed for facility B1 or D1)
  - 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).

The following features, although not required, heighten the convenience and usability of the terminal:

- Print Inhibit (#5501)
- Red Ribbon Control RPQ #868019 (supported for PTTC/EBCD keyboard only)
- Typamatic Keys (#8341)
- Pin Feed Platen (#9509).

1050 Control Units, Models, and Features: Supported, Required, and Desirable 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)

The following features, although not required, heighten the convenience and usability of the terminal:

- Automatic Ribbon Shift and Line Feed Select (#1295)
- Automatic EOB on Carrier Return RPQ #E28235.

3270 Information Display System Terminals : 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 also designed to attach the Category B terminals with certain limitations. A maximum of 16 of the 32 attachable terminals on a 3274 can be Category B terminals.

ATTACHABLE TERMINALS

Category A Terminals

3262 Line Printer Models 3, 13
3278 Display Station Models 1, 2, 3, 4, 5
3279 Color Display Station Models 2A, 2B, 3A, 3B
3287 Printer Models 1, 2, 1C, 2C
3289 Line Printer Models 1, 2
4250 Printer

Category B Terminals

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

The basic 3271/3272 Control Unit permits attachment of up to 4 devices. Up to 32 devices may be attached in sets of four devices by adding up to seven Device Adapters (#3250).

The basic 3274 Control Unit permits attachment of up to 8 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 on a 3274 Control Unit. Each adapter provides for the attachment of an additional 8 Category A terminals. These terminal adapters must be installed in sequence:

Terminal Adapter Type A1 - Category A terminals 9-16
Terminal Adapter Type A2 - Category A terminals 17-24
Terminal Adapter Type A3 - Category A 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 to a 3274 Control Unit, 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 each. A maximum of one each of the B1, B2, B3 and B4 adapters can be installed for a combined total of sixteen Category B terminals on a 3274 Control Unit. These terminal adapters must be installed in sequence:

Terminal Adapter Type B1 - Category B terminals 1-4
Terminal Adapter Type B2 - Category B terminals 5-8
Terminal Adapter Type B3 - Category B terminals 9-12
Terminal Adapter Type B4 - Category B terminals 13-16

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

For additional information on 3270 Information Display Terminals, see the *3270 Information Display System Library User's Guide.*

Local Configurations Supported

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

IBM 3272 Control Unit Model 1 and 2 for attachment of up to 32 Category B terminals.

These 3272 configurations require:

- Device Adapter feature (#3250) if more than four devices are attached to the 3272. Up to four additional devices can be attached with each device adapter.
- A 3271/3272 Attachment (#8330) to attach each 3287 Printer.

IBM 3274 Control Units Model 1B, 1D, 21B, 21D, 31D for the attachment of up to 32 display stations and printers. All of the 32 devices can be Category A Terminals. The 3278 Display station Model 5 is not supported with the 3274 Control Unit Model 1B. A maximum of 16 of the 32 devices can be Category B terminals.

Remote Configurations Supported

CONTROL UNITS: The following control units can be remotely attached to leased lines via a:

- 2701 Data Adapter Unit
- 2703 Transmission Control Unit
- 3704/3705 Communications Controller in emulation mode
- Integrated Communication Adapter (ICA) :

  IBM 3271 Control Unit Model 2 for attachment of up to 32 Category B terminals.

  This configuration requires:

  - Device Adapter feature (#3250) if more than four devices are attached to the 3271. Up to four additional devices can be attached with each device adapter.

  - A 3271/3272 Attachment (#8330) is required to attach each 3287 Printer Model 1 or 2.

  - Copy feature (#1550) is required to use the full screen copy function.

  - Transmission Speed feature (#7820 or #7821).

  IBM 3271 Control Unit Model 11 and 12 for attachment of up to 32 Category B terminals.

  IBM 3274 Control Unit Model 1C, 21C, 31C for the attachment of up to 32 display stations and printers. All of the 32 devices can be Category A terminals. A maximum of 16 of the 32 devices can be Category B terminals.

  IBM 3274 Control Unit Model 51C for the attachment of up to 12 display stations and printers. Eight of the twelve devices can be Category A terminals. Up to four Category B terminals can be attached via Terminal Adapter Type B1.

  IBM 3276 Control Unit Display Station Models 2, 3, and 4 for attachment of up to 7 additional:

  - 3278 Display Stations Models 2, 3, and 4. 3278 Model 3 cannot be attached to a 3276 Model 2. 3278 Model 4 cannot be attached to a 3276 Model 2 or 3.

  - 3279 Color Display Stations Models 2A, 2B, 3A, 3B. 3279 Models 3A and 3B cannot be attached to 3276 Model 2.

  - 3287 Printers Models 1, 1C, 2, 2C.

    Note: Extended color printing, highlighting, and programmed symbols are not supported for the 3276.

  - 3289 Printer Models 1 and 2.

To support this configuration, the following is required:

- The basic 3276 Control Unit Display Station contains 1 integral display station, and permits attachment of one of the following:
  - 3278 Display Station Model 2, 3, or 4
  - 3279 Model 2A, 2B, 3A, or 3B. 3279 Models 3A and 3B cannot be attached to a 3276 Model 2.
  - 3287 Printer Models 1 or 2
- A 3274/3276 Attachment (#8331) is required for each 3287 Printer Model 1 or 2.
- Each 3276 requires one of the Communications features (#6301 or #6302) and either the External Modem Interface (#3701) or the 1200 BPS Integrated Modem feature (#5500).
- Color Display Attachment (#1950) permits attachment of 3279 Color Display Terminal Models 2A, 2B, 3A, and 3B. This feature is not available for the 3276 Models 1 and 2. The 3276 does not support programmable symbol sets, extended color, or extended highlighting. 3279 Models 2B and 3B are supported on the 3276 for base color. Extended Function Base (#1068) is prerequisite.

The following control unit is remotely attached to either leased or switched lines via a:

- 2701 Data Adapter Unit
- 2703 Transmission Control Unit
- 3704/3705 Communications Controller in emulation mode
- Integrated Communication Adapter (ICA):

  IBM 3275 Display Station Model 2 stand-alone control unit and display station. A 3284 Printer Model 3 or 3286 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. The 3275 Dial feature does not support full screen read/write.
  - Transmission Speed feature (#7820 or #7821)
  - A Printer Adapter feature (#5550), to attach a 3284 Printer Model 3
  - RPQ MB4317 is required to attach a 3286 Printer Model 3

3767 Features: Required and Desirable 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 or 3705 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).

The following features, although not required, heighten the convenience and usability of the terminal:

- Alternate Character Set (#1291), plus a defined character subset for the keyboard:
  - If the primary character set is Correspondence (#9381), the alternate character set can be APL (#9383) or EBCDIC (#9382).
  - If the primary character set is EBCDIC (#9391), the alternate character set can be APL (#9393) or Correspondence (#9392).

    Note: Line control is PTTC/EBCD with this feature.
- Acoustic Coupler (#1110) at 300 bps.

Transmission Control Units

VM/SP supports the following transmission control units:

- IBM 2701 Data Adapter Unit
- IBM 2702 Transmission Control
- IBM 2703 Transmission Control
- IBM Integrated Communications Attachment (ICA), (#4640)
- IBM 3704, 3705-I, and 3705-II Communications Controllers.

2701 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 Required and Optional 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 Required and Optional 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)

  - Pluggable End Characters (return key generates an interrupt) RPQ #E66707, optional.

- 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.

IBM Integrated Communications Attachment (ICA) Required and Optional Features

The ICA (#4640) is available on the System/370 Models 135, 135-3, and 138. Additional lines (#4722-4728) are supported.

- For 1050, 2741, and 3767 (as a 2741) terminals, the following are required:

  - Terminal Adapter Type I Model II (#9721-9728)

  - Switched Network Facility (#9625-9632), optional

  - Write Interrupt (#9745-9752)

  - Read Interrupt (#9737-9744)

  - Unit Exception Suppression (#9729-9730), optional

  - For the 3767 only, as a 2741, 200 bps (#2711-2718) or 300 bps (#9593-9600).

- 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 (#9649-9656)
  - Half-Duplex Facility (#9617-9624)
  - EBCDIC Transparency (#9673-9680).

- For line control of CPT-TWX (Model 33/35) terminals and 3101 display terminals, the following are required:

  - Telegraph Adapter Type II (#9785-9792)
  - Switched Network Facility (#9625-9632).

3704/3705 Required Features

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

Note: VM/SP 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 or 3705.

VM/SP supports all models of 3704 and 3705 Communications Controllers. The features required on a communications controller do not depend on VM/SP. Other 3704/3705 features depend on the planned use of the communications controller and the type of 3704/3705 control program (emulation) to be run.

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

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

Other Considerations for Planning Your Configuration

Two-Channel Switch

If any I/O devices controlled by VM/SP 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 "VM/SP Using Channel Switching" in Chapter 8 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.

Devices Used Only by an Operating System in a Virtual Machine and Not by VM/SP

Any input/output device that can be attached to the processor can be used by a virtual machine under VM/SP as long as there are:

- No timing dependencies in the device or the program
- No dynamically modified channel programs except OS Indexed Sequential Access Method (ISAM) or OS/VS Telecommunications Access Method (TCAM) Level 5
- No violations of the other restrictions outlined in "Appendix D. VM/SP Restrictions."

Dynamically modified channel programs (except those that have input/output involving page zero) are permitted if run in a virtual=real machine.

Input/output devices that are part of a virtual machine's configuration require real device equivalents, except for:

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

Service Record File

On 3031, 3032, and 3033 processors, each console station of the 3036 system console has a 7443 diskette attached to it, which is usable 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. When 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.

Multiple Service Record Files

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 systems support more than one service record file device. For VM/SP systems operating on a 3033AP or 3033MP, specify more than one SRF device 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=cuu operand in both macro statements. Identify the SRF device addresses in the RIOGEN macro statement as SRF=(cuu,cuu,...). 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 systems with I/O configured asymmetrically to one processor, a channel path must be available from the I/O processor to both SRF devices in order to access the SRF devices in both 3036 consoles.

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 uses 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 define this information to the controller by running the Input/Output Configuration Program. For more information about the Input/Output Configuration Program, see "Input/Output Configuration Program" in Chapter 8 and "Considerations for Coding the Input/Output Configuration Source File" in Chapter 19.

Chapter 3. Estimating VM/SP Storage Requirements

This section contains information about:

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

"Specifying a Virtual=Real Machine" in Chapter 10, includes information about estimating real storage requirements for a virtual=real machine.

Real Storage Requirements for CP

Figure 2 lists various CP requirements and the amount of real storage required for each.

CP Requirement	Real Storage Allocated
Resident nucleus	Approximately 235.5K[3]
Internal trace table	Conventionally, 4K of storage is allocated for each 256K of real storage. This storage is set aside at IPL time. See the "SYSCOR Macro" in Chapter 21 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: • 104 bytes/real device • 80 bytes/real control unit • 96 bytes/real channel • 40 bytes for each remote 3270 or real 3704/3705
Permanently allocated free storage (virtual control blocks and tables). For installation control of free storage, use the SYSCOR macro. For the format of the SYSCOR macro see Chapter 21.	The default value is a minimum of 12K, plus an additional 4K for each 64K of real storage above 256K[4]. 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.

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

---

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

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

CP Requirement	Real Storage Allocated
	The storage required is: • 568 bytes for the VMBLOK • 64 bytes/1M of virtual storage for the SEGTABLE • 40 bytes/64K of virtual storage for the PAGTABLE • 136 bytes/64K of virtual storage for the SWPTABLE • 72 bytes/virtual device • 40 bytes/virtual control unit • 48 bytes/virtual channel

Figure 2 (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

you would estimate CP real storage requirements as follows:

235.5K	for the CP resident nucleus
16K	for the CP internal trace table (see the SYSCOR macro in Chapter 21)
4K	for the real control blocks, calculated as follows: 104 X 29 = 3016 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: 3016 + 480 + 288 = 3784 bytes (approximately 4K)
60K	for permanently allocated free storage (default value)
315.5K	real storage required

Also, as each of the 12 (512K) virtual machines defined logs on, approximately 3.0K of real storage is allocated to each from the permanently allocated free storage. A breakdown of this 3.0K of real storage is:

568	bytes for a VMBLOK
64	bytes for the SEGTABLE
320	bytes for the PAGTABLE
1088	bytes for the SWPTABLE
72	bytes for a virtual reader
72	bytes for a virtual printer
72	bytes for a virtual punch
216	bytes for three virtual disks
144	bytes for three virtual channels
72	bytes for a virtual machine console
120	bytes for three virtual control units
2808	bytes for each of the logged-on users defined

See "Specifying the Amount of Virtual=Real Space" in Chapter 10, for an example of estimating storage requirements and determining the maximum virtual=real area size.

Reducing the CP Nucleus Size

You can use the small CP option to reduce the size of the CP nucleus. A response of "YES" to the GENERATE EXEC question:

```
DO YOU WANT THE SMALL CP OPTION?--RESPOND (YES|NO)
```

during the system generation process, causes CPLOADSM EXEC to be used in place of the CPLOAD EXEC. The CPLOADSM EXEC removes V=R support, AP/MP support, and support for the following:

Support	**Number of Bytes***	**Modules Removed**
Missing interrupts	1,700	DMKDID
MVS Guest	7,200	DMKFPS, DMKQVM, DMKVSC[5]
SNA(CCS)	14,100	DMKVCP, DMKVCR, DMKVCT, DMKVCV, DMKVCX
TTY terminal support	300	DMKTTZ
3066	1,500	DMKGRH
Remote 3270	14,600	DMKRGA, DMKRGB, DMKRGC, DMKRGD
3340 Alternate Track	1,800	DMKTRK
3375/3380	4,700	DMKDAD
3704/3705	6,300	DMKRNH
3850 MSS	3,800	DMKSSS, DMKSSU
3800 printers	1,400	DMKVSV

*Approximate

---

[5] Removal of module DMKVSC also removes V=R support.

Removal of the support listed in the preceding table reduces the CP resident nucleus by approximately 57,400 bytes. If you need any of the support listed in the table, reply "NO" when asked:

```
DO YOU WANT THE SMALL CP OPTION?--RESPOND (YES|NO)
```

Note: The GENERATE and VMFLOAD commands, and the CPLOAD and CPLOADSM loadlists are described in the *VM/SP Installation Guide*. If you want a smaller CP nucleus, but require some of the function removed by using the small CP option, you can tailor the loadlist to your own specifications. Edit the CPLOAD loadlist to remove only modules that are associated with functions (refer to the small CP option table above) that you do not require. The CP nucleus will be reduced by the amounts shown in the Small CP option table.

The graphic device support for locally attached terminals is handled by the module DMKGRF. Removal of local graphics support is not a part of the Small CP option. If you do not require local graphics support, you may remove the module DMKGRF and reduce the CP resident nucleus by approximately 10,300 bytes.

Caution should be exercised before removing the modules from the loadlist. 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 loadlist. If you do remove those modules, unpredictable results may occur.

The following names are undefined during the VMFLOAD procedure if DMKTRK is removed from the loadlist:

```
DMKTRKIN     DMKTRKFP     DMKTRKVA
```

The following names are undefined during the VMFLOAD procedure if DMKRNH is removed from the loadlist:

```
DMKRNHIC     DMKRNHND     DMKRNHTR     DMKRNHIN     DMKRNH
```

The following names are undefined during the VMFLOAD procedure if DMKRGA, DMKRGB, DMKRGC, and DMKRGD are removed from the loadlist:

```
DMKRGADH     DMKRGBEN     DMKRGBIC     DMKRGDOB
DMKRGAIN     DMKRGBFM     DMKRGC       DMKRGDOI
```

The following names are undefined during the VMFLOAD procedure if DMKSSS and DMKSSU are removed from the loadlist:

```
DMKSSSAS     DMKSSSHR     DMKSSSMQ     DMKSSSVM
DMKSSSCA     DMKSSSL1     DMKSSSNS     DMKSSUCF
DMKSSSCV     DMKSSSL2     DMKSSSNV     DMKSSUI1
DMKSSSDE     DMKSSSL3     DMKSSSRL     DMKSSUI2
DMKSSSEN     DMKSSSLN     DMKSSSVA     DMKSSULO
```

The following names are undefined during the VMFLOAD procedure if modules DMKFPS, DMKVSC and DMKQVM are not on the loadlist:

```
DMKQVMRT     DMKQVMEP     DMKVSCVR     DMKQVMTS     DMKQVMCU
DMKFPS       DMKVSC
```

The following names are undefined during the VMFLOAD procedure if DMKVCP, DMKVCR, DMKVCT, DMKVCV, and DMKVCX are removed from the loadlist:

```
DMKVCPIL     DMKVCRWT     DMKVCTER     DMKVCTCN     DMKVCTRM
DMKVCRNR     DMKVCTLO     DMKVCTEN     DMKVCTSV     DMKVCXIO
DMKVCRRD     DMKVCTCH     DMKVCTDA     DMKVCTQS     DMKVCRMT
DMKVCVER
```

The following name is undefined during the VMFLOAD procedure if DMKDAD is removed from the loadlist:

```
DMKDADER
```

The following name is undefined during the VMFLOAD procedure if DMKTTZ is removed from the loadlist:

```
DMKTTZLF
```

The following name is undefined during the VMFLOAD procedure if DMKVSV is removed from the loadlist:

```
DMKVSVLD
```

The following names are undefined during the VMFLOAD procedure if DMKGRF is removed from the loadlist:

```
DMKGRFIN     DMKGRFMT     DMKGRFIC     DMKGRFEN     DMKGRF
```

The following name is undefined during the VMFLOAD procedure if DMKGRH is removed from the loadlist:

```
DMKGRHIN
```

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.

The following names are undefined during the VMFLOAD procedure when you remove DMKDID from the load list:

```
DMKDIDDA     DMKDIDTA     DMKDIDMS     DMKDIDEP     DMKDIDTR
DMKDIDGR     DMKDIDUR
```

Direct Access Storage Requirements for CP

In the following paragraphs and in "Part 2. Defining Your VM/SP System," there are many references to "DASD space." With the support of fixed block DASD architecture, it is important to understand the fundamentals of "DASD space" to avoid confusion when dealing with various DASD types.

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). CP also 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.

CKD Device Geometry

CP views CKD devices by their geometric characteristics, eg. as a certain number of cylinders. Each cylinder has a fixed page capacity, meaning that a fixed 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 "Chapter 21. Preparing the CP System Control File (DMKSYS)," you are asked to figure your particular DASD requirements as a number of records or pages, then convert this number to an equivalent number of cylinders. This means 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 also 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)/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 figure your DASD space requirements as a number of pages, you can use these numbers directly in the system generation macros and in the Format/Allocate service program. No conversion to other units is required.

Although convenient for CP DASD space, this causes an inconvenience when assigning minidisks because of 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 your available space in block numbers. Be careful that you provide input to Format/Allocate in page numbers and assign minidisks by block number.

To obtain the corresponding 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 below, pages 2000 through 9999 were allocated as PERM space for use as minidisks. The allocation results would show:

```
PERM    2000    9999
```

This corresponds to block numbers 16000 through 79999 by doing the following:

For the beginning block number:

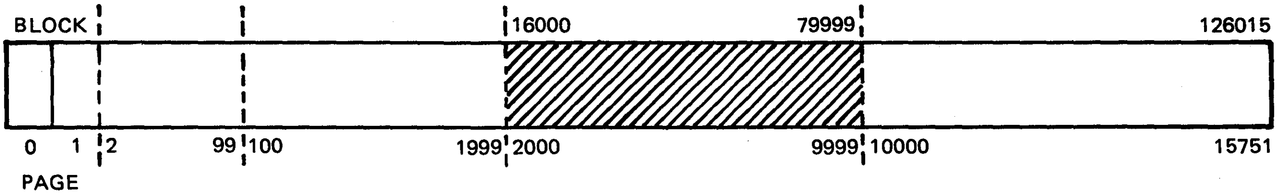
Multiply 2000 (1st page) X 8 (8 blocks per page) =
16000 (beginning block Number)

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)

The block numbers must be used on the MDISK statement.



Pages 0 and 1 are reserved (see the *VM/SP Operator's Guide* for details)
Pages 2-99 for directory (DRCT)
Pages 100-1999 for nucleus and error recording (PERM)
Pages 2000-9999 for 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 FB-512 16000 2000 LABEL
MDISK 19F FB-512 18000 5000 LABEL
MDISK 296 FB-512 23000 7000 LABEL
```

DASD Space Requirements

Figure 3 shows minimum 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.

	3330	2305	3340	3350	3375/ 3380	FB-512
CP Nucleus	varies	varies	varies	varies	varies	varies
Error Recording[6]	2	2	2	2	2	114
Warm Start	1	1	1	1	1	57
Checkpoint Start	1	3	3	1	1	57
Directory	4	2	10	2	4/2	228
Saved Systems	varies	varies	varies	varies	varies	varies
Paging Space	18	40	40	10	10/11	1000
Spooling Space	30		70	15	15/20	1700
Total System[7]	56 cyl	48 cyl	126 cyl	31 cyl	33/37 cyl	3156 pgs

Figure 3. DASD Space Requirements by DASD Type

CP Nucleus DASD Requirements

The CP nucleus (without a virtual=real area) requires about 220 pages of disk space for resident and pageable functions.

---

[6] The default is 2 cylinders but up to 9 cylinders may be specified via the SYSERR operand of the SYSRES macro.

[7] These figures do not include space for the nucleus or saved systems.

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 DC (hexadecimal), 220 pages of disk space are required for CP nucleus residence, since hex DC converts to decimal 220. 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:

10 cylinders on a 2305 or 3340
4 cylinders on a 3330 or 3333
2 cylinders on a 3350
3 cylinders on a 3375
2 cylinders on a 3380

The amount of space required on FB-512 devices is equal to the number of pages as computed above.

Error Recording DASD Requirements

Error recording space varies from 2 to 9 cylinders and is established by the SYSERR operand of the SYSRES macro instruction as described in Chapter 21.

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 21.

Checkpoint Start Data DASD Requirements

The space required for dynamic checkpointing of the VM/SP spool file system is discussed under the description of the SYSRES macro in Chapter 21.

Dump Space DASD Requirements

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

The following formula may be used to approximate the amount of DASD dump space needed:

```
Number of pages (FBA)       =  number of pages real memory + 5


Number of cylinders (CKD)   =  number of pages real memory + 5
                               -------------------------------
                                   number of pages/cylinder
```

The number of pages per cylinder for the various CKD DASD devices is listed under the heading "Paging and Spooling DASD Requirements" later in this chapter.

Allocating DASD Space for the VM/SP Directory

The VM/SP directory normally requires two cylinders so that it can be rewritten without disturbing the active directory and swapped after a successful update.

Before you create a VM/SP 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 VM/SP directory. The space must be allocated DRCT. To calculate the total space required, first calculate the total number of records used:

$$NR = \frac{NU}{169} + \frac{((NU + NM) \times 2) + \text{all other control statements}}{170}$$

where:

NR = total number of records used
NU = number of USER control statements
NM = number of MDISK control statements (except for temporary disks)

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

Note: You should initially format and allocate enough space for two VM/SP 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 VM/SP 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, the 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 succeeding directories to be swapped. 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.

Saved System DASD Requirements

Saved systems require one page for each page saved, plus an additional information page. However, a 3704/3705 may require up to four additional information pages.

Saving one copy of the CMS system requires:

5 cylinders on a 2314 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 3 on page 38 are examples.)

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 will yield adequate paging space.

number of logged-on users x average virtual machine size
(in 4k 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 7 cylinders of a 3380 (994/150 = 6.6). 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 on page 38 assumed a maximum of 200 spool files of 8-9 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 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 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 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 run the VSE logical transients, and for buffers and work areas. VSAM issues a GETVIS macro to request free storage.

If CMS/DOS is entered with the VSAM option

```
SET DOS ON (VSAM
```

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

Estimating DASD Storage Requirements for CMS Minidisks

The following information is provided 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. 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 Storage Requirements

CMS requires virtual storage and auxiliary storage. A minimum of 1 megabyte of virtual storage is recommended 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)
  - 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) —

  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,
  74 cylinders on a 3375,
  45 cylinders on a 3380,
  45,568 blocks on an FB-512.

  Note: The CMS system and the CMS and CMSL nuclei reside on the 190 minidisk.

- Resident disk space for application programs (CMS commands, user programs, IBM program products) — the space needed is program-dependent, and must be assigned by you.

- Work space for application programs (CMS commands, user programs, IBM program products) — the space needed is program-dependent, and must be assigned by you.

Device Support

CMS supports the virtual machine devices shown in Figure 4.

Virtual IBM Device Type	**Virtual Address**[1]	**Symbolic Name Default**	**Device Use**
3210, 3215, 1052, 3066, 3270	cuu[2]	CON1	System console
2314, 2319, 3310, 3330, 3340, 3350, 3370, 3375, 3380	190	DSK8	CMS System disk (read-only)
	191[3]	DSK1	Primary disk (user files)
	cuu	DSK2	Minidisk (user files)
	cuu	DSK3	Minidisk (user files)
	192	DSK4	Minidisk (user files)
	cuu	DSK5	Minidisk (user files)
	cuu	DSK6	Minidisk (user files)
	cuu	DSK7	Minidisk (user files)
	19E	DSK9	Minidisk (user files)
	cuu	DSK0	Minidisk (user files)
	cuu	DSKH	Minidisk (user files)
	cuu	DSKI	Minidisk (user files)
	cuu	DSKJ	Minidisk (user files)
	cuu	DSKK	Minidisk (user files)
	cuu	DSKL	Minidisk (user files)
	cuu	DSKM	Minidisk (user files)
	cuu	DSKN	Minidisk (user files)
	cuu	DSKO	Minidisk (user files)
	cuu	DSKP	Minidisk (user files)
	cuu	DSKQ	Minidisk (user files)
	cuu	DSKR	Minidisk (user files)
	cuu	DSKT	Minidisk (user files)
	cuu	DSKU	Minidisk (user files)
	cuu	DSKV	Minidisk (user files)
	cuu	DSKW	Minidisk (user files)
	cuu	DSKX	Minidisk (user files)
2540, 2501, 3505	00C	RDR1	Virtual reader
2540, 3525	00D	PCH1	Virtual punch
1403, 1443, 3203, 3211, 3262, 3800, 4245, 3289-4	00E	PRN1	Line printer
2401, 2402, 2403, 2415, 2420, 3410, 3411, 3420, 3430, 8809	181-4	TAP1-TAP4	Tape drives

Figure 4. Devices Supported by a CMS Virtual Machine

[1] The device addresses shown are those that are preassembled into the CMS resident device table. These need only be modified and a new device table made resident to change the addresses.

[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 initial program load (IPL).

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

CMS Disks

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*.

CMS Libraries

CMS supports simulated partitioned data sets that contain:

- CMS and OS macro/copy files to be used at compilation or assembly time (source/macro libraries). The CMS filetype for these files is MACLIB.
- Object routines to be referred to at load and/or execution time (text libraries). The CMS filetype for these files is TXTLIB.
- 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 thses files is DOSLIB.

System Macro Libraries

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

Library	**Contents**
DMSSP MACLIB	CMS and DOS macros versioned by VM/SP
CMSLIB MACLIB	CMS macros 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 macros distributed in SYS1.TSOMAC
DOSMACRO MACLIB	Internal macros used by CMS/DOS support routines

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 macros in CMS/DOS, you must first create a CMS macro library that contains all the VSE macros you need. The *VM/SP Installation Guide* contains a procedure for copying an entire macro library. The procedure for copying individual macros 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

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, and the PROPLIB LOADLIB, which supports the Programmable Operator Facility.

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 use the CP commands available to all virtual machines under VM/SP directly from CMS. Using these CP commands, you can send messages to the operator or to other users, change your virtual machine's configuration, and use 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 languages available under CMS include:

S/370 Assembler
VS BASIC
PL/I
OS FORTRAN IV
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 run under CMS, see "Appendix A. Licensed Programs and Integrated Emulators."

CMS runs 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 use the compilers within the conversational environment of CMS.

Limited Support of OS and VSE in CMS

Object programs (TEXT files) produced under CMS or OS can be executed under CMS if they do not use certain OS functions not 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 simulates the control blocks, supervisor and I/O macros, linkage editor and fetch routines necessary to compile, test, and run VSE programs under CMS. The support for the VSE user is comparable to that for the OS user.

CMS supports VSAM and Access Method Services for VSE 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 the VSE/VSAM macros and their options and a subset of the OS/VSAM macros.

CMS/DOS support of VSAM is based on the VSE/VSAM program product.

Application programmers who normally use CMS to interactively create, modify, and test programs may require facilities not supported in CMS (for example, an OS program using ISAM). They can alternately run CMS and another operating system in the same virtual machine.

A description of the actual processes for reading OS or VSE files is in the *VM/SP CMS User's Guide*. A description of alternating operating systems is in *VM/SP 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 run any data base description generation and program specification block generation. The data base recovery and reorganization utilities must be run 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 VSE format
- VSAM format

When VM/SP MSS support is installed, and the VM/SP 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 VSE programs running in CMS may read data sets or files on OS or DOS disks, but may not write or update them. OS and DOS minidisks can 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 DSF. Full disks must be initialized using the appropriate OS/VS, DSF, 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 VM/SP directory entry, or it can be acquired 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 your primary read/write work disk. Disks may be 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 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 5.

Field Description	**Byte Displacement Length**	**Contents by Disk Block Size (800 byte)**	**Contents by Disk Block Size (512, 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'512', 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'
Reserved for IBM use	50,2	not used, zeros	not used, zeros
Disk offset when reserved	52,4	not used, zeros	F'n'
Reserved for IBM use	56,24	not used, zeros	not used, zeros

Figure 5. Volume Label Contents for CMS Formatted Disks

On count-key-data devices, each 512-, 800-, 1K-, 2K-, or 4K-byte block (called CMS block in the following discussion) represents one physical record of that size on disk. For FB-512 devices, each CMS block consists of the appropriate number of contiguous 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 fixed length CMS blocks, spanning blocks where necessary. As a file grows or contracts, its space is expanded or reduced as needed.

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^{31}-1)-132,000 disk blocks of data, grouped into as many as 2^{31}-1 logical records, all of which must be on the same minidisk. The maximum number of data blocks available in a variable format file on a 512-byte blocksize minidisk is about 15 times less than 2^{31}-1. This number is the maximum number of data blocks that can be accessed by the CMS file system due to its 5-level tree structure.

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 block 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[8]. 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.

OTHER RESTRICTIONS:	
Maximum number of files per minidisk (exceptions - 2314,2319)	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 6. CMS Disk File Statistics for 800-byte CMS Blocks

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 6 compares the disk devices supported by CMS in the case of 800-byte CMS blocks.

---

[8] The DASD address of the Y-DISK will be whatever was specified when CMS was generated. For the standard system this will be 19E.

Identifying Disk Files

CMS commands can list the identifications of files on CMS and non-CMS formatted disks and minidisks. The LISTFILE and FILELIST commands list 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 four magnetic tape units at virtual addresses 181, 182, 183, and 184. They may be 2401, 2402, 2403, 2415, 2420, 3410/3411, 3420, 3430, or 8809 drives.

Three tape-handling commands (ASSGN, FILEDEF, and TAPE) allow you to specify the modeset of the tape: track (7-track or 9-track), density, and, for 7-track only, 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, eg. FILEDEF, you can issue a CMS TAPE MODESET 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.

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 only apply when you run CMS. VSE 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 VM/SP 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 7. Planning for CMS/DOS."

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 VM/SP, these devices are spooled. CMS does not support real or dedicated unit record devices, nor does it support a virtual 2520 Card Punch. Figure 4 on page 46 lists the devices supported as virtual devices by CMS.

Saving CMS

CMS is designed to be used as a shared system. The default DMKSNT entry places the CMS nucleus from location X'190000' to X'200000'. It is intended that the CMS nucleus is 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. This requirement indicates that the CMS nucleus should be generated at a higher virtual address, or that 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 will construct a segment table that 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'190000' 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. Therefore, you may wish to relocate the CMS nucleus, based upon your 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 that 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 needs. For more information about relocating the CMS nucleus, see the *VM/SP Installation Guide.*

For more information about saved systems, see "Chapter 9. Saved Systems and Discontiguous Segments," and see the *VM/SP System Programmer's Guide.*

Chapter 5. Minidisks

External storage requirements of multiple virtual machines running at the same time would be excessive if each virtual machine were assigned one real direct access storage device for each virtual DASD specified in its configuration. Therefore, if you do not require the full capacity of a real DASD, you can be assigned one or more minidisks instead. A minidisk is a logical subdivision of a physical disk pack with its own virtual device address, virtual cylinders or blocks (starting with 0, 1, 2, and so on), and a VTOC (volume table of contents or disk label identifier). Each minidisk is preallocated the number of contiguous full cylinders or blocks specified in the VM/SP MDISK directory record. That space is considered to be a complete virtual disk device.

Minidisks are controlled and managed by CP. 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 Restrictions."

The remainder of this section describes minidisk:

- Definition
- Space allocation
- Initialization
- Alternate tracks/blocks
- Labels

Defining Minidisks

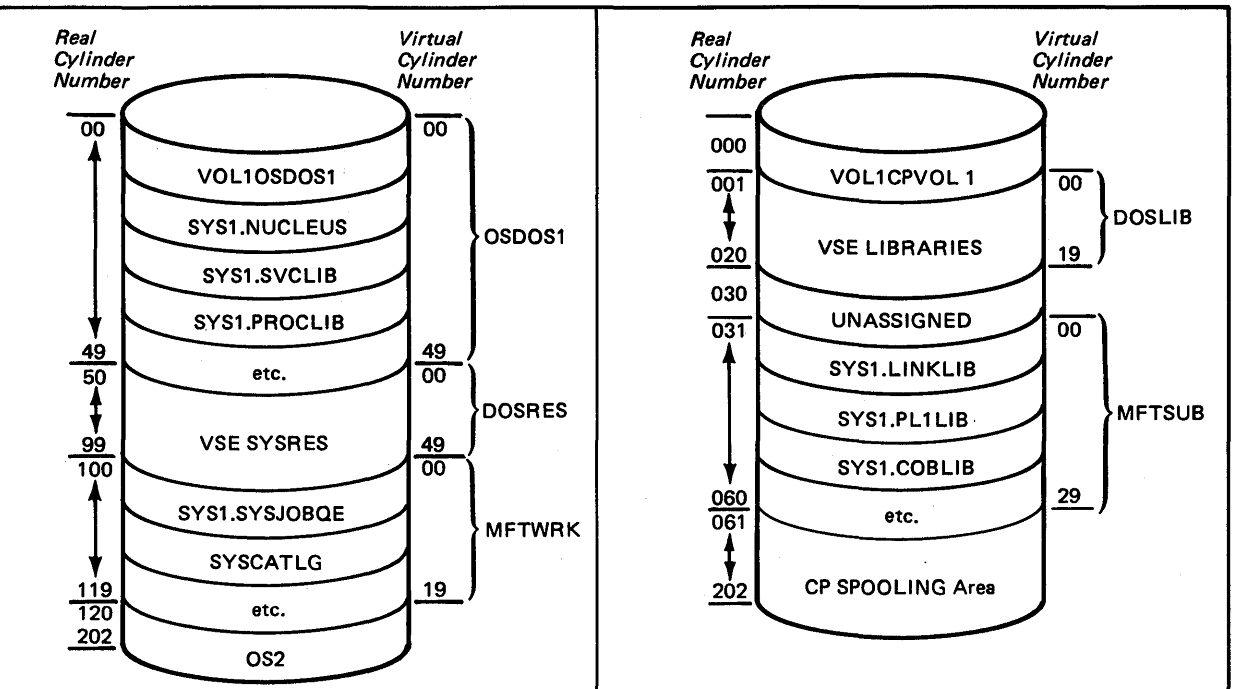
Permanent minidisks are defined in the VM/SP directory entry for a virtual machine. A minidisk defined in the directory via an MDISK statement is a permanent part of the virtual machine configuration. Data on the minidisk is available to you whenever you are logged on.

If any virtual machine has a temporary requirement for direct access space it can be filled from a pool of T-DISK space. You specify the size of the T-DISK pool when you allocate disk space with the stand-alone Format/Allocate program. Minidisks created from the T-DISK area must be initialized, and are available to the virtual machine for the duration of one terminal session. When the virtual machine logs off or issues a CP command to release the temporary minidisk, the area is returned to CP.

It is up to you to allocate minidisks on VM/SP disks in a manner that minimizes arm contention and physical overlap. Information about minimizing arm contention is found in "Chapter 21. Preparing the CP System Control File (DMKSYS)."

Note: The VM/SP 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 7 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 area, and virtual device 231 as a 20-cylinder area on real volume OSDOS1. The directory entry for userid XYZ (a VSE user) describes virtual device 130 as a 50-cylinder disk area on a real volume OSDOS1.



VM/SP User Directory Entry for user ABC (An OS user)

```
USER      ABC    123      512K
  ACCOUNT  985
    CONSOLE  009   3215
    MDISK    230   2314  000  050  OSDOS1 W
    MDISK    231   2314  100  020  OSDOS1 W
    MDISK    232   2314  031  030  CPVOL 1 W
    SPOOL    00C   2540  READER A
    SPOOL    00D   2540  PUNCH A
    SPOOL    00E   1403  A
```

VM/SP User Directory for user XYZ (A VSE user)

```
USER XYZ PASSWORD
  ACCOUNT NUMBER BIN14
    CONSOLE   01F   3215
    SPOOL     C     2540 READER
    SPOOL     D     2540 PUNCH
    SPOOL     E     1403
    MDISK     130   2314 050 050 OSDOS1 W
    MDISK     131   2314 001 020 CPVOL 1 W
```

Note: VM/SP allows cylinders on a 2314 or 2319 normally reserved for alternate track assignment (cylinders 200 to 202) to be optionally used for normal data storage if included within the limits of a minidisk.

Figure 7. Use and Definition of Minidisks

Real volume CPVOL1 also contains disk areas assigned to userid ABC (virtual device address 232) and userid XYZ (virtual device address 131).

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. See the list of restrictions in "Appendix D. VM/SP Restrictions" for more information and an explanation of minidisk restrictions.

Minidisk Space Allocation

A minidisk always begins at virtual cylinder or block zero. For count-key data, it's minimum size is one cylinder unless it is on a 2314 or 2319 disk and is formatted by the Device Support Facility service program. 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, which can be used 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, in increments of one block.

A DASD volume containing multiple minidisks contains some tracks in which the cylinder address in the count fields of records R0 and R1 do not agree with each other. 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.

VM/SP controls the boundaries of minidisks. If an attempt is made to refer to 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.

Note: If the cylinder or block addresses in the MDISK statements overlap each other, the integrity of data in the overlapped cylinders or blocks may be compromised with no error indicated.

OS Minidisks

OS bases all of its space allocation parameters on the volume table of contents (VTOC) label written on each disk. It determines the amount of space available on that volume 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 size of the minidisk. The remaining disk space on the real disk appears to OS to be permanently dedicated, and not assignable by OS space accounting routines. The Device Support Facility service program should be used to format minidisks for use by OS or VSE.

VSE Minidisks

VSE space allocation is 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 cylinder of any minidisk initialized by the Device Support Facility is always reserved for use as an alternate track cylinder. Therefore, a VSE minidisk on a 2314 or 2319, must have at least two cylinders. For example, if you are specifying a ten-cylinder minidisk, the EXTENT card must refer to cylinders 0 through 8 only. This leaves the last cylinder for alternate track assignment. However, on a 3330, 3333, 3340, or 3350 minidisk, the Device Support Facility does not reserve a cylinder 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. A minidisk is initialized for use by running one of the following service programs in a virtual machine:

- For CMS disks other than CMS/VSAM, the CMS FORMAT command formats the specified tracks into 512-byte, 800-byte, 1024-byte, 2048-byte, or 4096-byte blocks or physical records.
- For CP disks, the stand-alone CP Format/Allocate program must be used to format specified tracks into 4096-byte blocks.
- For OS, VSE, and CMS/VSAM minidisks on count-key-data devices, the Device Support Facility 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). For count-key-data, any disk initialization program that supports the operating system's use of the DASD type may be used if you are initializing full disks.

Minidisks defined in the VM/SP directory are initialized only once. Temporary minidisks must be initialized each time they are used.

Alternate Tracks/Blocks

3330/3350 Disks

Alternate tracks assigned at the point of manufacture or by the Device Support Facility in the field are automatically handled on the 3330 or 3350 by the control unit. Minidisks on the 3330 Model 1 or 2 should be specified on cylinder 0 through cylinder 403 only. The remaining cylinders (404 to 411) are automatically used by the 3830 control unit for alternate tracks. Minidisks on the 3330 Model 11 can be specified on cylinder 0 through cylinder 807. Minidisks on the 3350 should be specified on cylinder 0 through cylinder 554 only. The remaining cylinders (555 to 559) are automatically used by the 3830 control unit for alternate tracks.

3340 Disks

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 (stand-alone) using a dedicated 3340 device. Cylinder 348 on the 3348 Data Module, Model 35 and cylinders 696 and 697 on the 3348 Data Module, Model 70 are reserved for this purpose. Once the Device Support Facility has assigned the alternate track, the disk, including the cylinder containing the defective track, may be used for any purpose whatever, including CP system residence, CMS minidisks, and so forth. There are two restrictions:

- A minidisk should not be located where its track 0, cylinder 0 falls on a defective track because it will be impossible for the CP IPL command to function for that minidisk.

- Any operating system doing SIO to this disk must be capable of doing normal alternate track error recovery.

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

Error Recovery Support

When an attempt to do I/O on a defective 3340 or 3344 track results in a track condition check, software error recovery procedures provide switching 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, a track condition check is reflected to the virtual machine so the operating system in the virtual machine will 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 VM/SP 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 seeking to the alternate track, any attempts at head switching to an unowned track in this cylinder are prevented.

3340 Cylinder Assignments

On 3340-35 devices, the primary data area is cylinder 0-347. Cylinder 348 is reserved for alternate tracks. On 3340-70 or 3344 devices, the primary data area is cylinder 0-695. Cylinders 696-697 are reserved for alternate tracks.

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, when changing from an old system (prior to Release 5 PLC 6) to a current system, it is necessary to revise space allocation and minidisk layouts on any 3340/3344 disk where "alternate track" cylinders had been used as a primary data area.

System Residence Devices: If the system residence device contains "alternate track" cylinders that have been used as the primary data area, files of existing control statements should be revised prior to generating a new system. In particular, the allocate function performed on the system residence disk and other CP-owned disks may have to be revised. Following this revision, the specification of the SYSRES, NAMESYS, and NAMENCP macros should be reviewed.

Minidisk Devices: If any minidisks on a 3340/3344 extend into the alternate track cylinders, they can be copied to another area of the disk or to another disk using the DASD dump restore (DDR) utility. In the past, when a 3340/3344 had a defective track, the cylinder with the bad track was unusable and minidisks would be allocated next to that cylinder, but not including it. In this case, all cylinders of the real disk should be dumped to tape using any version of the DDR utility.

If you use the new version of the DDR utility and the alternate track cylinders have been used as a primary data area, make sure that you specify the cylinder range exactly. For example, enter:

```
DUMP 0 TO 697
```

rather than specifying ALL. This no longer dumps anything from the final cylinders except tracks that have been assigned as alternates. Then you can run the Device Support Facility program to assign alternate tracks to the defective tracks so that all cylinders become usable. Then the new DDR utility can be used to restore minidisks from the tape, possibly reordering them into the previously unusable cylinders.

Note: Whenever a minidisk is moved to a new location or its size is changed, the corresponding MDISK statements in the system directory must be revised.

Only the most current versions of the DDR, DIR, and FMT utilities should be used with 3340/3344 devices after alternate tracks have been assigned.

Starter System Considerations: The starter system reserves cylinder 348 for alternate track use. Therefore, the 3340 starter system can be restored to a disk that has defective tracks (provided that alternate tracks have already been assigned by the Device Support Facility).

2314/2319 Disks

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 CMS/VSAM minidisks, however, do recognize and support alternate tracks on these types of DASD. The Device Support Facility program 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, you can:

- Run the stand-alone CP Format/Allocate service program if the minidisk is used by CP, and flag the whole cylinder containing the defective track as permanently assigned (PERM). This prevents CP from ever allocating that cylinder for CP paging, spooling, or temporary files. You must remember not to include this cylinder when you allocate disk space for any virtual machine's minidisk in the VM/SP directory.

- If the minidisk is used by either VSE, OS, or CMS/VSAM, reformat the minidisk (including the defective track) with the Device Support Facility program. 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, or with the VSE Initialize Disk utility program (INTDK) for DOS disks. 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. If a defective track is encountered, the hardware switches to an alternate track. When the end of the alternate track is reached, the hardware switches back to the first track following the defective track.

FB-512 Disks

Alternate blocks flagged at the point of manufacture are automatically handled on the FB-512 devices. Alternate blocks are assigned in the field by using the Format Defective Block procedure on the LOCATE CCW. The defective block number is provided and the hardware assigns 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 on real cylinder 0, track 0, record 3 for count-key-data devices or on block 1 for FB-512 devices. This label identifies the physical volume to VM/SP and must be in the form:

VOL1xxxxxx —or—

CMS=xxxxxx (for disks using an 800-byte format) —or—

CMS1=xxxxxx (for disks using a 512, 1K, 2K, or 4K format)

where xxxxxx is a 6-character volume label.

In addition, all virtual machine minidisks should have a label at virtual cylinder 0, track 0, record 3 for count-key-data devices or on block 1 for FB-512 devices. Labels created by the Device Support Facility, IEHDASDR, or INTDK are in the form

VOL1xxxxxx

where xxxxxx is a 6-character volume label.

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 7 on page 58, the volume indicated as OSDOS1 has its real cylinder 0 allocated to a minidisk that is formatted for use by OS. The volume serial number of that minidisk must be OSDOS1, the label that is associated with the real volume. Since the minidisk label identifies the physical volume, changing it affects the directory entries of all users who have minidisks on that volume.

You should not assign real cylinder or block 0 to a user as a data area, because if that user has read/write access to the disk, the label can be destroyed.

Additionally, you must not assign user minidisks to begin on real cylinder or block 0 of any physical volumes that are 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 written are compatible with OS, but indicate to OS that there is no space on that DASD. The initialization programs used to format OS, VSE, and CMS/VSAM minidisks write over and destroy this necessary control information if the space is assigned to a user minidisk. This causes CP system failures.

Sharing Minidisks

A minidisk can be shared by multiple virtual machines. One virtual machine is designated as the owner of the minidisk (it has an MDISK control statement in its VM/SP directory entry describing the minidisk). Other virtual machines can link to the minidisk either by a LINK control statement in their own VM/SP directory entry or by issuing a CP LINK command with the proper password during a terminal session.

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

```
MDISK 150 3380 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, or has the following LINK statement in its VM/SP directory entry, can read the 150 minidisk belonging to USERA:

```
LINK USERA 150 cuu RR
```

Where cuu is the virtual device address at which the 150 minidisk belonging to USERA is linked to another virtual machine. If you define another virtual machine, USERB, with the following statement in its VM/SP 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 VM/SP directory if both of the following conditions are met:

1. The minidisk being linked to has a password specified in the MDISK directory control statement corresponding to the type of link requested.

— AND —

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

Three primary types of sharing may exist for a minidisk and, correspondingly, three passwords may be specified on the MDISK statement (read-only, write, and multiple).

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

Chapter 6. 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, VSE 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. However, CMS 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 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 running 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 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. With the exception of the 3380, only disks supported by VSE/VSAM can be used for VSAM data sets in CMS. 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
- IBM 3375 Direct Access Storage
- IBM 3380 Direct Access Storage (OS VSAM enivornment of CMS only)

When the VM/SP processor is attached to an MSS, the CMS disk may be defined as a 3330 Model 1 that is mapped by VM/SP 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.

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. However, the format-5 DSCB created by OS 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 data sets are defined, and space is released whenever data spaces, catalogs, and 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 you have (1) OS PL/I programs that have files declared as ENV(INDEXED) and (2) if 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 Installation Guide*.

Chapter 7. Planning for CMS/DOS

Those of you who use CMS/DOS must, in certain cases, have available a VSE SYSRES. If you wish to use either the DOS/VS COBOL or PL/I 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 running VSAM and other VSE programs under CMS. Running VSAM under CMS is dependent on the generation of CMSDOS and CMSBAM.

For complete details on installing of the CMSDOS and CMSBAM discontiguous saved segments, see the *VM/SP 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 to 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 RPG II, 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 run 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 code from your system or private source 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 run 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 or are non-labeled 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 non-labeled and 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 and any existing labels are written over when the tapes are used for output.

CMS/DOS Disk Label Information Area

CMS/DOS does not support a disk 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.

However, in CMS/DOS, 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 run a program in an EXEC file and invoke that EXEC file by its filename.

Chapter 8. Planning for Virtual Machine Operating Systems (Other than CMS)

This section contains information about:

- The VM/VS Handshaking Feature
- Multiple Virtual Machines Using the Same Operating System
- VM/SP Using Channel Switching
- Alternate Path Support
- Operating Systems Using Reserve/Release

The VM/VS Handshaking Feature

The VM/VS Handshaking feature is a communication path between VM/SP 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:

- Closing VM/SP spool files when the system control program's output writer operation is complete
- Providing an optional nonpaging mode for operating systems running under the control of VM/SP
- Providing miscellaneous aids for an operating system's virtual machine running under the control of VM/SP

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

When the handshaking feature is active, the operating system using VM/VS handshaking closes the CP spool files by issuing the CP CLOSE command when a task or job has completed. Once these spool files are closed, they can be processed by VM/SP without operator interruption.

Operating systems using VM/VS handshaking can run 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 VM/SP 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 running under the control of VM/SP. With the handshaking feature, the guest system avoids some of the instructions and procedures that would be inefficient under VM/SP.

When the VM/VS Handshaking feature is active, the operation of a system control program closely resembles the stand-alone operation because much repetition of function between VM/SP and the operating system is eliminated.

Refer to *VM/SP Operating Systems in a Virtual Machine* for more details on handshaking.

Multiple Virtual Machines Using the Same Operating System

In general, an operating system that 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 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. CMS/DOS simulates VSE/AF supervisor and input/output functions, thus allowing the running 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 System Programmer's Guide* for more details.

VM/SP Using Channel Switching

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

- Two processors; one running VM/SP, the other running an operating system that supports channel switching.
- Two virtual machines running under VM/SP; each virtual machine operating system must support the channel switch feature (CMS does not).
- A single virtual machine running under VM/SP; the virtual machine operating system must support the channel switch feature.
- A processor running VM/SP and managing more than one path to devices through VM/SP alternate path support.

You can use the two- or four-channel switch for devices attached to two processors. For example, one processor could be running VM/SP and the other could be running OS, as shown in Figure 8.

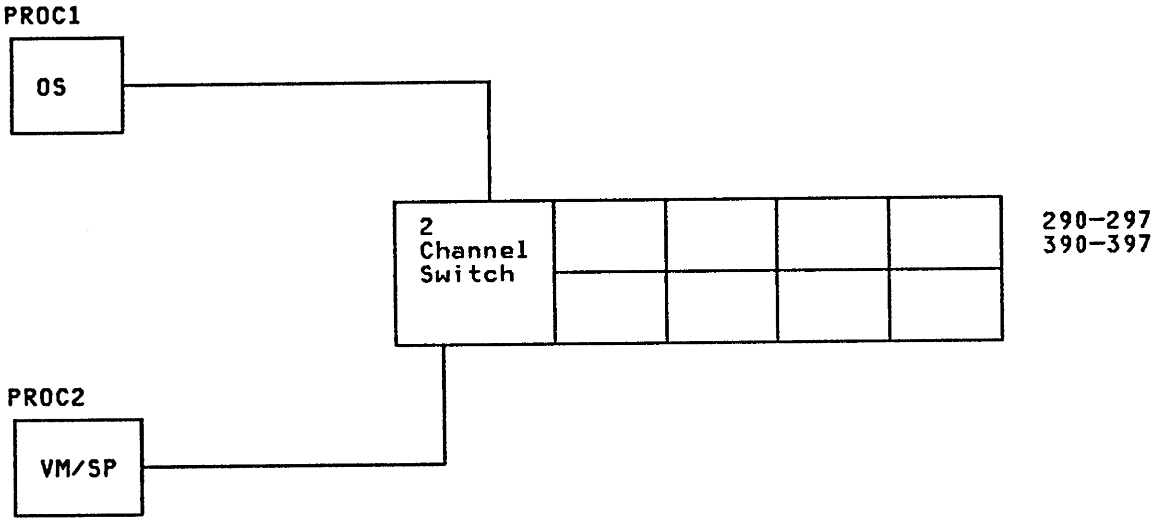


Figure 8. Channel Switching between Two Processors

VM/SP requires the following RDEVICE and RCTLUNIT macros 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 macros make it possible for you to run VM/SP on PROC1 or PROC2. If you are always going to run VM/SP on PROC2, you can eliminate one path (eliminate one set of RDEVICE and RCTLUNIT macros).

If any I/O devices controlled by VM/SP 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 configuration above 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. For example, one processor could be running VM/SP with OS running in a VM/SP virtual machine as shown in Figure 9. In this case, the virtual machine operating system supports channel switching.

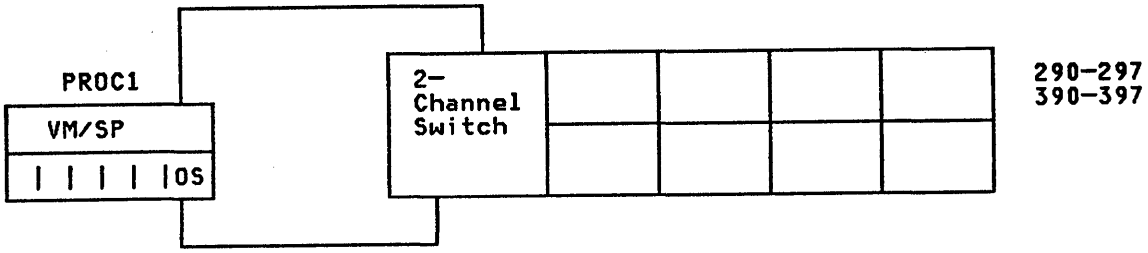


Figure 9. Channel Switching on One Processor

VM/SP requires the following RDEVICE and RCTLUNIT macros to support this configuration:

```
RDEVICE  ADDRESS=(290,8),DEVTYPE=2314
RDEVICE  ADDRESS=(390,8),DEVTYPE=2314
RCTLUNIT ADDRESS=290,CUTYPE=IFA
RCTLUNIT ADDRESS=390,CUTYPE=IFA
```

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

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

The 3880 Storage Control Unit contains two director modules. Each director module acts as a control unit providing input/output operations to a string of devices. Since each director module is separately addressable, one RCTLUNIT macro statement is required for each module. The optional ALTCH operand of the RCTLUNIT macro allows specification of up to three alternate channel interfaces to a single director module. The two- or four-channel switch feature allows up to four channels to have access to a director module. VM/SP supports a maximum of four channel paths to a single director module.

DASD devices 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 VM/SP directory. 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 VM/SP directory).

Alternate Path Support

Alternate path logic provides support for the Two-Channel Switch, and Two-Channel Switch Additional Feature, and the String Switch Feature by VM/SP. This support allows up to four channels on one control unit to be attached to VM/SP and/or one device to be attached to two logical control units. This provides the control program up to eight paths to one device when the maximum number of alternate channels and alternate control units are specified. When an I/O request is received for a device, VM/SP 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 more than one busy/scheduled path. 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 random choice of queuing based on number of IOBLOKs already queued, primary path, last busy scheduled path encountered, etc.

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

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

The goal of alternate path support is to define alternate paths to a device on the VM/SP processor. The virtual operating system does not define alternate paths. Instead, VM/SP defines alternate paths to the device with RCTLUNIT and RDEVICE macros. VM/SP then performs the alternate path I/O scheduling. If you wanted VM/SP, rather than the virtual operating system, to perform alternate path I/O scheduling, the following RDEVICE and RCTLUNIT macros would be required:

```
RDEVICE ADDRESS=(290,8),DEVTYPE=2314
RCTLUNIT ADDRESS=290,CUTYPE=IFA,ALTCH=(3)
RCHANNEL ADDRESS=3
RCHANNEL ADDRESS=2
```

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

```
RDEVICE ADDRESS=cuu,DEVTYPE=nnnn,MODEL=n,ALTCU=cuu
```

Example:

```
RDEVICE ADDRESS=(340,32),DEVTYPE=3330,MODEL=1,ALTCU=250
RCTLUNIT ADDRESS=340,CUTYPE=3830,FEATURE=32-DEVICE
RCTLUNIT ADDRESS=250,CUTYPE=3830,FEATURE=32-DEVICE
```

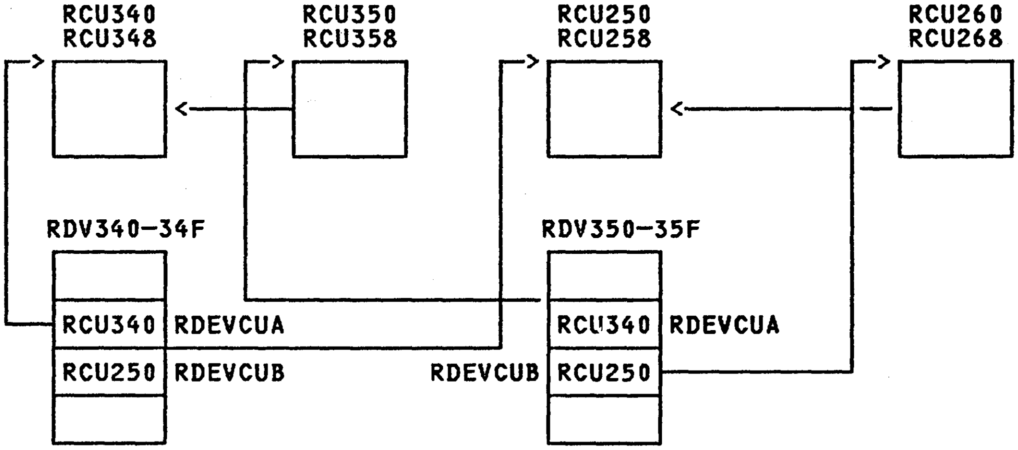


Figure 10. Real I/O Control Block Structure for Alternate Control Unit Specification

Figure 10 shows how the real I/O control block structure is coded and logically appears when an alternate control unit is specified.

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

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

Example:

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

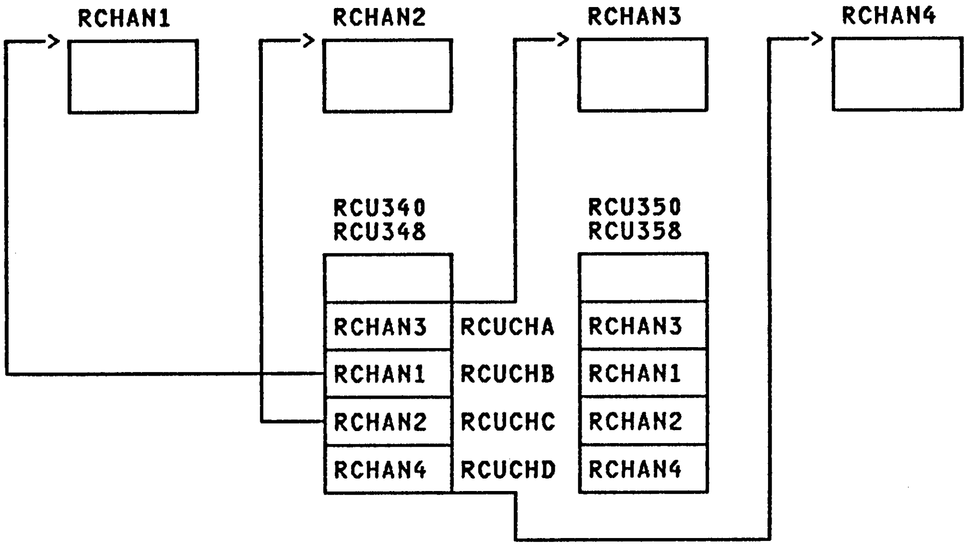


Figure 11. Real I/O Control Block Structure for Alternate Channel Specification

Figure 11 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 in line with the CP block structure.

Restrictions

The following restrictions apply directly to Alternate Path processing:

- VM/SP does not support alternate paths for devices that issue attention interrupts to cause a read response from the host; for example, the 3851 Mass Storage Control (MSC) unit.
- All devices on one physical control unit must be defined as either alternate path or no alternate path. There can be no splitting of control units when dealing with alternate paths.
- Only one alternate channel can be specified for MP systems.

Channel-Set Switching Facility

The channel-set switching facility is available on the 3033 attached processor and multiprocessor systems and the 3081 processor complex. This feature permits a set of channels to be switched from one processor to another in a multiprocessor or attached processor system. A channel-set is the collection of channels that are switched as a group. On a 3033 attached processor system, all online channels make up the channel-set.

VM/SP, 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 3081 attached processor systems, 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 the attached processor is equipped with the channel-set switching facility. The channel-set switching facility 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 issuing:

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

Monitoring and Service Support Facility

The 3081 Processor Complex uses the 3082 Processor Controller, a service processor that handles central communications for the processor complex. The 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 processor controller. MSSF provides I/O configuration and storage information for the 3081 processor complex. Virtual machine operating systems that are able to communicate with the MSSF can use the MSSF command word SCPINFO to obtain information about the processor's configuration and storage allocation.

If an MVS virtual machine operating in V=V mode issues the MSSF command word SCPINFO, VM/SP simulates the MSSF response by returning preformatted data to the virtual machine. If the MVS virtual machine operating in V=R mode issues the MSSF command word SCPINFO, VM/SP returns the unaltered information to the virtual machine.

MSSF also processes VARY PROCESSOR commands. When you issue the CP command VARY PROC ONLINE or VARY PROC OFFLINE VPHY on a 3081 processor, VM/SP generates an MSSFCALL instruction to the MSSF. MSSF then brings the processor online or takes the processor offline. When the request finishes, MSSF returns a completion status code.

VM/SP uses the MSSFCALL diagnose instruction and MSSF command words to communicate with the MSSF. The MSSFCALL diagnose (function code X'80') command words and completion status codes are described in the *VM/SP System Programmer's Guide*. The *VM/SP System Logic and Problem Determination Guide Vol.1 - CP* describes the response codes that VM/SP simulates for a V=V virtual machine.

MSSF does not require system generation macro definitions; however, you *must* 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 3081 processor complex. You supply this information to the controller by running the Input/Output Configuration Program.

The Input/Output Configuration Program (IOCP) processes IOCP macro definitions that contain I/O configuration information for the 3081 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. 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

Note: The device addresses (channel, control unit and device) you define in IOCDS should match the addresses you define in the real I/O configuration file (DMKRIO).

There are three versions of IOCP:

Stand-Alone Version Input/Output Configuration Program

VM/SP Input/Output Configuration Program

MVS/SP Input/Output Configuration Program

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

Stand-Alone Version Input/Output Configuration Program

If you are installing a 3081 processor for the first time, it may be necessary to run the stand-alone version 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 remove I/O configuration data from the IOCDS
- Obtain I/O configuration reports
- Write a new or updated IOCDS

The stand-alone version IOCP is shipped with the 3081 processor. For your convenience, a starter input/output configuration data set (IOCDS) is also shipped in level 0 of the processor controller. You should examine the starter IOCDS to determine whether a sufficient configuration of I/O devices, control units, and channels are defined and whether they match your system configuration. A listing of the starter IOCDS appears in the *OS/VS2 MVS, VM/SP and Stand-Alone Versions Input/Output Configuration Program User's Guide and Reference.*

If there are sufficient entries in the starter IOCDS for you to generate VM/SP, it is not necessary to run the stand-alone version IOCP. Simply power-on reset the processor controller and generate your VM/SP system. The processor controller uses the starter IOCDS defined in the level 0 IOCDS of the processor controller. After VM/SP 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, run the stand-alone version of IOCP.

VM/SP Input/Output Configuration Program

VM/SP IOCP runs 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 system (that is, you have defined a sufficient configuration to the processor controller in order to generate VM/SP) you can use the IOCP command to define your entire system configuration.

The IOCP command writes the new input/output configuration source file to the level 1 IOCDS in the processor controller. To test the new IOCDS:

1. Shutdown the VM/SP system
2. Power-on reset the processor controller using the level 1 IOCDS
3. Start VM/SP

MVS/SP 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 run 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.

You can run the MVS version of IOCP in a virtual machine operating under VM/SP; however, you cannot run IOCP in an MVS virtual machine if your system is operating in single processor mode.

References

For more information about IOCP source file and matching real I/O configuration file, refer to "Chapter 19. Preparing the Real I/O Configuration File (DMKRIO)."

For details concerning IOCP macro definitions, the CMS IOCP command, and the starter IOCDS, refer to the *OS/VS2 MVS, VM/SP and Stand-Alone Versions Input/Output Configuration Program User's Guide and Reference.*

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 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 attempts to correct the condition. 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 among devices. Certain devices are more critical than others. In order to allow you greater flexibility for monitoring I/O activity, CP allows you to specify a different time interval for each class of device. The time interval is set using the IBM-supplied defaults (provided in file DMKSYS) or as determined by you. You may change the default intervals by coding the SYSMIH macro statement and reassembling DMKSYS.

The SET MITIME command 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 Operator's Guide*. See "Chapter 21. Preparing the CP System Control File (DMKSYS)" for a description of the SYSMIH macro statement.

You can use the missing interrupt 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 (CLASSPE TYP3851) and 3330V DASD (CLASDASD FEATURE=VIRTUAL or FEATURE=SYSVIRT).

Operating Systems Using Reserve/Release

Shared DASD describes the capability of accessing direct access devices from two or more systems. The systems can be in virtual machines on the same or 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 they 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 end 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 action is reduced. 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 lessened. One update to a shared data set is needed, instead of the many 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 eased in a multi-system environment if the needed data is available to surviving systems without moving it.

- Direct access storage space may be saved because only one copy of the data is required instead of many 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 simulation packages. 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 option is the 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.

VM/SP reserve/release support can be addressed in two forms:

- Shared DASD
- Virtual Reserve/Release

Shared DASD refers to 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. 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 initialization routine issues a release CCW to tape and DASD volumes to determine if the two- or four-channel switch feature is installed.

Shared DASD

Operating systems that support shared DASD use reserve/release CCWs to preserve data integrity in the following cases:

- Two virtual machines running under VM/SP 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 VM/SP control program 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 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, in which case the virtual machine operating system uses reserve/release CCWs to maintain data integrity. Or it can be a non-VM/SP operating system with reserve/release capability.

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

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

  The device can be defined as a minidisk, on the VM/SP processor, which begins at real cylinder 0. Again the use of reserve/release and alternate path support are mutually exclusive. It should be noted that virtual reserve/release support should not be used 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.

  The performance of virtual machine operating systems may be degraded when sharing DASD. If not running single processor mode, I/O requests may be queued and the virtual machine left in I/O wait when a device is being used by another virtual machine or system. (A virtual machine is not left in I/O wait if a "SIO FAST" has been issued.) If in single processor mode, a device busy is reflected to the V=R guest when a device is in use and a device-end interrupt is reflected when an interruption occurs. Depending on conflict for the device, the V=R guest may get multiple device busy and device ends before the device is available to it.

Note: Defining a DASD as a minidisk gives many users the capability of linking to the volume. However, extended IOWAITS can occur when another processor has already issued a reserve to the pack because the busy condition will not be reflected to the guest operating system. The other possibility is for the disk to be attached or dedicated. This allows only one path to the device for each guest operating system, but the busy condition will be reflected to the guest operating system and prevent long IOWAITS.

Virtual Reserve/Release

Reserve/release software simulation in VM/SP 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 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. The VM/SP control program 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. However, no "device busy" or "device end" status 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, at which time the DIAGNOSE request will be reissued.

VM/SP Control Program Handling of a Reserve CCW

VM/SP reserve/release support and VM/SP alternate path support are mutually exclusive. The VM/SP CCW translation routine changes a reserve CCW to a sense CCW when alternate paths have been defined to the device from the VM/SP processor. Data integrity is not preserved when sharing a device between processors or virtual machines and alternate paths are defined. When using virtual reserve/release to share a minidisk between virtual machines on the VM/SP processor, VM/SP still changes a reserve CCW to a sense CCW when alternate paths are defined to the real device. However, since hardware reserve/release is simulated when virtual reserve/release is being used, data integrity is preserved when alternate paths are defined. Figure 12 on page 88 below identifies those cases when CP changes a reserve CCW to a sense CCW.

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)	CCW Comnd sent by VM/SP to Device	Note
Dedicated DASD or Tape	Not defined	Not applicable	Not applicable	Reserve	1
	Defined	Not applicable	Not applicable	Sense	2
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 system has been generated with alternate path support for those devices, it prevents the devices from being reserved. This action causes VM/SP to avoid a possible channel lockout. VM/SP 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, VM/SP 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, VM/SP 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 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 processor. This path allows VM/SP 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), VM/SP changes reserve/release CCW commands to SENSE CCW commands to prevent a possible channel lockout.

Figure 12. Summary of VM/SP Reserve/Release Support

Restrictions: Device Sharing Between Real Processors

- When a device is shared between processors and at least one of the processors is running VM/SP, the shared volume cannot contain more than one minidisk. The single minidisk may include 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 only pertains when using real reserve/release in addition to virtual reserve/release. Assume that two virtual machines are using 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 is more than one path from the VM/SP processor to the shared devices, as well as a path from the same devices to another processor, the paths from the VM/SP 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.*

Restrictions: 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.

Logical Device Support Facility

The Logical Device Support Facility allows an application running 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 (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 (7C)

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

The user virtual machine is signaled asynchronously by CP via the external interrupt code X'2402'. 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 interrupt and the associated logical device address. Control register 0, mask bit 22, must be on to receive this external interrupt.

Subfunctions that may be requested via DIAGNOSE (7C) are summarized in Figure 13. More complete descriptions of these subfunctions are in the *VM/SP System Programmer's Guide*.

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 13. Logical Device Support Facility Subfunctions

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 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 the pageable CP modules, DMKHPS and DMKHPT. If an application program does not call the Logical Device Support Facility, then modules DMKHPS and DMKHPT need not be assembled.

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:

- Communication paths and messages are begun by either CP or a virtual machine.
- A communicator can selectively start and stop 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 you 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 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 "Directory Program" in Chapter 18.

Although IUCV is similar to the Virtual Machine Communication Facility, IUCV does not replace VMCF. Both IUCV and VMCF are available and can be used. IUCV should be considered for new applications requiring inter-user communication. IUCV is used for communication between 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 system. The VMCF external interrupt 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 control the interrupts 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 waiting 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.

Chapter 9. Saved Systems and Discontiguous Segments

Saved systems are described in detail in the *VM/SP System Programmer's Guide*. If you plan to save core-image copies of virtual machine operating systems you should do the following when you generate VM/SP:

- Create an entry in the system name table for each system you wish to save.
- Reserve space on a CP-owned volume for each system you wish to save.

You create entries in the system name table by coding NAMESYS and NAMENCP macros and assembling the system name table (DMKSNT) file during system generation. You allocate DASD space as permanent (PERM) by running the Format/Allocate program. This program is run during system generation, but it is a stand-alone program that can be run at any time. You specify which volumes are to be owned by CP by coding the SYSOWN macro and assembling the CP system control (DMKSYS) file during system generation. These macros and files are described in Part 2.

If you decide to add entries to the system name table after you have installed VM/SP, you must code appropriate NAMESYS or NAMENCP macros, reassemble the system name table file (DMKSNT), and reload the CP nucleus. Likewise, if you must add a CP-owned volume after system generation, you must recode the SYSOWN macro, reassemble the CP system control file (DMKSYS), and reload the CP nucleus. Use the GENERATE EXEC procedure to reassemble DMKSNT and/or DMKSYS and to reload the CP nucleus. GENERATE is described in the *VM/SP Installation Guide.*

VM/SP supports discontiguous saved segments and provides shared segment protection.

With discontiguous saved segment support, you can attach and detach segments of storage to and from your virtual machine. These segments may contain reenterable code that can be shared by many users. Thus, programs that are required sometimes, but not all the time, can be saved and only loaded when they are needed. Also, discontiguous saved segments can be attached to your virtual machine in non-shared mode for testing and debugging.

When in attached processor (AP) or multiprocessor mode (MP), all protected shared segments are duplicated. Sufficient storage is obtained to construct duplicate page and swap tables in contiguous storage. This additional storage space should be planned for, when running in AP or MP mode.

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. However, 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 tables to be freed. See the *VM/SP System Programmer's Guide* for a description of shared segments.

Segments to be saved in this manner must be loaded at an address within your virtual machine and then saved. To do this in CMS (following CMS conventions) you must:

- Define your virtual machine size large enough to contain the discontiguous segments, loader tables, and CMS control block storage at the end of virtual storage.
- Load the segments.
- Save the segments.
- Reduce the virtual storage to its normal size.

When you attach these segments, they are attached beyond the end of your virtual machine. The procedures for loading and saving discontiguous segments are similar to the procedure that already exists for loading and saving systems.

The segment following your CMS nucleus cannot be used for a shared segment. This segment contains vital free storage pointers that would be overlaid with the text decks for the segment as a shared segment. An alternate CMS nucleus can be built at a different storage location, and this CMS nucleus can be used during the creation and saving of the shared segment that follows your normal CMS nucleus.

CMS has EXEC procedures to help you place portions of CMS in discontiguous saved segments:

- DOSGEN, which loads and saves CMS/DOS support
- SAMGEN, which loads and saves CMSBAM support
- VSAMGEN, which loads and saves CMS/VSAM and Access Method Services support (CMSAMS)

See the *VM/SP Installation Guide* for descriptions of how these EXEC procedures are used.

CP checks to see if a virtual machine has altered any shared segments before it dispatches the next virtual machine. When a shared segment is found to have been modified as a result of a CP STORE, ADSTOP, or TRACE command, CP issues a message to indicate that the shared copy has been replaced by a nonshared copy. Execution continues in the virtual machine with the nonshared copy. However, if a protected shared segment is found to be altered by any other means, and segment protection is on, CP sends a message to the current virtual machine to identify the altered page. The altered page is made unavailable and the virtual machine's execution is stopped by placing it into console function mode.

Saved systems must be named and may be shared. Discontiguous saved segment support is similar to saved system support. Therefore, you should understand saved systems before you read this section. See the *VM/SP System Programmer's Guide* for a description of saved systems.

A discontiguous saved segment is a segment that:

- Has a name associated with it.
- Was previously loaded and saved.
- May or may not be shared by multiple virtual machines.
- Can be loaded by a particular virtual machine in nonshared mode for testing and debugging.

A discontiguous saved segment can be logically attached to a virtual machine when it is needed and detached when it is not needed. The attaching and detaching is done by the name associated with the segment. The virtual machine attaching and detaching discontiguous saved segments must issue CP DIAGNOSE instructions to perform the proper linkage. Discontiguous saved segments are loaded at the same address at which they were saved. This address must be higher than the highest address of the virtual machine that is attaching it. A discontiguous saved segment cannot be attached by a virtual machine running in the virtual=real area. An example of discontiguous saved segments are the segments of CMS that support VSE program development and testing under CMS. These segments are reentrant and are named CMSDOS and CMSBAM. The starter system includes EXEC procedures (DOSGEN and SAMGEN), which help you load and save these segments. CMS contains all the necessary linkage to load the CMSDOS and CMSBAM segments when they are needed.

The main advantage of placing the CMS support for VSE in discontiguous saved segments is that it uses less real storage. Not all CMS users need the VSE support, and those who do need it probably do not need it all the time. CP keeps the segment tables in real nonpageable storage. These segment tables have an entry for each segment (whether it is saved or nonsaved) of virtual storage available to each active virtual machine. By placing the VSE support in discontiguous saved segments (called CMSDOS and CMSBAM), less real nonpageable storage is used. Your segment table has entries for the CMSDOS and CMSBAM segments (and all segments up to it) only when these segments are attached to your virtual machine.

To use discontiguous saved segments you must:

- Allocate permanent space on a CP-owned volume to contain the saved segment.

- Assign a name to the segment and specify where it is to be stored on disk. To do this, define an entry in the system name table (DMKSNT) with the NAMESYS macro. See "Coding the NAMESYS Macro" in Chapter 20, for the suggested layout of DMKSNT.

- Load and save the segment, using the appropriate EXEC procedure (DOSGEN, SAMGEN, or VSAMGEN).

- Be sure that the proper linkage for attaching and detaching discontiguous saved segments is in the operating system that needs the segment. CMS contains the linkage necessary to attach and detach the discontiguous saved segments it supports.

You can load and save a discontiguous saved segment any time after system generation.

Chapter 10. Performance Guidelines

The performance characteristics of an operating system when it is run in a virtual machine are difficult to predict. This unpredictability is a result of:

- The processor model.
- The system type (uniprocessor, attached processor, or multiprocessor).
- The total number of virtual machines running.
- The type of work being done by each virtual machine.
- The speed, capacity, and number of the paging devices.
- The use of fixed-head cylinders for preferred paging.
- The amount of real storage available.
- The degree of channel and control unit contention, as well as arm contention, affecting the paging device.
- The type and number of VM/SP performance options in use by one or more virtual machines.
- The existence of hardware assist.
- The favored priority and V=R options in effect.

Also, the virtual machine's channel mode, block multiplexer or selector, has an effect on the virtual machine's performance.

Note: The performance of an MSS being used by the operating system and shared with other systems depends on the total MSS usage and contention.

Performance Measurement and Analysis

The VM/SP control program has two commands that measure system performance to help you identify problem areas.

The MONITOR command controls the collection of performance data and the writing of it to system spool files or tapes. Both summary and trace data can be collected. You may specify classes of data to be 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. IBM Field Developed Program (FDP) VM/370: Performance/Monitor Analysis Program can be used to reduce the data collected. Guidelines for using this program provide you with information that will aid in determining the overall load and performance profile of your system. The VM/370 Performance/Monitor Analysis Program should enable you to analyze usage 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. Entering INDICATE displays system conditions existing at the time the command is issued. This includes attached processor (AP) and multiprocessor (MP) usage measurement when operating an AP or MP system. 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 21. Preparing the CP System Control File (DMKSYS)." See the *VM/SP 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, is another program designed specifically to help you monitor and tune your system.

Using Performance Options

Performance of a specific virtual machine can be improved by assigning it one or more performance options. These include:

- favored execution
- priority
- reserved page frames
- locked pages
- virtual=real
- queue drop elimination

Performance of a VM/SP system running virtual storage operating systems can be improved if you use virtual machine assist or Extended Control-Program Support. The manner in which these and MVS/System Extensions Support are supported by various VM/SP processors is detailed on the following page.

Virtual Machine Assist			MVS/System Extensions and MVS/SP Support		Extended Control-Program Support[3]	
Standard	RPQ	Not Available	System/370 Extended Facility	System/370 Extended Feature	Standard[2]	Not Available
135 135-3 138 145 145-3 148 158[1] 158-3[1] 158AP[1] 158MP[1] 3031UP 3031AP 4321 4331[4] 4331-2[4] 4341 3081-D16	168 168-3 168AP 168MP 3032 3033UP 3033AP 3033MP	155 155 II 165 165-3	3031UP 3031AP 3032 3033UP 3033AP 3033MP 3081 ECPS: MVS[5] 4341	158[1] 158-3[1] 158AP[1] 158MP[1] 168 168-3 168AP 168MP	135-3 138 145-3 148 3031UP 3031AP 4321 4331[4] 4331-2[4] 4341	135 145 155 155 II 158 158-3 158AP 158MP 165 165-3 168 168-3 168MP 3032 3033UP 3033AP 3033MP 3081

[1] Virtual machine assist and the System/370 Extended Feature are mutually exclusive on a Model 158 processor except for Model 3 with RPQ #MK3272 installed. However, in a Model 158 attached processor complex, virtual machine assist can be installed on one processor while the System/370 Extended Feature is installed on the other, or both may be installed on the attached processor (not the I/O processor).

[2] Users running VM/SP on a 135-3, 138, 145-3, or 148 with ECPS:VM/370 may not realize the full benefit of ECPS:VM/370 because shadow table maintenance algorithms may be used in preference to some ECPS:VM/370 algorithms.

[3] Compatibility must be established when using the functions contained in VM/SP on systems with ECPS:VM/370. To establish compatibility, make sure that the service level is compatible with the latest functional update to the hardware. If compatibility is not established, an error message is issued and ECPS:VM/370 is nullified.

[4] No charge special feature if ordered with the processor.

[5] ECPS:MVS and ECPS:VM/370 are mutually exclusive on the 4341 processors. On the 4341-2 and 4341-12 processors, ECPS:MVS and ECPS:VM/370 may be used concurrently if the ECPS Expansion Feature is installed.

Additional planning is needed to support the virtual=real option and virtual machine assist, as well as ECPS:VM/370. All of these performance options are described in detail in the *VM/SP System Programmer's Guide.*

Specifying a Virtual=Real Machine

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. Virtual page 0 does not exist as real page 0; it is relocated to the first real page after 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 a CP command 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 they do not use this area for input/output. However, violation of this restriction may cause damage to the entire VM/SP 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 run in the virtual=real area.

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 VM/SP 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 VM/SP. The virtual=real area must be released in total, that is, unused pages of the area cannot be selected for release.

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 VM/SP 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 VM/SP initialization is based on real machine size minus virtual=real size. Therefore, 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 counteracted 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 21. Preparing the CP System Control File (DMKSYS)." The examples used in the following discussions assume that you are allowing VM/SP to determine the number of free storage pages to allocate.

To use the virtual=real option effectively on a multipoint teleprocessing system with no CCW translation (SET NOTRANS ON), lines must be dedicated to that system via the ATTACH command or by VM/SP directory assignment. Conversely, on a multipoint 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 VM/SP directory for all virtual machines 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 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 18. Creating Your VM/SP Directory" describes the Directory program, including information about the VIRT=REAL operand of the OPTION control statement.

Note: With VM/SP extra DASD space is not required for a virtual=real system.

Real Storage Validation

VM/SP automatically validates real storage on the 3081 processor using the 3081 hardware instruction (TEST BLOCK). TEST BLOCK (TB) is an RRE format instruction (four bytes long) that has an operation code of X'B22C'. When VM/SP is loaded or initialized on a 3081 processor, VM/SP issues TB instructions to validate 4K blocks of real storage. This ensures that all page frames to be occupied by system modules are valid.

3081 real storage is not necessarily contiguous. One or more storage frames is 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 uses TB on the 3081 processor to detect non-contiguous blocks of real storage and recognize unusable storage frames.

At VM/SP 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 VM/SP 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 stopped 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 VM/SP load. Instead, informational messages are sent to the system operator and the load continues. For this reason, virtual machine operating systems that run 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 automatically validates the area using TEST BLOCK.

When VM/SP is initialized on a 3081 processor, those modules involved in the IPL process issue TB instructions to determine the status of every frame of real storage. If a non-addressable frame or a frame containing errors is detected within the area reserved for the nucleus (excluding the V=R area), system initialization is stopped with a disabled wait state code X'14'. Storage frames reserved for the V=R area are not validated at VM/SP initialization. V=R frames are validated only at VM/SP load time as described earlier. In both cases, VM/SP load and VM/SP initialization, non-addressable or invalid frames encountered outside the nucleus area are identified to the system operator by a series of informational messages.

VM/SP simulates TB for any virtual machine with EC mode capability regardless of real processor type. However, VM/SP and MVS/SP are the only operating systems that may issue TB. When a virtual machine is running V=V on a 3081, 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 and returning a condition code zero. For a V=R virtual machine running on a real 3081, CP performs a real TEST BLOCK and reflects the result to the virtual machine. If the storage block is usable, the storage block and storage key are set to zero 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. A protection exception is reflected to a virtual machine that attempts to issue a TB instruction to a shared page.

Virtual Storage Requirements

When generating VM/SP you have three limitations 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 Amount of Virtual=Real Space

If you are generating a VM/SP system to include a virtual=real machine, during 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.

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 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 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:

$$ARS = RM - \left[I + T + 12K + 4K \left[\frac{RM - 256K}{64K} \right] \right]$$

where:

RM is the real machine storage size.

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 for the CP internal trace table:

$$4K \left[\frac{RM}{256K} \right]$$

If the calculation RM/256K results in a fraction, the result should be rounded upward to the next higher integer.

12K is the fixed free storage allocated for the first 256K of real storage.

$$4K \left[\frac{RM - 256K}{64K} \right]$$

is the fixed free storage allocated for real storage beyond the first 256K (if there is no virtual=real area). 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 this time, 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:

$$VRS = RM - \left[I + T + 12K + 4K \left[\frac{RM - 256K - ARS}{64K} \right] + 16K \right]$$

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

$$\frac{RM - 256K - ARS}{64K}$$

results in a negative value, replace it with zero. If the same calculation results in a fractional number, disregard the fraction (see Examples 1 and 2). 16K is the storage needed at IPL time for the dynamic paging area. After VM/SP 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
512K	80K
768K	332K
1M	580K
2M	1582K

The values in this table assume the value of I is equivalent to 388K[9].

---

[9] Since the amount of storage required to IPL VM/SP varies with the inclusion of optional features and the number of devices in DMKRIO, this figure is used in the following examples for illustrative purposes only.

Example 1

Determine the maximum size of the virtual=real area for a real machine with 768K of storage running in a VM/SP system that requires 388K to IPL.

Formula 1

$$ARS = 768K - \left[388K + 4K\left[\frac{768K}{256K}\right] + 12K + 4K\left[\frac{768K - 256K}{64K}\right]\right]$$

$$ARS = 768K - [388K + 12K + 12K + 32K]$$

$$ARS = 768K - 444K$$

$$ARS = 324K$$

Formula 2

$$VRS = 768K - \left[388K + 12K + 12K + 4K\left[\frac{768K - 256K - 324K}{64K}\right] + 16K\right]$$

$$VRS = 768K - \left[412K + 4K\left[\frac{188K}{64K}\right] + 16K\right]$$

$$VRS = 768K - [412K + 4K[2] + 16K]$$

$$VRS = 332K$$

Note that the fraction (188/64) resulting from the

$$\frac{RM - 256K - ARS}{64K}$$

calculation in Formula 2 is rounded to the next lower integer, two.

Example 2

Determine the maximum size virtual=real area for a real machine with 2048K of real storage. The VM/SP system requires 388K to IPL.

Formula 1

$$\text{ARS} = 2048\text{K} - \left[388\text{K} + 4\text{K}\left[\frac{2048\text{K}}{256\text{K}}\right] + 12\text{K} + 4\text{K}\left[\frac{2048\text{K} - 256\text{K}}{64\text{K}}\right]\right]$$

$$\text{ARS} = 2048\text{K} - [388\text{K} + 4\text{K}[8] + 12\text{K} + 4\text{K}[28]]$$

$$\text{ARS} = 2048\text{K} - [388\text{K} + 32\text{K} + 12\text{K} + 112\text{K}]$$

$$\text{ARS} = 2048\text{K} - [544\text{K}]$$

$$\text{ARS} = 1504\text{K}$$

Formula 2

$$\text{VRS} = 2048\text{K} - \left[388\text{K} + 32\text{K} + 12\text{K} + 4\text{K}\left[\frac{2048\text{K} - 256\text{K} - 1504\text{K}}{64\text{K}}\right] + 16\text{K}\right]$$

$$\text{VRS} = 2048\text{K} - \left[432\text{K} + 4\text{K}\left[\frac{288}{64}\right] + 16\text{K}\right]$$

$$\text{VRS} = 2048\text{K} - [464\text{K}]$$

$$\text{VRS} = 1584\text{K}$$

Note that the fraction (288/64) resulting from the

$$\frac{\text{RM} - 256\text{K} - \text{ARS}}{64\text{K}}$$

calculation in Formula 2 is rounded to the lower integer, four.

Example 3

Determine the maximum size virtual=real area for a real machine with 400K of real storage. The VM/SP system requires 388K to IPL.

Formula 1

$$\mathrm{ARS} = 400K - \left[388K + 4K \left[\frac{400K}{256K} \right] + 12K + 4K \left[\frac{400K-256K}{64K} \right] \right]$$

$$\mathrm{ARS} = 400K - [388K + 4K[2] + 12K + 4K[2]]$$

$$\mathrm{ARS} = 400K - [416K]$$

$$\mathrm{ARS} = -16K$$

Since ARS is a negative number, CP cannot IPL and the following error message informs you of this condition:

```
DMKCPI955W INSUFFICIENT STORAGE FOR VM/SP
```

Calculating DASD Space

To evaluate 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 VM/SP programming. It improves the performance of VM/SP. Virtual storage operating systems that run in problem state under control of VM/SP use many privileged instructions and SVCs that cause interrupts that VM/SP must handle. When virtual machine assist is used, many of these interrupts are intercepted and handled by the processor. Consequently, VM/SP performance is improved. The manner in which virtual machine assist and Extended Control-Program Support (ECPS:VM/370) are supported by the various VM/SP processors is detailed under "Using Performance Options" earlier in this chapter.

Certain interrupts must be handled by VM/SP. 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 interrupt, VM/SP must handle SVC interrupts while address stops are set. Whenever you issue the ADSTOP command, VM/SP 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 system. Also, if the function is available to VM/SP, each virtual machine operator can turn the function off, and on again, for their own virtual machine. When you create your VM/SP directory, 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.

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 VM/SP'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 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 extensive 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

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 an accurate and repeatable interval timer value for virtual machines.

Both virtual machine assist and expanded virtual machine assist are automatically turned off if you start certain TRACE functions. In addition, virtual interval timer assist is turned off if external interrupts are traced. When the tracing function is stopped, CP automatically restarts these hardware assist functions.

For more details on VM/370 Extended Control-Program Support, refer to the *VM/SP System Programmer's Guide*.

Queue Drop Elimination

VM/SP attempts to optimize system throughput by monitoring the execution status of virtual machines. When a virtual machine becomes idle, VM/SP drops it from the active queue, invalidating the virtual machine's resident page and segment tables. In certain special cases, a virtual machine is determined to be idle and is dropped from the queue. However, the virtual machine becomes active again sooner than expected. If this cycle of queue dropping and reactivation is run repeatedly, the overhead involved in invalidating and revalidating the virtual machine's pages may result in system degradation.

CP SET QDROP allows you to control this situation. See the *VM/SP System Programmer's Guide* and the *VM/SP Operator's Guide* for details.

MVS/System Product and MVS/System Extensions Support

VM/SP enables an MVS system, running in a virtual machine, to use the MVS/System Product or MVS/System Extensions program product. "Using Performance Options," earlier in this chapter, details VM/SP processors that have this support available.

The following conditions are necessary to use the MVS/System Product or MVS/System Extensions support on your virtual machine:

- Hardware is available on the real machine
- The operator has entered a SET S370E ON for your VM/SP system
- 370E appears on the directory OPTION statement for your virtual machine, or you have entered the SET 370E ON command for your virtual machine

Note: The SET S370E command is invalid when running a VM under VM system. Therefore, if you are attempting to run MVS on a VM under VM system, do not set S370E on.

When this support is enabled, an operating system running in your virtual machine can use these functions of 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

Chapter 11. Attached Processor and Multiprocessor Systems

To generate an attached processor system it is necessary to reply "AP" when the GENERATE or service EXEC (5664167) asks:

```
ARE YOU GENERATING AN "AP" or "MP" SYSTEM?--RESPOND (NO|AP|MP)
```

A response of "AP" causes DMKSPA CNTRL and APLOAD (or AVLOAD for a system with a virtual=real area) to be used in place of DMKSP CNTRL and CPLOAD (or VRLOAD) EXECs.

DMKSPA MACLIB

The OPTIONS COPY member of DMKSPA MACLIB is identical to OPTIONS COPY in DMKSP MACLIB, except that the variable "&AP" is set to 1, causing AP support to be included in the module you are assembling. DMKSPA CNTRL uses this MACLIB to create a TXTAP rather than the usual TEXT, if the module is affected by attached processor support.

Modules Providing 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 loadlists (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. Combining the two lists should produce a complete list of all TXTAP decks (all AP versioned modules).

To generate a multiprocessor system it is necessary to reply "MP" when the GENERATE or service EXEC (5664167) asks:

```
ARE YOU GENERATING AN "AP" or "MP" SYSTEM?--RESPOND (NO|AP|MP)
```

A response of "MP" causes DMKSPM CNTRL and APLOAD (or AVLOAD for a system with a virtual=real area) to be used in place of DMKSP CNTRL and CPLOAD (or VRLOAD) EXECs.

DMKSPM MACLIB

The OPTIONS COPY member of DMKSPM MACLIB is identical to OPTIONS COPY in DMKMAC MACLIB except that the variable "&MP" is set to 1, causing MP support to be included in the module you are assembling. DMKSPM CNTRL uses this MACLIB to create a TXTMP rather than the usual TEXT, if the module is affected by multiprocessor support.

Modules Providing MP Support

DMKIOS is the one module used in MP support that is different from AP. DMKSPM MACLIB contains OPTIONS COPY with the "&MP" variable set on. The DMKRIO sample supplied with the VM/SP 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 DMKSPM control file, which will pick up the DMKSPM MACLIB. To assemble, use the command:

```
VMFASM DMKRIO DMKSPM
```

Chapter 12. Planning for 3270s

VM/SP attachments can be either local or remote. Local attachments do not require telecommunication lines. However, many devices supported for local attachment are also supported for remote attachment. 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).

Planning considerations for VTAM supported terminals attached to a VTAM service machine can be found in "Chapter 14. Planning for SNA Console Communications Services."

Remote 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 start and stop the teleprocessing lines, display stations and printers.

- CMS Editor and System Product Editor (XEDIT) support.

- The recording of MDR (Miscellaneous Data Recorder) records and OBR (Outboard Recording) records on the VM/SP 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 requested, you receive a NOT ACCEPTED message on the screen.

The following restrictions apply to VM/SP's support of remote 3270s:

- Remote 3270 terminals cannot be used as primary or alternate VM/SP system consoles.

- The number of binary synchronous lines supported by VM/SP 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 slow teleprocessing transmission speeds, 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. However, for remote 3270 users, and local 3270 users whose terminals may be distant from the system printer, VM/SP provides a limited hard-copy function 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 on the 3270 terminal 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 Hardware Configurations Supported

VM/SP's support of remote 3270s 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, and cluster and stand-alone control units supporting remote 3270s, are described in "Chapter 2. Configurations."

System Generation Requirements for Remotely Attached Display Systems

When generating VM/SP 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 use as virtual machine consoles. After DMKRIO assembles successfully, you should make a list of resource identification codes of all the remote 3270 lines and terminals. Give the list to your operations group. 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 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 19. Preparing the Real I/O Configuration File (DMKRIO)."

The Resource Identification Codes

The resource identification code is a four-digit hexadecimal code. The low-order two digits of the resource identification code is the resource address. The high-order two digits is the line code. The resource address is generated by VM/SP. 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 low-order two digits of the resource identification code.

The line code is also generated by VM/SP. Refer to the assembly listing for DMKRIO to determine the line code (the high-order two 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 high-order two digits is the line code that is assigned in the order that 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. VM/SP supports a maximum of 256 binary synchronous lines for use by remote 3270s. Thus, the maximum value of the two high-order digits is F. Figure 14 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 14. 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 DMKRIO to define:

- A clustered 3271 control unit with eight ports.
- A stand-alone 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,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=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 assigned to display devices, and printers can be interchanged among resource addresses assigned 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. However, the RDEVICE macros must be coded in the same sequence as their corresponding CLUSTER macros. As in the example above, CLUST078 is coded, then CLUST07A, followed by CLUST079. The RDEVICE macros follow the same sequence; RDEVICE 078, RDEVICE 07A, RDEVICE 079.

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

Figure 15. Sample List of Resource Identification Codes for Operations

Note: The line code is determined by referring to Figure 14; it corresponds to this example.

Local Hardware Configurations Supported

Control units attached directly to the processor channels, and the display stations and printers that attach directly to them, are described in "Chapter 2. Configurations."

System Generation Requirements for Locally Supported Display Systems

System generation requirements for locally supported terminals and control units are no different than are 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 19. Preparing the Real I/O Configuration File (DMKRIO)" for further details.

Chapter 13. Planning for the 3704/3705 Control Program

The IBM 3704 and 3705 Communications Controllers

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, 3287, 3288 and 3289 printers (EP mode only).
- Remote 2780 terminals (EP mode only).
- Emulator Program (EP) Version 3.0.

VM/SP's support of the 3704/3705 does not include:

- Remote 3704/3705 Communications Controllers.

Related Publications

The *Introduction to the IBM 3704 and 3705 Communications Controllers,* Order No. GA27-3051, describes the general functions of the 3704 and 3705. It is an essential publication for generating a 3704/3705 control program under VM/SP.

Additional 3704/3705 publications are listed in the Preface.

3704 and 3705 Support Package

Before you can generate a 3704/3705 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 files required for generating and loading the 3704/3705 control program under VM/SP. The support package is:

- IBM 3704/3705 Emulation Support and System Support Package (EP/VS SCP) for OS/VS (order No. 5744-AN1). VM/SP supports this package in emulation mode only.

This package contains the following basic material:

- A Program Directory
- Related 3704/3705 publications

- A magnetic tape containing the macros and modules of the 3704/3705 control program and the OS/VS system support programs.

VM/SP Support of the 3704 and 3705

VM/SP supports all models of 3704 and 3705 Communications Controllers. Three terminals are supported on start-stop lines: 1050, 2741, and CPT-TWX 33/35. The 3767 terminal (operating as a 2741) is supported by lines in EP mode. The 3101 and 3232 display terminals are supported as CPT-TWX 33/35.

The minimum internal storage required by an EP control program is 16K.

When planning for the installation of IBM 3704 and 3705 Communications Controllers, be sure that you are familiar with device characteristics, have the appropriate publications and support package, and have a VM/SP system that supports the 3704/3705.

Emulation Program (EP) with VM/SP

The IBM 3704 and 3705 Communications Controllers are programmable units. The Emulation Program (EP) can be generated to run in 3704/3705 storage.

The Emulation Program permits existing teleprocessing systems, including VM/SP, which use the IBM 2701, 2702, or 2703 Transmission Control Units, the 2703 Compatible Communications Adapter of the 4331 processor, or the Integrated Communications Adapter (ICA) of the System/370 Models 135, 135-3, and 138 to run without change on the 3704/3705.

In this publication, the term "3704/3705 control program" refers to the EP control program.

The EP 3704/3705 control program under VM/SP:

- Emulates 2701, 2702, and 2703 operations.
- Attaches to a 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 generate the EP control program in a CMS virtual machine.
- Supports the CP NETWORK command that allows you to load or dump the 3704/3705 and provides for automatic dumping and reloading if a fatal error occurs.

Planning Considerations

The generation of a 3704 or 3705 Communications Controller control program that runs under VM/SP is normally done after VM/SP system generation is complete. However, when a 3704 or 3705 is to be generated, the following preparations must be made:

- An RDEVICE macro instruction for the 3704 or 3705 must be included in the real I/O configuration (DMKRIO) file.

- 3704/3705 control programs to be used by VM/SP 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 image to be saved must be defined by a NAMENCP macro instruction in the system name table (DMKSNT), and saved with the SAVENCP command.

- Enough space to contain the 3704/3705 control program image must be allocated on the CP-owned volume specified in the NAMENCP macro instruction.

Note: The alternate console for VM/SP must not be on a telecommunication line on a real 3704/3705, unless the 3704/3705 is loaded by another operating system (OS/VS1, OS/VS2, or DOS/VS) before VM/SP is loaded.

The *VM/SP 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.

Coding the RDEVICE Macro

The RDEVICE macro is described in "Chapter 19. Preparing the Real I/O Configuration File (DMKRIO)."

Creating an Entry in the System Name Table

You must create an entry in the system name table (DMKSNT) for each different 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. In this way, you need not regenerate the VM/SP system just to update the system name table. If you should have to regenerate the VM/SP system to add a new entry to the system name table, see the discussion about the GENERATE EXEC procedure in the *VM/SP Installation Guide.*

The NAMENCP macro is described in "Chapter 20. 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 enough 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.

Chapter 14. Planning for SNA Console Communication 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 VM/SP console capabilities to operators on SNA terminals. 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 Communications Network Application Program Product (VCNA)

IUCV and SNA CCS are part of VM/SP.

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 consists of:

- Virtual Telecommunication Access Method (VTAM). This can be either ACF/VTAM or ACF/VTAME.

- VTAM Communications Network Application Program Product (VCNA) and one of the following operating systems with External Interrupt Support (EIS):

  - VSE/Advanced Functions (most current level)
  - OS/VS1 with Basic Programming Extensions Program Product

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 your procedures.

One method is to initialize CP, but not enable lines until the VTAM service machine (VSM) has been initialized. A similar technique could be used when you wish to load the NCP prior to initializing the VSM. You can initialize CP without enabling lines. Load the NCP using your own 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 disabled. The lines can then be enabled for use by CP. If the 370x is reloaded under control of the VSM, EP lines must be enabled 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 important characteristics of the transaction. The transaction can be tracked throughout the system by use of the SNA CCS work element block, WEBLOK (passed to VCNA), the SNA CCS and VCNA path id's, and the IUCV message id. These fields can be matched with corresponding or similar fields in the IUCV trace elements in CP and VCNA trace elements in VTAM.

Normal Trace

SNA CCS creates trace table entries in the CP trace table, leaving an audit trail of its activities. This trace is started 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 independent 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. If the WEBLOK that is passed between SNA CCS and VCNA is invalid, data in the trace element pertains to the invalid WEBLOK.

The error trace function is started 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 save storage. You should delete the five SNA CCS modules (DMKVCP, DMKVCR, DMKVCT, DMKVCV, and DMKVCX) by excluding them from the loadlist. See "Chapter 3. Estimating VM/SP Storage Requirements" for more information on excluding SNA CCS support modules. 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 15. Planning for the 3800 Image Library

The generation of a 3800 image library that runs under VM/SP is normally done after the VM/SP system generation is completed. However, when a 3800 image library is to be generated, the following preparations must be made:

- An RDEVICE macro instruction for the 3800 printer must be included in the real I/O configuration (DMKRIO) file.

- The 3800 image libraries to be used by VM/SP must be stored 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.

- Enough space to contain the 3800 image library must be 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. The specification of these elements is done through IEBIMAGE control cards as described in the *VM/SP Operator's Guide*.

Creating and Updating a 3800 Named System

A named system must be established (via the NAME3800 macro) in DMKSNT for each system data set capable of image library activation. The named system is used to contain 3800 character arrangement tables, copy modifications, graphic modifications, and FCBs. They can then be referenced by name and data for them obtained from this named system when the file referencing them is about to print on a 3800. The active named system for a particular 3800 is in its RDEVBLOK and can be changed by the START command. See the NAME3800 macro description in "Chapter 20. Preparing the System Name Table File (DMKSNT)."

Programs exist to enable you to dynamically change:

- Character arrangement tables
- Graphic modifications
- Copy modifications
- FCBs available

With these programs (GENIMAGE, IMAGELIB, and IMAGEMOD), and the named system support discussed above, you can make these changes dynamically, without a VM/SP system load. GENIMAGE, IMAGELIB, and IMAGEMOD are described in detail in the *VM/SP Operator's Guide*.

Coding the RDEVICE Macro

The RDEVICE macro is described in "Chapter 19. Preparing the Real I/O Configuration File (DMKRIO)."

As a VM/SP real spooling device, the following hardware features of the 3800 are supported:

- 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. However, this feature is supported through use of a virtual 3800.

Related Publications

The *Concepts of the IBM 3800 Printing Subsystem* manual, GC20-1775, is a first reader for users of printers who wish to take a quick look at the non-impact IBM 3800 Printing Subsystem, its basic concepts, and 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, provides 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 listed for copy modification and control and graphic character modification.

The *IBM 3800 Printing Subsystem Programmer's Guide*, GC26-3846, provides planning information for the IBM 3800 Printing Subsystem and information on how to use the 3800.

Chapter 16. Planning for the 3850 Mass Storage System

Generating a VM/SP System that Supports a 3850

The 3850 Mass Storage System (MSS) supplies large amounts of online data under system control. Up to 472 billion bytes of data space is 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:

> System/370 Models 145, 145-3, 148, 155II, 158UP/AP/MP, 165II, 168UP/AP/MP, 3031UP/AP, 3032, 3033UP/AP/MP, 3033-N, 3033-S, 3042AP-2, 3081 and the 4300 processors.

Major hardware components of MSS are:

- The 3851 Mass Storage Facility (MSF)
- The 3830 Model 3 Storage Control for System/370 Models 145, 145-3, 148, 155II, 158, 165II, 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 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, or in a real processor, connected to the same MSC as VM/SP. 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 system, the addresses known to VM/SP 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 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 system residence, paging, and spooling.

3330V devices may be used in three ways in VM/SP:

- Mounted and used as VM/SP 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 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, and full MSC support is in OS/VS1 and MVS. Therefore, to mount and demount 3330V volumes for VM/SP 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. 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 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 system, but runs in problem program state in an OS/VS1 or MVS system. This DMKMSS program is the interface between CP and MSC support in OS/VS.

Obtaining the MSS Communicator Program

When there is a Mass Storage System (MSS) attached to your VM/SP system, you can use the DMKMSS program to communicate between the VM/SP control program and the MSC. This enables VM/SP to dynamically mount and demount MSS volumes. In this case, you should obtain the file that will install the DMKMSS program in a VS system. The required file is distributed with the VM/SP control program object code, which resides on userid MAINT's 194 minidisk. The first step is to ensure that MAINT has access to its 194. Logon the MAINT virtual machine and issue the CMS command:

```
access 194 d/a
```

For a cardless system, prior to punching any files, spool the virtual punch to yourself:

```
sp pu *
```

Next, punch the file; this will install DMKMSS in your VS system. If your VS system is OS/VS1, issue the command:

```
punch mssvs1 jcl
```

If your VS system is OS/VS2, issue the command:

```
punch mssvs2 jcl
```

The punched output you receive is a series of OS/VS jobs. This file must be saved. When you run the jobs in your OS/VS system, they will install the DMKMSS program and create a VS operator procedure called DMKMSS. This procedure is later used to start the program in the communicator virtual machine.

OS/VS1 Jobs

There are four OS/VS1 jobs. They are:

- LINKDMK - This job linkedits the object code for DMKMSS into the SYS1.LINKLIB data set; the load module name is DMKMSS. The DMKMSS program must be located in SYS1.LINKLIB; this is one of the requirements of APF (Authorized Program Facility).
- DUMPT - This job prints two lists (named IEFSD161 and IEF161SD) in the system program properties table. These lists are used in the next job.
- APFZAP - This job, as distributed with VM/SP, replaces the module IEHATLAS with DMKMSS in the program properties table; this adds DMKMSS as an authorized program and removes IEHATLAS. If you wish to retain IEHATLAS as an authorized program, examine the lists produced in job DUMPT above. Change the control statement provided in APFZAP to add DMKMSS rather than replace IEHATLAS.
- LINKPROC - This job adds the procedure DMKMSS to the SYS1.PROCLIB data set. You must place the communicator device address on the COMM control statement before running this job. After the job has completed, the OS/VS1 system operator may start the DMKMSS program by issuing the command 'START DMKMSS.P*' where * is the number of the partition in which DMKMSS is to run.

OS/VS2 Jobs

There are two OS/VS2 jobs. They are:

- LINKDMK - This job linkedits the object code for DMKMSS into the SYS1.LINKLIB data set; the load module name is DMKMSS. In OS/VS2, this linkedit provides the necessary APF authorization.
- LINKPROC - This job adds the procedure DMKMSS to the SYS1.PROCLIB data set. After this job completes, the OS/VS2 system operator may start the DMKMSS program by issuing the OS/VS2 operator command 'START DMKMSS'. Before you run job LINKPROC, you must place the communicator device address on the COMM control statement.

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 use VM/SP 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. The 3330V address is passed to MSC. The order tells MSC to demount any volumes currently mounted on that 3330V.

A 3330V may be offline to a virtual machine because none of VM/SP's 3330Vs were allocated to the virtual machine at that virtual address. However, 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 or another virtual machine MSC can be demounted.

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 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 issues an MSS mount or demount request by generating an attention interruption on a specified device. This device must be specified in the directory of the virtual machine 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 those who 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 VM/SP, then this section does not apply. However, you must follow the guidelines in this topic if you have a virtual machine that has 3330Vs dedicated to it (that is, you plan to run more than one MSS virtual machine or to run VS MSS jobs in the MSS communication virtual machine).

MSC is controlled by 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. 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 processor 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 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 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'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 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 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. Interruptions received from the virtual machine's 150 are received on VM/SP'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's 250 is the volume demounted.

Two different virtual machines, having the same virtual device addresses can run concurrently under VM/SP. 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 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 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 under 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 issued from either a native processor or a VS virtual machine. CONVERTV 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 using the 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 VM/SP can make the virtual IPL device appear to be a 3330-1. The following steps describe one way this can be done:

- Use the CREATEV command to create an MSS volume with a volume serial number of VOL001.

- Define a directory entry for a virtual machine (VS2VM) with an MDISK statement, describing a minidisk spanning cylinders 1 through 401 on volume VOL001.

- VM/SP mounts VOL001 and allocates the minidisk when VS2VM logs on. The operator can then attach a 3330-1 containing a VS2 system to VS2VM.

- Copy cylinders 0-400 of the 3330-1 to the minidisk within VS2VM.

- IPL the virtual device address corresponding to the minidisk as a VS2 system residence device.

The VM/SP RDEVICE Macro

The 3330V device addresses generated in CP can be used for two purposes. They can have 3330V system volumes containing minidisks mounted on them, or they can be dedicated to a virtual machine. In either case, the control program can dynamically select a specific device to satisfy a request. You must divide the pool of available 3330V devices into two types, one for system volumes and one for dedicated volumes. The FEATURE= operand of the RDEVICE macro is used to first indicate that a device address is a 3330V as opposed to a 3330-1, and second, to indicate the type of 3330V (system or dedicated).

When coding the RDEVICE macro for a 3330V device address, either FEATURE=VIRTUAL or FEATURE=SYSVIRT must be coded, where:

- VIRTUAL defines a 3330V that may not be used for system volumes. It may be dedicated or attached to virtual machines as a 3330-1 or 3330V.
- SYSVIRT defines a 3330V that is used for VM/SP system volumes. MSS volumes that are 3330V, can be mounted as SYSVIRT 3330V devices, but cannot be dedicated or attached to a virtual machine.

Part 2. Defining Your VM/SP System

Part 2 describes macros and control statements you need to define your VM/SP system. It contains:

- Introduction to System Definition
- Creating your VM/SP Directory
- Preparing the Real I/O Configuration File (DMKRIO)
- Preparing the System Name Table File (DMKSNT)
- Preparing the CP System Control File (DMKSYS)
- Additional System Definition Files

Chapter 17. Introduction to System Definition

Before starting system generation on a real machine, you must create three files that describe the VM/SP system you are generating. There are two optional additional files. You can use card input, or create these files using the System Product Editor. If you are modifying an existing VM/SP system, you can use the System Product Editor to alter existing files. You can also use other systems to create these files on tape in a card-to-card image, or use another VM/SP or VM/370 system to create a tape in tape dump format. In these cases you have to bring the files into the system generation process using the CMS commands TAPE LOAD, MOVEFILE, and TAPPDS.

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, and starter system defaults.
- VM/SP directory file (normally a CMS file named USER DIRECT), which contains the VM/SP directory entries that define the virtual machine configuration for each user.

  You can also change the forms control buffer load (DMKFCB).

In addition, you should prepare the system name table (DMKSNT) file if you plan to save systems. If you generate the 3704/3705 control program, you must save it. Otherwise, the 3704/3705 control program can not be loaded by VM/SP. "Chapter 9. Saved Systems and Discontiguous Segments" describes the requirements for saving systems.

System Integrity

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

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

The following terms apply to the system integrity definition:

main storage protection: refers to the isolation of one virtual machine from another, which the Control Program accomplishes by the use of hardware Dynamic Address Translation.

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

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

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

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

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

- V=R with the NOTRANS option
- V=R with the Shadow-Table-Bypass SET command option
- The Preferred Machine Assist
- The Single Processor Mode.

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

VM/SP Control Program system integrity does not specifically include the protection of data between multiple users of a single CMS batch system, nor does it apply to virtual machines using Non-Disruptive Transition (NDT) support.

Customer Responsibilities

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

Some areas for which effective controls should be implemented are:

- Password protection
- Assignment of appropriate privilege classes
- Assignment of directory options
- Set up and authorization of guest virtual machines.

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

Note: IBM will accept APARs that describe exposures to the system integrity of VM/SP or that describe problems where a program running in a virtual machine not authorized by a mechanism under the customer's control introduces an exposure to the system integrity of VM/SP. A customer who discovers a system integrity problem or exposure should report it to the Customer Support Center (FE Level 1).

Chapter 18. Creating Your VM/SP Directory

The VM/SP directory contains the entries of all potential virtual machines permitted to logon the VM/SP system. Without the proper directory entry, a user cannot logon to VM/SP. 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 VM/SP 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 VM/SP Directory program (module DMKDIR, started by the DIRECT command, or run stand-alone) processes your control statements and writes the directory on disk. You describe your real configuration when you create the real I/O configuration file (DMKRIO). You describe your many virtual configurations with Directory program control statements.

To create a VM/SP directory, you must:

- Prepare Directory program control statements
- Format and allocate DASD space to contain the directory
- Run the Directory program

During system generation, you must (1) format and allocate DASD space for the directory and (2) generate it.

Considerations for Preparing the Directory Control Statements

First, prepare a directory control statement that defines the device on which the VM/SP directory is to be written. This statement (DIRECTORY) must be the first control statement in the input to the Directory program. It is followed by the sets of statements describing your 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 VM/SP directory.

VM/SP does not check for overlapping extents. Therefore, you must ensure that minidisk extents defined in the VM/SP directory do not overlap each other and (in the case of 3330, 3340, and 3350 disks) do not overlap the "alternate track" cylinders. If overlap conditions exist, file data damage is inevitable. 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 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 functions

and virtual machines to update or operate:

- The CP system
- The CMS system
- The hardware
- Other operating systems that run in the virtual machine environment

System Support Virtual Machines

At system generation, two additional virtual machines should be created beyond those needed by VM/SP users. These machines include one each for hardware and software support. IBM FE programming support should be consulted when you establish 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 running problem programs.
- Input/output equipment, which can be supported using online test (OLT) under OLTSEP. The OLTSEP program can be run 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 VM/SP directory for the IBM service representative. The virtual machine should have enough virtual storage defined to run OLTSEP. Normally, the device being tested is dedicated to the service representative's 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 VM/SP recorded machine check and channel check errors. This directory entry is included in the VM/SP directory provided with the starter system.

Software Support

The virtual machine for software support should have sufficient system resources allocated to recreate problems (virtually) that occur on the real machine, and to perform software service activities. ECMODE ON must be specified for this machine.

Sample Directory Entries

The following sample VM/SP directory entries provide you with some of the virtual machines necessary for system operation and updating. The indentations are for readability and are not required by the directory program. LINK control statements are used whenever possible to reduce the number of changes to the VM/SP directory whenever a minidisk extent is moved. A brief explanation of some of the virtual machine userids follows.

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 publication *VM/SP OLSTEP 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 A
 LINK MAINT 190 190 RR
 LINK MAINT 19D 19D RR
 LINK MAINT 201 192 RR
 MDISK 191 3380 099 001 VMSRES WR READ WRITE
```

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, 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.

```
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
 LINK MAINT 19E 19E RR
 MDISK 191 3380 040 001 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 system was generated. All abnormal termination dumps are sent to this virtual machine. This user normally is given command privilege classes B, C, E, and G. If the directory entry contains all disks normally attached to the system, described as full-volume minidisks, you can rewrite the VM/SP 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 SYSPROG
 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 3380 212 001 VMSRES MR RIPCS WIPCS MIPCS
 MDISK 193 3380 213 008 VMSRES MR RIPCS WIPCS MIPCS
```

Other System Virtual Machines

In addition to virtual machines discussed to this point, there are virtual machines that:

- Support and update the VM/SP system
- Test new releases of the system before placing them into production
- Provide other users with a remote file spooling capability.

A Virtual Machine for Updating and Supporting VM/SP (MAINT)

The following directory entry defines a virtual machine (MAINT) that can support and update the VM/SP 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, CMSL, and other systems.

```
USER MAINT CPCMS 3M 16M ABCDEFG
 ACCOUNT 1 SYSPROG
 OPTION ECMODE REALTIMER
 IPL 190
 CONSOLE 009 3215
 SPOOL 00C 2540 READER *
 SPOOL 00D 2540 PUNCH  A
 SPOOL 00E 1403 A
 MDISK 123 3380 000 885 VMSRES MW RSYSRES WSYSRES MSYSRES
 MDISK 125 3380 000 885 VMPK01 MW RSYSRES WSYSRES MSYSRES
 MDISK 126 3380 000 885 VMSTGE MW RSYSRES WSYSRES MSYSRES
 MDISK 190 3380 052 045 VMSRES MW ALL     WMAINT  MMAINT
 MDISK 191 3380 011 008 VMSRES MW RMAINT  WMAINT  MMAINT
 MDISK 194 3380 019 021 VMSRES MW RMAINT  WMAINT  MMAINT
 MDISK 201 3380 832 021 VMSRES MW RMAINT  WMAINT  MMAINT
 MDISK 293 3380 853 011 VMSRES MW RCMSAUX WCMSAUX MCMSAUX
 MDISK 294 3380 864 011 VMSRES MW RCPAUX  WCPAUX  MCPAUX
 MDISK 393 3380 001 068 VMSTGE WR READ WRITE
 MDISK 394 3380 809 076 VMSTGE WR READ WRITE
 MDISK 19E 3380 127 045 VMSRES MW ALL     WMAINT  MMAINT
 MDISK 319 3380 100 027 VMSRES MW ALL     WMAINT  MMAINT
 MDISK 19D 3380 221 027 VMSRES MW ALL     WMAINT  MMAINT
```

Following is a summary of MAINT's minidisk usage:

Disk	**Usage**
123	Provides full pack read/write access to volume VMSRES.
125	Provides full pack read/write access to volume VMPK01.
126	Provides full pack read/write access to volume VMSTGE.
190	CMS system disk (S-disk). Contains the CMS nuclei (CMS and optionally, CMSL) and the CMS modules and EXEC procedures available to CMS users. Any virtual machine that wants to use CMS links to this disk.
191	Work area. This is MAINT's A-disk.
194	CP TEXT retention.
201	EREP TXTLIBs.
293	CMS PTFs and updates.
294	CP PTFs and updates.
393	CMS source files.
394	CP source files.
19E	Y-disk extension of the 190 S-disk.
319	Optional Program Products.
19D	HELP files.

The Directory Program

You can run the VM/SP directory program under CMS (using the DIRECT command) or stand-alone. The stand-alone version of the directory program 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 named filename 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 VM/SP directory in the VOL1 label. If the active system directory is the directory being updated, it is placed in use at this time. You can print the new directory by issuing the CMS command PRINT USER DIRECT (or PRINT filename DIRECT).

The virtual machine running 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 VM/SP system residence volume, your virtual machine's current directory entry must have write access to the volume containing the current VM/SP 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 C 2540 READER *
    SPOOL D 2540 PUNCH A
    SPOOL E 1403 A
    SPECIAL OFF TIMER
    LINK MAINT 190 190 RR
    LINK MAINT 19E 19E RR
    MDISK 191 3380 200 003 VMSRES MR RDIRM WDIRM MDIRM
    MDISK 193 3380 086 009 VMSEXT MR RDIRM WDIRM MDIRM
    MDISK 195 3380 203 009 VMSRES MR RDIRM WDIRM MDIRM
    MDISK 123 3380 000 885 VMSRES MW
```

Using XEDIT you can create or modify a card-image file of the VM/SP 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 formats 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, directory changes for a user currently logged on to the system do not 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).

Invoking the Directory Program (DMKDIR) Under CMS

The VM/SP Directory program records the configuration of each user's virtual machine in the VM/SP directory. Each virtual machine configuration includes counterparts of the components found in a real machine: 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 XEDIT to include additional directory entries.

The CMS DIRECT Command

Use the CMS DIRECT command to process a directory file to see if it follows the required format. To actually change or swap the currently active VM/SP directory, you must have 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 qualification 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 VM/SP 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 VM/SP directory each time it is created. If you run the directory program under VM/SP, the newly created directory is dynamically swapped, and placed in use by VM/SP (provided that the directory you update is currently in use by the system, or the target directory pack is present in the system owned list). The format of the DIRECT command is:

```
DIRECT   [            [            [            ]]]
         [ filename   [ filetype   [ filemode   ]]]   [(EDIT)]
         [ USER       [ DIRECT     [    *       ]]]
         [            [            [            ]]]
```

where:

`filename [filetype [filemode]]`
is the identification of the file containing control statements for the Directory program. If no filename and filetype are specified, the program defaults to a file named USER DIRECT; otherwise, it looks for the file named. If only filename is specified, filetype defaults to DIRECT. The filemode defaults to * if not specified.

`(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 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 abnormally ends, the old directory is not altered. Normal completion places the directory in use by VM/SP. After the new directory is created, it can be printed by issuing the CMS command PRINT USER DIRECT or PRINT filename DIRECT.

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 file 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.

Directory Control Statements

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 verify all control statements before ending. If the directory runs out of space, the program stops immediately. After an abnormal ending, 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  cuu  devtype  volser  [alt-cuu]
```

where:

`cuu`	is the address of the device to contain the directory and is specified in three hexadecimal digits.
`devtype`	represents a supported device type suitable for the VM/SP 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 VM/SP directory.
`volser`	is the volume serial number of the directory volume (one to six-alphameric characters).
`alt-cuu`	identifies an alternate device address for writing the directory if the primary cuu is not available. For multiprocessing systems, an alt-cuu specification allows native execution of the directory program on either processor.

USER Control Statement

The USER control statement defines a virtual machine and creates a VM/SP directory entry. It identifies the directory entry for one user. A separate USER statement must be prepared for each directory 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 VM/SP system VMBLOK, and should never be used for a virtual machine.

If you plan to use any of the following CMS commands, the userid must contain those characters 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).

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.

Notes:

1. The userid should not contain the characters "LOGONxxx," where xxx is one of your terminal addresses. During logon this character string is assigned to the terminal at address xxx 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.
3. Do not specify ALL as a userid. It is reserved for VM/SP.
4. If the userid of AUTOLOG1 (a reserved system user identification) is used, during the VM/SP 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 system and the virtual machine you are defining in these control statements. **Note:** Use the reserved password NOLOG for users who do not have a virtual machine configuration in the VM/SP directory. The NOLOG user uses the real card reader spool device as a means of entry for processing by the CMS batch facility. NOLOG is used for spooling purposes only. Attempts to logon using this password are not allowed. Specifying NOLOG provides additional security for your VM/SP installation.
`stor`	is one to eight-decimal digits that define the virtual machine's storage size. It must be a multiple of 4K. The last character must be K or M. The default is 128K. The minimum size is 8K. All entries not on a 4K boundary are rounded up to the next 4K boundary. The maximum size is 16M.
`mstor`	is one to eight-decimal digits that define the maximum virtual machine storage size that this user can define after logging on the system. It must be coded in multiples of 4K. The last digit must be K or M. The default size is 1M. All entries not on a 4K boundary are rounded to the next 4K boundary. The minimum size is 8K. The maximum size that can be specified is 16M.
`cl`	is one to eight-alphabetic characters from A to H (with no intervening blanks) defining the privilege class(es) of this user. The default is G. **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 desired.
`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 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 started 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.

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. If this statement is omitted, both the account number and the distribution code default to the userid. This statement (if coded) must follow the USER statement and precede the first device statement. 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.

OPTION Control Statement

The OPTION control statement selects specific options. This statement is optional and, if used, must follow the USER statement or another OPTION statement. It 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]  [Acct]
          [Svcoff]  [BMX]  [CPUID bbbbbb]  [AFFinity nn]
          [VMsave]  [STFirst]  [370E]  [Maxconn nnnnn]  [MIH]
```

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:

1. Operate in System/370 extended control mode (such as VM/SP itself).
2. Use the dynamic address translation facility (such as OS/VS1, OS/VS2, DOS/VS, DOS/VSE, VSE/AF, VM/370, and VM/SP).
3. Use control registers other than zero (such as OS GTF (General Trace Facility), which uses Monitor Call and requires control register eight).
4. Depend on the System/370 extended channel masking feature.

The ECMODE option must also be specified for the virtual machine that is to perform system support or updating. 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 runs 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 runs 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 (that 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 running either in a V=R partition or in 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). 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/SP 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 one 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 system with a virtual=real machine, see "Specifying a Virtual=Real Machine" in Chapter 10.
ACCT	users with the ACCT option in their 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.

The channel mode setting for all channels except virtual channel zero can be changed by the CP DEFINE CHANNEL command.

`CPUID bbbbbb`

provides a processor identification (CPUID) to be stored in response to the STIDP instruction. It is necessary to associate a different 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. There is no checking by VM/SP to ensure that all virtual machines using the SET CPUID command have specified different 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.
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 useful only in the VM/SP 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 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 VM/SP dispatch queue. An affinity setting in the VM/SP directory can be overridden by the CP SET AFFINITY command. 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 the non-IPL processor is cancelled. Virtual machine processing may continue on the IPL processor.
`VMSAVE`	specifies that the virtual machine contents are to be saved if VM/SP is terminated or if VM/SP 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 you have 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. 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 support be enabled for the indicated virtual machine. See the table under "Using the Performance Options" in Chapter 10, for a list of VM/SP processors that support this 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`	specifies that Missing Interrupt Handler support be enabled for the indicated virtual machine. When a missing interrupt is detected, an interrupt will be simulated by CP.

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    │
       │ ALLOW   │
       └ ANY     ┘
```

where:

`userid`	is the one to eight-character user identification of the virtual machine or CP system service with which this virtual machine is authorized to establish a communication path.
`*CCS`	is the CP system service name required for VCNA.
`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.

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 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.

IPL Control Statement

The IPL control statement contains the name of the system to be loaded for the user when they log on. This statement is optional. If specified, it must follow the USER statement, and must precede the first device statement (CONSOLE, MDISK, 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 at logon time.

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`	passes up to 48 bytes of data after the keyword, PARM, excluding all leading and trailing blank characters, but including all embedded blanks, to your virtual machine's general purpose registers (4 bytes per register), starting with the high order byte of general register 0.
`Usage`	Although the VM/SP IPL command allows up to 64 characters on the PARM option, the directory restricts each statement to a single card image, limiting the number of characters that can be entered on the IPL Control Statement in the directory. The 'parmdata' on the IPL Control Statement is loaded into general registers 0-12 and is passed to the application specified by 'iplsys'.

SCREEN Control Statement

The SCREEN control statement is used to define the color and extended-highlight options for the user terminal. The SCREEN control statement is optional. If used, it must follow the USER control statement and precede the first device statement (CONSOLE, MDISK, DEDICATE, LINK, or SPOOL). The format of the SCREEN control statement is:

```
SCREen   area  {color [hilight]} {hilight [color  ]}....
                      [NONe   ]            [DEFault]
```

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 following may be used:

INArea	input area.
STArea	status area.
OUTarea	output areas. If this operand is specified, none of the following may be used:
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. Valid colors are:

BLUe	blue
RED	red
PINk	pink
GREen	green
TURquois	turquoise
YELlow	yellow
WHIte	white

values for hilight are:

NONe	none
BLInk	blink
REVvideo	reverse video
UNDerlin	underline

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.

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.

Usage Notes:

1. 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).
2. If the ALL operand is used, it must be the first operand specified on the first SCREEN control statement for the user.
3. If the OUTAREA operand is used, CPOUT, VMOUT, or INREDISP may not be specified.
4. 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
CPOUT	red	none
VMOUT	red	none
INREDISP	red	none
INAREA	green	blink
STATAREA	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

A 3270 specification forces TERMINAL CONMODE 3270 but is otherwise identical to a 3215 specification. Similar to the TERMINAL CONMODE 3270 specification, the devtype 3270 specification applies 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
- 3277 (Model 2)
- 3278 (Models 2, 3, 4, 5 2A, 2C, and 3C)
- 3279 (Models 2A, 2B, 2C, 3A, and 3B)
- 3036.

Only one console may 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 a user is disconnected.

`class` is a one-character spooling class. A through Z and 0 through 9 may be specified. 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.

For more information about defining consoles, see *VM/SP Operating Systems in a Virtual Machine.*

MDISK Control Statement

The MDISK control statement describes the cylinder extent to be owned by the user on a direct access device. The DASD area assigned with this statement becomes the user's minidisk. During logon, the owner of the minidisk obtains a link to it in the access mode specified on the MDISK control statement.

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 1 to 3 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
3375
3380
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.

```
⎧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 they log on. It is a part of their virtual configuration until they log off or detach 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 they and the VM/SP 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(s)**	**CMS/VSAM**	**CMS 800-byte Format**	**CMS 512, 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	-	558,000 blocks	not supported	558,000 blocks
3375	-	959 cyls.	182 cyls.	959 cyls.
3380	-	885 cyls.	121 cyls.	885 cyls.

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 IBCDASDI or Device Support Facilities for 3375/3380, the minimum minidisk size is two cylinders. For these devices, IBCDASDI 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 access mode that consists of up to two letters. The first letter specifies the primary access mode (read-only, write or multiple). The optional second letter indicates the alternate access mode (read-only or write access) desired if the primary access is not available. An optional 'V' character, when appended to the mode request on an MDISK statement, specifies virtual RESERVE/RELEASE processing. Valid modes are:

Mode	**Meaning**
R	Primary read-only access. The read-only link is established as long as no other user has the disk in write status. If there is an existing write link to the disk no link is given. R is the default mode if the link is to another userid.
RR	Primary read-only access or alternate read-only access. The read-only access is established even if another user has the disk in write status. The alternate access of R assures that the user will get the read link no matter what links currently exist to the disk.
W	Primary write access. The write link is established only if there are no other current links to the disk. If another user has the disk in read or write status no link is given.
WR	Primary write access or alternate read-only access. If write access is available then the link is established. Otherwise, the alternate access of a read-only link is given.
M	Primary multiple access. A write link is established unless another user already has write access to the disk, in which case no link is given.
MR	Primary multiple access or alternate read access. A write link is established unless another user already has write access to the disk, in which case a read link is given since it was the alternate access requested. **Note:** Unpredictable results can occur when one user has a read-only (R or RR) link to a device that is being updated by a user who has the device in write status (W or WR).
MW	Primary multiple access or alternate write access. A write link is established in all cases. CAUTION **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.)**

If a 'V' is appended to the immediate right of the primary access mode specification (or the alternate access mode specification, if any), then CP's virtual RESERVE/RELEASE support will be used in the I/O operations for the specified device. 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 MDISK statement, it defaults to W.

`pr`	is the password that allows other users to share the device in read-only mode (a one to eight-character field).
`pw`	is the password that allows another user to access the device in write mode (a one to eight-character field).
`pm`	is the password that allows other users to gain multiple access to the device (a one to eight-character field).

Notes:

1. A write password (pw) cannot be specified without a read password (pr). A multiple 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, any user is allowed to link with the corresponding access mode 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, VM/SP 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. Protecting minidisks by specifying passwords on the MDISK statement provides additional security for your VM/SP installation.

Examples:

```
MDISK 230 3380 5 10 WORK01 W ALL WRITE
```

is an MDISK statement for a minidisk with read/write access to 10 cylinders located on a real 3380 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 3380 50 15 CPDSK4 W RDPASS WRX2*
```

is an MDISK statement for a minidisk with read/write access to 15 cylinders located on a real 3380 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 by 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 the device type:

1403
2501
1443
3203
3211
2540 R[EADER]
2540 P[UNCH]
3262
3289
3525
3505
3800
4245

`class` is a one-character spooling class. The characters A through Z, 0 through 9, and * can be used. For spool output devices, the class governs the punching or printing of the real spooled output. If this operand is omitted, the default class A is used. This operand is required for all output devices defined on the spool record. 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 printer. (ww) indicates the hexadecimal width code of the paper. See Figure 16 for width codes. (`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 can have either two or four WCGMs. If 2WCGM is specified, the virtual 3800 printer is assumed to have two WCGMs. If 4WCGM is specified, four WCGMs are assumed. If neither is specified, 4WCGM is the default.

```
[ DATCK   ]
[ NODATCK ]
```

specifies processing of certain virtual 3800 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.

```
[ CFS ]
[ BTS ]
```

designates the stacker assumed for the virtual 3800 printer. You may specify either CFS (Continuous Form Stacker) or BTS (Burster Trimmer Stacker). If neither is specified, CFS is assumed.

Note: All parameters of the SPOOL control statement are positional. They must be specified in the order shown.

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)

Figure 16. Available Form Width Codes

When defining devices, make sure the devices are defined (and separated) within their own control unit range, and not shared with other 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 VM/SP. 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 log on gains access to it. The format of the DEDICATE control statement is:

```
DEDicate  ⎧ NETwork  cuu resource                                      ⎫
          ⎨          cuu⎧ rdev [VOLID] [volser] [3330V] [R/O]⎫         ⎬
          ⎩             ⎩      [VOLID] [volser] [3330V] [R/O]⎭         ⎭
```

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.
`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 NET is specified.
`rdev`	is the one to three 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 the user 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 alphameric 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 to or detach a dedicated 2305 from a user, all 8 exposures are processed.

2. Use caution in defining the hexadecimal addresses of virtual devices (cuu) 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. (Subchannel protocols for all devices supported by VM/SP are listed in Appendix B under the table heading "Shared Subchannel.") Devices should be grouped by control unit within a given channel according to their subchannel usage. Grouping devices that use the shared subchannel protocol together with devices that do not use the shared protocol can result in errors if all of the devices are using the same control unit. While the DEDICATE statement controls real devices, this restriction applies equally to real and virtual devices. The following is an example of a virtual machine's DEDICATE statements that can cause operational conflict.

   ```
   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 therefore will 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. Thus, the subsequent processing of a channel program involving these devices can result in a hung or busy condition (caused by a conflict in real-to-virtual I/O processing through the common VCUBLOK). Therefore, when defining devices, make sure the devices are defined (and separated) within their own control unit range and not shared with other devices.

   Since there is no control unit on the real hardware for a system console it should be noted that this restriction applies to any system console such as the 3138, 3148, 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 SPOOL statements and combinations of DEDICATE and SPOOL statements.

Remote 3270 Information Display System Printers can also be attached by the NETWORK ATTACH command. For more details see the *Operator's Guide.*

3. When the real device is a 3330V, the action VM/SP 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 VM/SP, transparent 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 VM/SP 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 previous one is that in this case the real device address is preselected and must have the VIRTUAL feature. This format allows you 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 previous one, except that the virtual device becomes a 3330V, such that VM/SP 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 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.

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.

Later, if you must move or change that volume, you need only update the one MDISK statement; the LINK statements need not be updated.

The LINK control statement and the MDISK control statement have the same authority level (neither has higher priority than the other). The format of the LINK control statement is:

```
Link  userid  vaddr1  [vaddr2 [mode]]
```

where:

`userid`	is the one to eight-character user identification of the user to be linked-to.
`vaddr1`	is the virtual device address of the device to be linked-to, which is owned by "userid." This virtual device address consists of three hexadecimal digits.
`vaddr2`	is the virtual device address that the device is to be linked-as for the virtual machine being defined. If not specified, "vaddr2" 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 access mode that consists of up to two letters. The first letter specifies the primary access mode (read-only, write or multiple). The optional second letter indicates the alternate access mode (read-only or write access) desired if the primary access is not available. Valid modes are:

Mode	**Meaning**
R	Primary read-only access. The read-only link is established as long as no other user has the disk in write status. If there is an existing write link to the disk no link is given. R is the default mode if the link is to another userid.
RR	Primary read-only access or alternate read-only access. The read-only access is established even if another user has the disk in write status. The alternate access of R assures that the user will get the read link no matter what links currently exist to the disk.

W	Primary write access. The write link is established only if there are no other current links to the disk. If another user has the disk in read or write status no link is given.
WR	Primary write access or alternate read-only access. If write access is available then the link is established. Otherwise, the alternate access of a read-only link is given.
M	Primary multiple access. A write link is established unless another user already has write access to the disk, in which case no link is given.
MR	Primary multiple access or alternate read access. A write link is established unless another user already has write access to the disk, in which case a read link is given since it was the alternate access requested. **Note:** Unpredictable results can occur when one user has a read-only (R or RR) link to a device that is being updated by a user who has the device in write status (W or WR).
MW	Primary multiple access or alternate write access. A write link is established in all cases. **CAUTION** **CMS supports multiple accessed read-only 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, but files on the disk that are altered by the write-mode user cannot be read by the other users.**

Note: If the mode is not specified, the default is R.

It is the responsibility of the operating system running in each virtual machine to keep data from being destroyed or altered on shared disks.

If userA owns a virtual device that was obtained via a directory MDISK statement:

```
MDISK 100 3380 5 10 VMDISK W READ WRITE
```

Then userB may have a directory LINK control statement to obtain this device at logon time:

```
LINK userA 100 200 RR
```

Any number of users may have directory LINK control statements to either userA's or userB's device. However, if userC has a directory LINK to userB's device (that 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 control statement or the LINK command) to this device through userC because no more than 2 levels of indirect directory links are permitted.

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)
`IBM TELE`	valid only if devtype is 2701, 2702, or 2703 For example, a virtual machine running 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: The Integrated Communications Attachment (ICA) on System/370 Models 135, 135-3, or 138 should be specified as a 2701.

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 CMS/DOS, you should update the VM/SP directory entries so the VSE system residence volume and the VSE private libraries are shared by all CMS/DOS users.

The VM/SP directory entry for one CMS virtual machine should contain MDISK statements defining VSE volumes. VM/SP directory entries for other CMS/DOS users should contain LINK statements.

For example, assume the VSE 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 3330 volume labeled DOSPRI. Then one CMS machine (for example, DOSUSER1) would have the MDISK statements in its directory entry.

```
USER DOSUSER1 password 1M 2M G
   .
   .
   .
   MDISK 331 3330 0 150 DOSRES R rpass
   MDISK 231 3330 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
```

For more information about directory entries for CMS/DOS virtual machines, see *VM/SP Operating Systems in a Virtual Machine.*

Note: Refer to the *VM/SP Installation Guide* for a list of the sample directories supplied with the Product Tape.

Chapter 19. 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 processor. VM/SP 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.

You can include entries for more devices than you have so devices can be added in the future without performing another system generation. Bear in mind, however, that the control blocks generated (RDEVBLOK, RCUBLOK, and RCHBLOK) occupy space in real storage.

For the 3081 Processor Complex, in addition to preparing the real I/O configuration file, you must prepare the input/output configuration program source file and run the Input/Output Configuration Program to define the I/O configuration to the processor. See "Coding Considerations for the Input/Output Configuration Program Source File" later in this chapter, for more information.

When preparing the RDEVICE and RCTLUNIT entries, refer to "Appendix B. Configuration Aid" 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 come before 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."

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 stand-alone display stations. The clustered configurations attach to either a 3271, 3274 Model 1C, or 3276 control unit. The stand-alone station is a 3275 display station that 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 different 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=cuu operand of the RDEVICE macro) is coded in the LINE=cuu 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 (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 (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, 3287 (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 3276) on a leased BSC line, or a stand-alone 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 come before all other macros in the DMKRIO file.

The format of the CLUSTER macro is:

Name	Operation	Operands
label	CLUSTER	CUTYPE= $\left\{\begin{array}{l}3271\\3274\\3275\\3276\end{array}\right\}$,GPOLL=cudv ,LINE=cuu ,DIAL= $\left\{\begin{array}{l}\text{YES}\\\text{NO}\end{array}\right\}$

where:

`label`

is a name of the CLUSTER macro. It must be specified. The label may be any assembler language symbol. The label establishes a special symbolic name for this cluster control unit or stand-alone station.

CUTYPE= $\left\{\begin{array}{l}3271\\3274\\3275\\3276\end{array}\right\}$

is the station control unit. It is either 3271, 3274 Model 1C, 3275, or 3276.

`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 a 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 17 on page 198 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 you to match polling and selection address characters shown in Figure 17 .

`LINE=cuu`

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 come before TERMINAL macros that define stations attached to that cluster or stand-alone 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 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, and 3289s in groups of 8 until all 3278s, 3279s, 3287s, and 3289s 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
label	TERMINAL	TERM=$\left\{\begin{array}{l}\texttt{3275}\\\texttt{3276}\\\texttt{3277}\\\texttt{3278}\\\texttt{3279}\\\texttt{3284}\\\texttt{3286}\\\texttt{3287}\\\texttt{3288}\\\texttt{3289}\end{array}\right\}$,SELECT=cudv $\left[\begin{array}{l}\texttt{,MODEL=1}\\\texttt{,MODEL=2}\\\texttt{,MODEL=3}\\\texttt{,MODEL=4}\\\texttt{,MODEL=5}\end{array}\right]$ [,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 come before all other macros in the DMKRIO file.

where:

$$\texttt{TERM=}\left\{\begin{array}{l}\texttt{3275}\\\texttt{3276}\\\texttt{3277}\\\texttt{3278}\\\texttt{3279}\\\texttt{3284}\\\texttt{3286}\\\texttt{3287}\\\texttt{3288}\\\texttt{3289}\end{array}\right\}$$

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 17 on page 198 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.

```
┌ 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 doesn't match the real device, unpredictable results may occur.

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 stand-alone 3275 station:

- IBM 3284 Printer Model 3
- IBM 3286 Printer Model 3 (via RPQ MB4317)

`FEATURE=OPRDR`

specifies the optional operator identification card reader feature, 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, 3A, and 3B.

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
```

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 17. Remote 3270 Control Unit and Device Addressing

3271, 3274, and 3276 Addressing			3275 Addressing		
		EBCDIC			EBCDIC
General Poll for Control Unit 5	Control Unit Address	C5 C5	General Poll for Control Unit 5	Control Unit Address	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 18. Examples of Remote 3270 Addressing

Figure 18 shows some examples of valid polling characters.

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 system is 4096.

RDEVICE macro instructions describe each device, or group of devices, attached to your processor. These can be in any order (except when used in conjunction with the CLUSTER macro[10]). They must be contiguous and must come before 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 RDV234 is generated for the device address 234.

Before you code an RDEVICE macro for a 3704 or 3705 device, see "Special Considerations for Coding the 3704/3705 RDEVICE Macro" in this chapter, for additional information and special considerations.

The RDEVICE macro statement is not used for SNA supported terminals.

---

[10] See "Chapter 12. Planning for 3270s" before you code an RDEVICE macro for a binary synchronous line used by remote 3270s.

The format of the RDEVICE macro is:

Name	Operation	Operands
label	RDEVICE	ADDRESS={cuu (cuu,nn)},DEVTYPE=type[,MODEL=model] [,FEATURE=(feature[,feature]...)] [,CLASS={(cl[,cl]...) DASD TAPE TERM GRAF URI URO}] [,ADAPTER={BSCA IBM1 SDLC TELE2 TYPE1 TYPE2 TYPE3 TYPE4}] [,CPTYPE={EP NCP PEP}] [,ALTCU=cuu] [,SETADDR=sadnum] [,CPNAME=cpname] [,BASEADD=cuu] [,CLUSTER=label] [,IMAGE=imagelib] [,CHARS=ffff] [,FCB=lpi] [,DPMSIZE=n]

where:

ADDRESS={cuu
(cuu,nn)}

is the real I/O device address (or addresses).

cuu is three-hexadecimal digits from 000 to FFF. The high-order digit is the address of the channel to which the device is attached. The low-order two digits represent the control unit and device address.

nn 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 nn 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. The maximum value of nn is 16.

If DEVTYPE=3066, 3138, 3148, or 3158, or if DEVTYPE=3278 and Model=2A, nn can only be 1. This is because only one system display console can be specified for each RDEVICE macro.

When using more than one 3705, you must generate each unit independently so that each has the controller attribute. Otherwise, VM/SP considers the 3705s to be 270x lines and, therefore, not usable by guest virtual machines.

`DEVTYPE=type`

is the type of device.

The device type can be CTCA, HFGD, ICA, 1017, 1018, 1052, 1053, 1403, 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, 3262, 3268, 3277, 3278, 3279, 3284, 3286, 3287, 3288, 3289, 3330, 3333, 3340, 3350, 3375, 3380, 3410, 3411, 3420, 3430, 3505, 3525, 3704, 3705, 3800, 3851, 4245, 4250, 7443, 8809, or FB-512.

Coding Considerations

Additional information relating to 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, or 3705. 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 that follows.

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. Note that while a DEVTYPE specification of 3289 and a MODEL specification of 4 is allowable, the result will be the generation of a graphic device rather than a system printer.

Since a CTCA may tie up a channel, it is recommended that you 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:

Processor	System Console
135, 135-3, 145, 145-3, 155 II	3210 or 3215
138	3138 (if in display mode) 3215 (if in printer-keyboard mode)
148	3148 (if in display mode) 3215 (if in printer-keyboard mode)
158	3158 (if in display mode) 3215 (if in printer-keyboard mode and has the 3213 Printer Model 1)
165 II, 168	3066
3031, 3032, 3033, 3033-N, 3033-S, 3042	3036
4331, 4341	3278 Model 2A (if in display mode) 3215 (if in printer-keyboard mode)
3081	3278 Model 2A (if in display mode)

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 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, 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 VM/SP directory 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 number for a particular device.

Model number, if not specified, defaults to zero except for the 3203, which defaults to 4. It must be coded for 2305, 3330, and 3333 DASD, 3278, and 3279 display devices, and 3203, 3289, and 3262 printers. 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 or 11	3262
1, 2, or 11	3330
1 or 11	3333
A1 - H8	3704, 3705-I, or 3705-II
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
2, 3, 4, or 5	3278
2A	3278 consoles for 4300 processors
2C	3279 consoles for 4300 processors
2 or 3	3279
4	3289

Notes:

1. For a 3704/3705 outside the A1 - H8 model range, specify MODEL=H8.
2. The 3277 Model 1 is a 480-character display screen and is supported by VM/SP only as a dedicated device.
3. If a model number is included for devices that do not require model numbers, system generation is ended with an error message.
4. If DEVTYPE=3278 or 3279, MODEL= *must* be specified.
5. Specify 3278 Model 2A for 4300s equipped with 3279 Model 2C consoles.

`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

Feature	Explanation
SYSVIRT	A 3330V (DEVTYPE=3330) device that may be used by VM/SP 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
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 device 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 by inserting a magnetically encoded card.

Use the FEATURE=OPRDR operand of the RDEVICE macro to specify that this is a display device with a card reader. FEATURE=OPRDR is invalid if DEVTYPE=3158.

Notes:

1. The 7-TRACK, CONV, DUALDENS, and TRANS features are not allowed for the 8809.
2. The 7-TRACK, CONV, and TRANS features are not allowed for the 3430.

Although allowable, it is 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.

FEATURE = FH is valid only for a 3350 DASD device or a 3330 in emulation mode. For all other DASD devices that may have the FH feature installed, it is either provided by the device type (e.g. 2305) or determined at IPL or VARY ONLINE time.

Specifying FEATURE=(2CHANSW/4CHANSW) on the RDEVICE macro to indicate hardware support of reserve/release CCWs is unnecessary. DMKCPI determines this 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 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.

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, or 3211 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 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 for VM/SP. It does not appear in the list of VM/SP 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 stopping 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
```

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, 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, or 3705. 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 or 3705.

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
- A Line Interface Base Type I in a 3704 or 3705

TYPE1 specifies the channel adapter accessed by a 3704. For DEVTYPE=3705, TYPE 1 or TYPE4 may be coded. In identifying the channel adapter, TYPE1 or TYPE4 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 or TYPE4.

`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  }
       { NCP }
       { PEP }
```

is the 3704/3705 control program to be run in a 3704 or 3705 Communications Controller.

`EP` specifies the 2701, 2702, or 2703 Emulation Program.

`NCP` specifies the Network Control Program.

`PEP` specifies the Partitioned Emulation Program.

`ALTCU=cuu`

specifies an alternate control unit address to be used if paths through the primary control unit are unavailable. cuu is a three-digit hexadecimal address. Only one ALTCU can be specified.

The ALTCU cuu 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 only 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 system, you must specify different control unit addresses.

The ALTCU cuu 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 cuu should identify the logical RCUBLOK in VM/SP. VM/SP constructs one RCUBLOK for each set of sixteen devices supported by the real control unit.

Assuming an alternate control unit configuration where each of two control units support thirty-two devices, the following two macro definitions are acceptable:

```
RDEVICE ADDRESS=(300,32),ALTCU=400
RCTLUNIT ADDRESS=300,FEATURE=32-DEVICE
RCTLUNIT ADDRESS=400,FEATURE=32-DEVICE
RCHANNEL ADDRESS=3
RCHANNEL ADDRESS=4

RDEVICE ADDRESS=(300,16),ALTCU=400
RDEVICE ADDRESS=(410,16),ALTCU=310
RCTLUNIT ADDRESS=300,FEATURE=32-DEVICE
RCTLUNIT ADDRESS=400,FEATURE=32-DEVICE
RCHANNEL ADDRESS=3
RCHANNEL ADDRESS=4
```

`CPNAME=ncpname`

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.

Note: Failure to code the CPNAME operand on the RDEVICE macro for the 3704/3705 base address causes VM/SP to mark the device "not operational" at IPL time. The cluster on that 3704/3705 is therefore unusable.

`BASEADD=cuu`

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 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,           X
        CLASS=(A,C)
RDEVICE ADDRESS=(0C0,4),DEVTYPE=2401
RDEVICE ADDRESS=(030,8),DEVTYPE=2702,ADAPTER=TELE2,          X
        SETADDR=2
```

Special Considerations for Coding the 3704/3705 RDEVICE Macro

The 3704/3705 Communications Controllers have varied uses. Consequently, there are special considerations for coding the RDEVICE macro.

IBM 3704 Communications Controller

Model	Storage
A1	16K
A2	32K
A3	48K
A4	64K

IBM 3705 Communications Controller

	Model	Storage
3705-I	A1, B1, C1, D1	16K
	A2, B2, C2, D2	48K
	B3, C3, D3	80K
	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 19. IBM 3704/3705 Models

Note: Specify any 3704/3705 outside the range of models listed in Figure 19 as MODEL=H8.

EP-Only Control Programs: If the 3704/3705 is to be run in emulation mode:

- Use the (cuu,nn) form of the ADDRESS operand to generate multiple RDEVBLOKs.

- Specify the appropriate name for CPNAME.

To generate additional emulator lines for the same 3704/3705, 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 (cuu,nn) 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.

Other 3704/3705 RDEVICE Considerations: For each physical 3704/3705 there should be only one RDEVICE macro that specifies the ADAPTER=TYPE1, TYPE2, TYPE3, or TYPE4, MODEL, CPTYPE, and CPNAME operands. This RDEVICE macro defines the base address of the 3704/3705 (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=TYPE1, TYPE2, TYPE3, or TYPE4, MODEL, and CPTYPE operands. There must *never* be a second use of the CPNAME operand. Even if CPTYPE=EP is specified, the 3704/3705 base address cannot be used as a telecommunication line. Its function is only to load and dump the 3704/3705. 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 that the EP control program may use.

If you have a 3704/3705 and a 2701/2702/2703 on the same VM/SP system, the virtual addresses for the 3704/3705 must not be the same as any of the real 2701/2702/2703 addresses.

Examples:

Examples of RDEVICE macro specifications follow.

Example 1

```
RDEVICE ADDRESS=(020,16),                                   X
        DEVTYPE=3704,                                       X
        MODEL=A2,                                           X
        ADAPTER=TYPE1,                                      X
        CPNAME=VMEP01
```

This describes a 32K 3704 at address X'020', with 15 emulator lines addressed 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),                                    X
        DEVTYPE=3704,                                        X
        ADAPTER=IBM1,                                        X
        SETADDR=2,                                           X
        BASEADD=020
```

This describes an additional 16 emulator lines on the same 3704 specified by Example 1.

3704/3705 Error Messages: The RDEVICE macro instruction generates an entry in a table of programmable communications controllers when DEVTYPE=3704 or 3705 is specified. This table has a maximum of 10 entries. The following message results if more than ten 3704 or 3705 devices are specified:

```
MORE THAN 10 TP CONCENTRATORS
```

This message indicates that the RDEVBLOK is generated, but no entry is made in the Programmable Communications Controller table.

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, 1443, 2501, 2540P, 2540R, 3203, 3211, 3262, 3289E, 3505, 3525, 3800, or 4245 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
```

Error 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, an 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.

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, providing you have enough real storage to hold the real control unit blocks (RCUBLOKs). Control units generally fall into two classes: 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 can be divided by eight. 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/punch and the system console 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.

VM/SP 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.

Error Example:

Device Addresses 150-157 and 160-167 on first 3830-2
Device Addresses 158-15F and 168-16F on second 3830-2

This address scheme is *not supported* by CP. All addresses on a 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 an RCTLUNIT macro coded for it. For example, if a 3210 is defined with an RDEVICE macro, it must have a corresponding RCTLUNIT macro.

The 3880 Storage Control Unit contains two director modules. Each director module acts as a control unit providing input/output operations to a string of devices. Since each director module is separately addressable, one RCTLUNIT macro statement is required for each module. The optional ALTCH operand of the RCTLUNIT macro allows specification of up to three alternate channel interfaces to a single director module. VM/SP supports a maximum of four channel paths to a single director module.

RCTLUNIT macro instructions describing the control units for your 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, RCU230 is generated.

The format of the RCTLUNIT macro is:

Name	Operation	Operands
label	RCTLUNIT	ADDRESS= cuu ,CUTYPE=type [,ALTCH=(n,n,n)] [,FEATURE=xxx-DEVICE] [,UCW= { SHR } { UNS }]

where:

ADDRESS=cuu

> is the real address of the control unit. cuu consists of 3 hexadecimal digits. The high order digit is the channel address of this control unit. The low order two digits must be the lowest address of the control unit. The first digit may be any hexadecimal number from 0 to F. If the xxx-DEVICE feature is supported, the low order digit must be 0. Otherwise, it must be either 0 or 8.
>
> **Note:** If you have a 2701, 2702, or 2703, and a 3704 or 3705 running in emulation mode, make 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, 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, 3262, 3272, 3274, 3345, 3411, 3430, 3505, 3525, 3704, 3705, 3803, 3811, 3830, 3880, 3851, 4245, 7443, CTCA, DPA, FTA, HFCU, ICA, IFA, ISC, SVPC.
>
> Only one CTCA may be defined at a time.

In addition, any other control unit that can be attached to a real processor may be specified in an 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 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, an 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 an 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 important.

`ALTCH=(n,n,n)`

specifies the alternate channel(s) to be used with the control unit address if the primary channel path is unavailable or offline. n represents the one-digit channel addresses for the alternate channel paths. You can specify up to three alternate channels for AP or UP systems. Only one alternate channel can be specified 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, 3088, 3272, 3274, 3345, 3704, 3705, 3803, 3830, 3880, FTA, HFCU, ICA, IFA, or ISC.

The prefix xxx for a 3088 must be either 32 or 64.

The Integrated File Adapter's 9821 feature, when used with 3344s, 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 x00, x20, x40, etc. (where x is the channel number); always an even multiple of x20 (decimal 32). Likewise, for 64 devices, the only correct addresses are x00, x40, x80, and xC0.

This information is important when an upgrade of installed DASD controllers occurs. Make sure the ADDRESS value is changed in addition to 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."

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 you have control units that share more than 16 devices that are 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.

$\texttt{UCW=}\left\{\begin{array}{l}\texttt{SHR}\\ \texttt{UNS}\end{array}\right\}$

is the attribute of the UCW. This can be specified only for 3272 or 3274 control units.

SHR indicates a shared UCW

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 Messages

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 error message is issued when one RCTLUNIT macro is coded (specifying the ALTCH operand), and an additional RCTLUNIT macro is coded, which creates an RCUBLOK for the alternate channel address (specified by the first RCTLUNIT macro).

```
CHANNEL TABLE FOR RCUxxx IN USE BY RCUyyy
```

Error 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 RCTLUNIT macro and RDEVICE macro for 350 must be removed.

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 RCHANNEL macro instructions describing your channels may be in any order, but they must be contiguous and follow all of the RCTLUNIT macro instructions in 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 RCHAN2 is generated.

The format of the RCHANNEL macro is:

Name	Operation	Operands
label	RCHANNEL	ADDRESS=address ,CHTYPE=$\left\{\begin{array}{l}\text{SELECTOR}\\ \text{MULTIPLEXOR}\\ \text{BLKMPXR}\\ \text{FTA}\end{array}\right\}$

where:

`ADDRESS=address`
: is the real address of the channel. It is a hexadecimal number from 0 to F.

CHTYPE=$\left\{\begin{array}{l}\text{SELECTOR}\\ \text{MULTIPLEXOR}\\ \text{BLKMPXR}\\ \text{FTA}\end{array}\right\}$

is the type of channel.

`SELECTOR` indicates a selector channel.

`MULTIPLEXOR` 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=MULTIPLEXOR
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
label	RIOGEN	CONS=cuu [,ALTCONS=(cuu[,cuu,cuu...])] [,SRF=(cuu[,cuu,cuu...])]

where:

`CONS=cuu`

is the address of the VM/SP 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=(cuu[,cuu,cuu...])]`

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 located as close as possible to the primary system console, may be any device supported as a VM/SP logon device (except for those remote terminals connected via 3704/3705 Communications Controllers). If the primary system console is not operational at VM/SP 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 VM/SP 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 Communications Controller unless the 3704/3705 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=(cuu[,cuu,cuu...])]`

is the address or a list of addresses of SRF (service record file) devices used for the 3031, 3032, or 3033 processors. cuu 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 systems 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 one time by the VM/SP control program. 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), and a system console (009) with no alternate console.

```
RIOGEN CONS=01F,ALTCONS=050
RIOGEN CONS=009
```

Example of Coding the Real I/O Configuration File (DMKRIO)

In this example, macros are coded to support the following real devices:

1 2540 Card Reader/Punch
1 3505 Card Reader
1 3525 Card Punch
2 1403 Printers with the Universal Character Set feature
1 3211 Printer with the Universal Character Set feature
1 3215 Console Printer-Keyboard
1 2955 Data Adapter Unit
7 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 VM/SP 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

Figure 20 shows the real configuration. The real I/O configuration file that supports this example is shown in Figure 21.

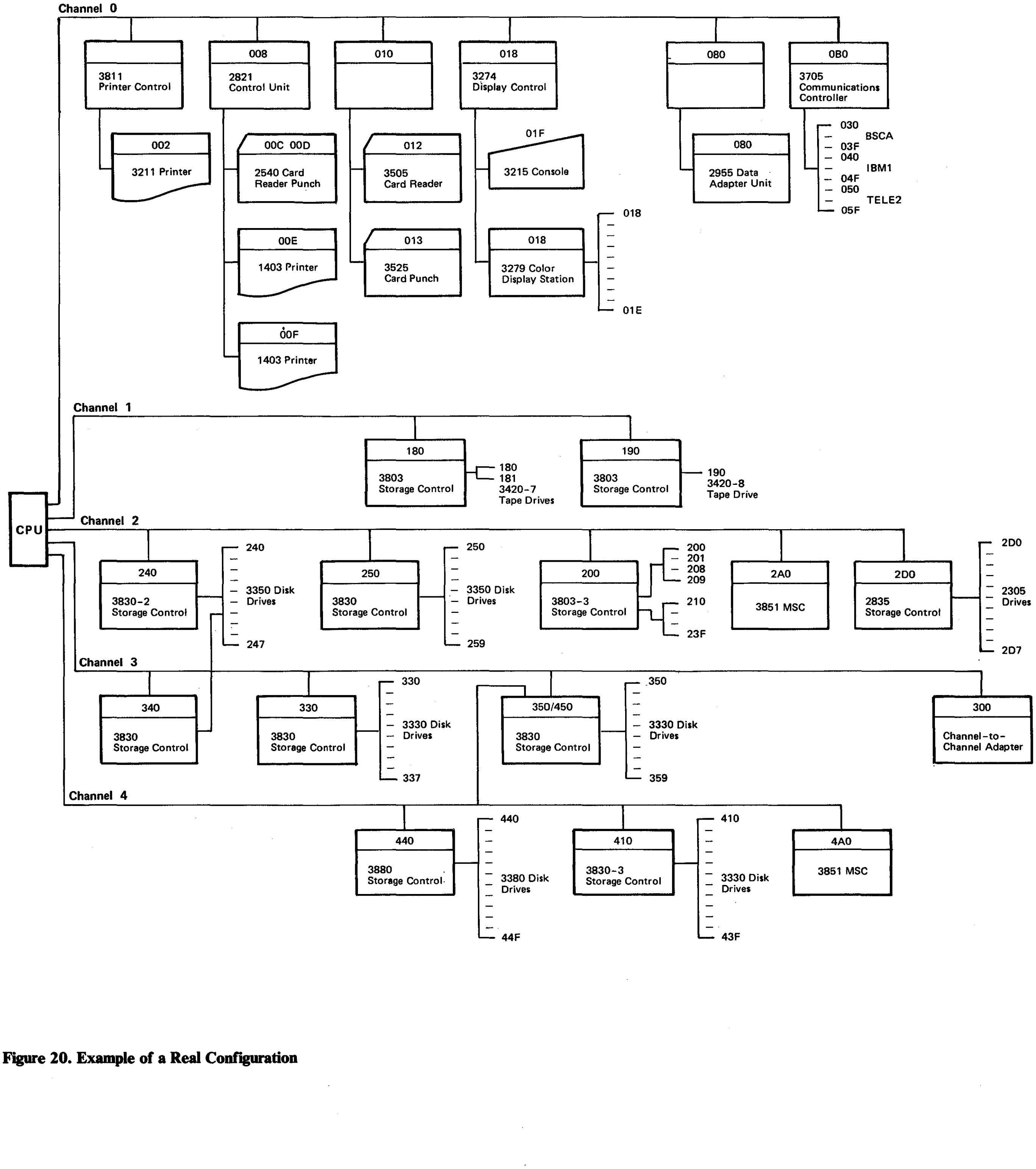


Figure 20. Example of a Real Configuration

```
DMKRIO CSECT
       COPY OPTIONS
       RDEVICE ADDRESS=002,DEVTYPE=3211,CLASS=(X,A),FEATURE=UNVCHSET
       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=012,DEVTYPE=3505
       RDEVICE ADDRESS=013,DEVTYPE=3525,CLASS=(X,A)
       RDEVICE ADDRESS=(018,7),DEVTYPE=3279,MODEL=3
       RDEVICE ADDRESS=01F,DEVTYPE=3215
       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
       DEVICE ADDRESSES 200, 201, 208, 209 ALLOW ACCESS TO MSC TABLES
       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
       RCTLUNIT ADDRESS=000,CUTYPE=3811
       RCTLUNIT ADDRESS=008,CUTYPE=2821
       RCTLUNIT ADDRESS=010,CUTYPE=3505
       RCTLUNIT ADDRESS=018,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
       RCTLUNIT ADDRESS=2A0,CUTYPE=3851
       RCTLUNIT ADDRESS=2D0,CUTYPE=2835
       RCTLUNIT ADDRESS=330,CUTYPE=3830
       RCTLUNIT ADDRESS=340,CUTYPE=3830
       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
       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=018
       END
```

Figure 21. Real I/O Configuration File

Considerations for Coding the Input/Output Configuration Source File

If you are generating VM/SP on a 3081 Processor complex, you must define additional I/O information to the processor controller. To define the I/O configuration to the 3081 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 match the real I/O configuration file.

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; however, you must insure that devices and control units are defined under the appropriate channel macros. If not, IOCP fails with an error message.

Do not intermix IOCP macro definitions and VM/SP I/O macro definitions in the same file. You must maintain two distinct files; the real I/O configuration file and the corresponding IOCP source file. See the *OS/VS2 MVS, VM/SP and Stand-Alone Versions Input/Output Configuration Program User's Guide and Reference* for IOCP source file coding conventions.

Figure 22 is an example of the IOCP source file needed to match the real I/O configuration defined in Figure 21. For additional information about the Input/Output Configuration program, refer to "Chapter 8. Planning for Virtual Machine Operating Systems (Other than CMS)."

Example: Coding the Input/Output Configuration Source File

```
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=3811,                       X
      UNITADD=((08,8))
CNTLUNIT CUNUMBR=002,PATH=00,SHARED=N,UNIT=3505,                       X
      UNITADD=((10,8))
CNTLUNIT CUNUMBR=003,PATH=00,SHARED=N,UNIT=3274,                       X
      UNITADD=((18,8))
CNTLUNIT CUNUMBR=004,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=((40,8))
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=((30,8))
CNTLUNIT CUNUMBR=032,PATH=(03,04),SHARED=N,UNIT=3830,                  X
      UNITADD=((50,16))
CNTLUNIT CUNUMBR=033,PATH=03,SHARED=N,UNIT=CTCA,                       X
      UNITADD=((C0,8))
CNTLUNIT CUNUMBR=034,PATH=03,SHARED=N,UNIT=3830,                       X
      UNITADD=((40,8))
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))
CNTLUNIT CUNUMBR=042,PATH=04,SHARED=N,UNIT=3851,                       X
      UNITADD=((A0,8))
IODEVICE ADDRESS=002,UNIT=3211,CUNUMBR=000
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=012,UNIT=3505,CUNUMBR=002
IODEVICE ADDRESS=013,UNIT=3525,CUNUMBR=002
IODEVICE ADDRESS=(018,7),UNIT=3279,MODEL=3,CUNUMBR=003
*IOCP    DUMMY DEVICE FOR CP ONLY
*IOCP    IODEVICE ADDRESS=01F,UNIT=3215,CUNUMBR=003
```

Figure 22 (Part 1 of 2). IOCP Source File

```
IODEVICE ADDRESS=(030,16),UNIT=3705,CUNUMBR=004
IODEVICE ADDRESS=(040,16),UNIT=3705,CUNUMBR=004
IODEVICE ADDRESS=(050,16),UNIT=3705,CUNUMBR=004
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=(021,034)
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=032
IODEVICE ADDRESS=(358,2),UNIT=3330,MODEL=11,CUNUMBR=032
IODEVICE ADDRESS=3D0,UNIT=CTC,CUNUMBR=033,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
```

Figure 22 (Part 2 of 2). IOCP Source File

Chapter 20. 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 (DMKSNT) is a pageable module. It may not exceed one page (4K) in size.

Note: For a list of the DMKSNT files supplied with the Product Tape, refer to the *VM/SP Installation Guide.*

The sample DMKSNT files each have seven entries: one each for saving copies of CMS, CMSL, CMSVSAM, CMSAMS, CMSDOS, INSTVSAM, and CMSBAM. The CMSL entry can be removed from your DMKSNT if you have no need of a CMS shared system located in high virtual storage. If you use all recommended labels and allocations and the starter system supplied DMKSYS, you can save these segments during the system generation procedure. The INSTVSAM segment is a CMSDOS segment that is used only during the procedure for loading and saving CMSVSAM, CMSAMS and CMSBAM segments. For an explanation of this procedure, and for an illustration of the storage layout resulting from this suggested configuration, see "Loading and Saving Discontiguous Saved Segments" in the *VM/SP Installation Guide.*

If you wish to change or add to the suggested system name table, code your own macro and create a DMKSNT file of your own. One entry can be created for each type of discontiguous segment.

If you create your own version of the system name table, your file must have a CSECT and END statement:

```
DMKSNTBL CSECT
         NAMESYS  macros (one for each virtual machine oper-
                  ating system or segment you wish to save)
         NAMENCP  macros (one for each 3704/3705 control
                  program you create)
         NAME3800 macros (one for each 3800 named system you
                  create)
         END
```

The loader automatically inserts a PUNCH SPB (Set Page Boundary) card to force this module to a 4K boundary when the CP system is built. 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={(cc,p),} [SYSPGCT=pppp,]
                                 {pppppp,}
                         SYSPGNM=(nn,nn,nn-nn,...),
                         SYSHRSG=(s,s,...),
                         PROTECT={OFF}
                                 {ON }

                         [USERID=userid,][RCVRID=rcvrid,][SAVESEQ={   10    }]
                                                                  {priority}
```

where:

`label` is any desired user label.

`SYSSIZE=nnnnnK`

where nnnnnk represents the least amount of 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) of 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 you code 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. "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 (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 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. This operand is flagged and ignored if USERID= is specified. For fixed-block DASD, specify the SYSBLOK= parameter. This operand is flagged and ignored if USERID= is specified.

`SYSSTRT=(cc,p),`
`        pppppp,`

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 (cc,p), where cc designates the starting cylinder address and p the starting page address. For fixed-block SYSVOL devices, specify pppppp, representing the starting page address. These numbers are specified in decimal.

The number of pages written to this area is the total number specified on the SYSPGNM operand, plus one information page. This operand is ignored if VSYSADR=IGNORE, but you must specify it as null (SYSCYL=, or SYSBLK=,).

`SYSPGCT=pppp`

is the total number of pages (pppp) to be saved (that is, the total number of pages you indicate via the SYSPGNM operand). This is a decimal number, up to four digits. 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=`$\begin{Bmatrix}\texttt{OFF}\\ \texttt{ON}\end{Bmatrix}$

indicates that VM/SP 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 the VM/SP control program; the change will be seen by all users of the shared page.

`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 USERID indicates that this is a VMSAVE target area specification.

Note: More than one target area 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=`$\begin{Bmatrix}\texttt{10}\\ \texttt{priority}\end{Bmatrix}$

specifies the order in which multiple virtual machines will be saved. Values can be 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.

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

Information on the following subjects is in *VM/SP System Programmer's Guide*:

- Determining when to save a system
- Using the SAVESYS command
- Saving the CMS system
- Saved system restrictions for CMS
- Saving OS
- Using discontiguous saved segments (CMSDOS, CMSVSAM, CMSAMS, CMSBAM)

Coding the NAMENCP Macro

You must create an entry in the system name table (DMKSNT) for each separate 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. In this way, you do not have to regenerate the VM/SP 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 your 3704/3705. 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)}` `{pppppp }`	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. For fixed-block SYSVOL devices, specify pppppp representing the starting page address. These numbers are specified in decimal.

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. More than one named system 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)`

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 contains several core image members. Find the size of each core image member that was created by GENIMAGE. Bytes seven and eight of the core image contain the 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=(ccc,p)`

is the starting cylinder (ccc) and page address (p) on SYSVOL at which this image library is to be saved. These numbers must be specified in decimal.

Chapter 21. 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
- System operator's user identification
- System timer value
- System pointer variables
- Automatic performance monitoring parameters
- Accounting parameters
- System identification
- System spool parameters
- Security journaling parameters
- System dump space parameters
- System T-DISK security parameters
- Missing I/O interruption timer intervals

You are 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 format of the DMKSYS file follows:

```
DMKSYS  CSECT
        SYSOWN   macro
        SYSRES   macro
        SYSOPR   macro
        SYSCOR   macro
        SYSTIME  macro
        SYSMON   macro
        SYSJRL   macro
        SYSACNT  macro
        SYSFORM  macro
        SYSPCLAS macro
        SYSID    macro
        SYSORD   macro
        SYSMIH   macro (optional)
        SYSLOCS  macro
        END
```

Notes:

1. Sample DMKSYS files are provided with the starter system, and are listed in the *VM/SP Installation Guide*. If you use these, you can save the CMS system at the end of system generation. The VM/SP installation procedure prompts you to ask if you want the sample DMKSYS file punched. You may modify this file if you wish. For example, you could modify the starter system DMKSYS file to add other CP-owned volumes.

2. SYSLOCS must always be the last macro coded.

3. You must code all macro statements, except for SYSMIH, even if you code no operands on some macro statements. SYSMIH is optional.

4. If the SYSMIH macro is coded, it must be placed immediately before the SYSLOCS macro.

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 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 running 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 user's disks.

- Try to keep read-only minidisks (for example, the CMS system residence disk and source disks) that are frequently accessed on separate volumes from users' read/write minidisks. If possible, also keep them on separate channels.

- If you have disks with fixed head areas (2305/3340/3350/3380) you should use them for preferred paging. To do this, allocate this area as preferred paging space using the Format/Allocate program. For more details see the *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 initially 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.

- Using the automatic monitoring facilities, study the load conditions and performance profile for your system as soon as possible. These facilities, used with programs similar to the IBM FDP (Field Developed Program) Virtual Machine Facility/370: Performance/Monitor Analysis Program are designed to make data collection and reduction easy, thus 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.

- You can determine the monitoring interval for the missing interruption handler support. Some classes of devices may require an interval other than the default interval provided in DMKSYS. For instance, if you have many direct storage devices, you may need to lengthen the time interval for the DASD class. This would eliminate the 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 change the time intervals with the SET MITIME command.

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 VM/SP 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 a VM/SP 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 SYSOWN macro statement does not distinguish preferred from nonpreferred cylinders on a volume. Use the SYSORD macro statement to specify the order in which the preferred/nonpreferred devices are to be searched to satisfy paging and spooling operations. The CP Format/Allocate service program accepts PAGE and TEMP as operands on the control statements to designate preferred and nonpreferred paging areas (spool, page overflow).

- 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 not available to VM/SP. 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, build the CP nucleus, and reload it on the system residence volume or on an alternate IPL device. Use the GENERATE EXEC procedure to reassemble DMKSYS and reload the CP nucleus. Refer to the *VM/SP Installation Guide* for details on using the GENERATE EXEC.

- If you have saved systems (systems that can be loaded by name, thus not performing 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 system has a 3704 or 3705, you must reserve space on a CP-owned volume to contain the 3704/3705 control program image. See "Chapter 13. Planning for the 3704/3705 Control Program" for information about how much DASD space you should reserve.

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 allows 2314 or 2319 "alternate track" cylinders 200-202 to be used for normal data if they are not needed to replace defective tracks.
- On a 3340, "alternate track" 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 VM/SP SYSRES volume.

The format of the SYSRES macro is:

Name	Operation	Operands
	SYSRES	SYSVOL=serial, SYSRES=[(]{address [,altaddr]}[)] SYSTYPE={2305 2314 2319 3330 3340 3350 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]})`

designates a three-digit hexadecimal device address (cuu) 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 allows 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.

$$\texttt{SYSTYPE=}\left\{\begin{array}{l}\texttt{2305}\\\texttt{2314}\\\texttt{2319}\\\texttt{3330}\\\texttt{3340}\\\texttt{3350}\\\texttt{3375}\\\texttt{3380}\\\texttt{FB-512}\end{array}\right\}$$

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.

$$\texttt{SYSNUC=}\left\{\begin{array}{l}\texttt{strtcyl}\\\texttt{strtpage}\end{array}\right\}\texttt{,}$$

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 three-digit decimal number.

`strtpage` is a one to six-digit decimal number.

Normally,

- A 2314 or 2319 device requires seven contiguous cylinders
- A 2305 or 3340 device requires ten contiguous cylinders
- A 3330/3333 device requires four contiguous cylinders
- A 3350 device requires two cylinders
- A 3375 device requires three cylinders
- A 3380 device requires two cylinders
- An FB-512 requires approximately 220 pages

to contain the CP nucleus.

$$\texttt{SYSCLR=}\left\{\begin{array}{l}\underline{\texttt{YES}}\\\texttt{NO}\end{array}\right\}$$

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 zeroes 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 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).

The strtcyl is a one to three-digit decimal number designating the starting cylinder of the error recording area.

The cylcount is a one-digit decimal number between 2 and 9 designating the number of cylinders.

The strtpage is a one to six-digit decimal number designating the starting page of the error recording area.

The pagecount is a one to six-digit decimal number designating the number of pages.

```
SYSCKP=     ( strtcyl   ┌,cylcount ┐ )
       [(]  {           │,1        │ }        [)]
            ( strtpage  │,pagecount│ )
                        └,50       ┘
```

is the number of the real starting cylinder (strtcyl), or the starting page number for FB-512 (strtpage), of the checkpoint area. 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 spool file IDs is the lesser of either the 9900 system limit or the number of checkpoint slots allocated. The strtcyl is a one to three-digit decimal number designating the first real cylinder where CP checkpoint start information is to be saved.

The cylcount value 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 spool file IDs.

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
3350	1 cylinder	4 cylinders
3375	1 cylinder	5 cylinders
3380	1 cylinder	3 cylinders
FB-512	50 pages	298 pages

Use the following formulas 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	$N = [2+(NSP/32)+(NRS/34)]$

Figure 23. Dynamic Checkpoint Start Area Calculations

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.

```
SYSWRM=      ( strtcyl    ┌,cylcount ┐ )
       [(]  <             │,1        │  >       [)]
             ( strtpage   │,pagecount│ )
                          └,50       ┘
```

is the number of the real starting cylinder (strtcyl), or the starting page number for FB-512 (strtpage), to contain the warm start data. 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).

The strtcyl is a one to three-digit decimal number designating the first real cylinder where CP warm start information is to be saved.

The cylcount value 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/strtpage 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 following formulas 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 with:

- A maximum of 32 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

$$N = \frac{[59 + 32/32 + 128/128 + 200/50]}{57} = \frac{65}{57} = 1$$

The formula for each device type is:

Device Type	**Formula**
2314/2319	$N = \frac{[34+(NSF/32)+(NCS/128)+((NAUx4)/50)]}{32}$
3340/2305	$N = \frac{[26+(NSF/32)+(NCS/128)+((NAUx4)/50)]}{24}$
3330	$N = \frac{[59+(NSF/32)+(NCS/128)+((NAUx4)/50)]}{57}$
3350	$N = \frac{[122+(NSF/32)+(NCS/128)+((NAUx4)/50)]}{120}$
3375	$N = \frac{[98+(NSF/32)+(NCS/128)+((NAUx4)/150)]}{96}$
3380	$N = \frac{[152+(NSF/32)+(NCS/128)+((NAUx4)/50)]}{150}$
FB-512	$N = [2+(NSF/32)+(NPS/85)+((NAUx4)/50)]$

Figure 24. Warm Start Area Calculations

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. There are 128 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.

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 123. The VM/SP nucleus resides at cylinder 882 and the warm start storage area is cylinders 877 and 878. The error recording area starts at cylinder 879 and the checkpoint start storage area is cylinder 442. The format of the SYSRES macro is:

```
SYSRES  SYSVOL=VMSRES,SYSRES=123,SYSTYPE=3380,SYSNUC=882,SYSCLR=,    X
        SYSWRM=(877,2),SYSERR=(879,2),SYSCKP=(442,1)
```

SYSOPR Macro

Use the SYSOPR macro instruction to specify the system operator's userid, and the userid of the operator who is to receive VM/SP 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 (a string of up to eight characters) 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 plan to use IPCS, allow this operand to default to OPERATNS or specify the IPCS userid.

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
```

SYSCOR Macro

Use the SYSCOR macro instruction to generate the internal control block called the CORTABLE. The AP and MP operands specify whether VM/SP 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} [,FREE=ffff] [,AP={YES NO} ,MP={YES NO}] [TRACE=nnn]

where:

RMSIZE={xxxxxK
yyM}

is the amount of real storage available for VM/SP. This value limits the amount of real storage used by VM/SP if 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 is initialized, a message indicating the amount of storage available is displayed at the operator's console.

The value, xxxxx, is a three to five-digit number that designates the amount of real storage in terms of K bytes, where 1K=1024 bytes. This value may range from 384K to 16384K. It must always be a multiple of 2.

The value, yy, is a one or two-digit number that designates the amount of storage in terms of M bytes, where 1M=1024K bytes. This value may range from 1M to 16M.

Note: Do not specify a value much larger than the size of real storage, because the generated core table uses a large amount of real storage.

FREE=ffff

is a one to four-digit number that specifies the number of fixed free storage pages to be allocated at VM/SP initialization. This number must be greater than three. The amount of storage represented must not be greater than ether 25% of the value specified for RMSIZE, or RMSIZE less the V=R area if generated.

The recommended initial 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, VM/SP allocates three pages for the first 256K of real storage and one page for each additional 64K thereafter not including the V=R size, if any. In AP and MP modes, the default is increased by 25%.

One method of determining if you need to increase 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.

$\texttt{AP=}\left\{\begin{array}{l}\texttt{YES}\\ \underline{\texttt{NO}}\end{array}\right\}$

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.

$\texttt{MP=}\left\{\begin{array}{l}\texttt{YES}\\ \underline{\texttt{NO}}\end{array}\right\}$

YES specifies that initialization processing occurs in multiprocessor mode.

NO specifies that initialization processing occurs 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 run 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), the default size will be used for the trace table.

Examples:

The first example defines real storage as 256K (262,144 bytes) and the second example defines real storage as 1M (1,048,576 bytes).

```
SYSCOR   RMSIZE=256K
SYSCOR   RMSIZE=1M
```

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 difference 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 difference 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 difference 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)
```

SYSMON Macro

The SYSMON macro is used to start daily automatic performance data collection with the VM Monitor. The IBM Field Developed Program Virtual Machine Facility/370: Performance/Monitor Analysis 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 default ,BUFFS=n

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 system directory entry may be specified.

```
CLASS=M
CLASS=class
```

specifies the spool file class to be generated to contain monitor data. Any class (A through Z and 0 through 9) may be used but the default M is preferred since the VMAP 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 you to make a specific change to cause automatic monitoring. All other parameters may be system default values, giving positive and useful monitoring results.

```
ENABLE=(PERFORM,USER,DASTAP)
ENABLE=(classa,classb,classc,...)
```

specifies any combination of monitor classes of data collection. It is assumed that the system analyst understands the use of various classes, overhead resulting from data collection, and relative magnitude of the corresponding data reduction. The default specifies sampled data classes only and is considered the least that can be specified for useful data reduction. The default classes are sufficient for analysis of a system's load and performance profile with a view to diagnosis of possible bottlenecks and for establishing long term growth patterns.

```
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) occurs 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: This 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, and whether or not monitoring should be stopped 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 you 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 your 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 default
BUFFS=n
```

specifies the number of data collection buffers needed by the monitor to avoid suspension incidents. Data collection suspension occurs when output to tape or spool files cannot keep ahead of data collection, and an overrun condition occurs. By increasing the number of monitor buffers, suspension incidents can be reduced or eliminated. The default depends on the processor on which the system is running. (See the *VM/SP System Programmer's Guide* for a description of the MONITOR command.) If not satisfied with the defaults, you may specify any number of buffers from 1 to 10.

Example:

```
SYSMON USERID=ANALYST,AUTO=YES,ENABLE=(PERFORM),
       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 journaling 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 ,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 journaling 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 journaling 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.

```
,LOGLMT=(2,3,4)
,LOGLMT=(x,y,z)
```

is the invalid LOGON/AUTOLOG password threshold specification. The value specified applies to a single userid for a single LOGON session. x is the value which, when reached or exceeded, causes a type 04 accounting record to be generated for that and each following 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 following 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 following 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 following 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 current fixed limit of 10 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 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.

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 can be specified. 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 your virtual machine, they should be processed periodically to avoid accumulating and tying up system resources.

Note: The accounting records can be moved to tape via the SPTAPE command for later processing.

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.

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.

You 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 alphameric.

`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. 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'),
         (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 things to consider about 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 keeps this from happening.

SYSID Macro

The SYSID macro statement allows you to identify a system with an eight-character identification, which is printed on the output separator page.

The processor ID is specified with a model number and a serial number. If you have more than one VM/SP processor, a list of system ID's with corresponding processor model and serial numbers can be specified. When VM/SP 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-to four-digit number that specifies the processor model number (for example 158, 3031, or 4341) for a multiprocessor system.

`serial`
is a five- or six-digit number that specifies the processor serial number corresponding with the system ID. An AP user should use the serial number and model number of the main (IPL) processor in the SYSID statement. For an MP user, 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 user.

   ```
   SYSID DEFAULT=CUSTOMER
   ```

   Gives this VM/SP system the ID "CUSTOMER." This ID will be printed on the separator page.

3. Multiple processor user.

```
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.

SYSORD Macro

Use the SYSORD macro statement to specify the order in which preferred and/or nonpreferred paging and spooling devices are searched for available pages to satisfy paging/spooling operations. Different search orders may be specified for preferred and nonpreferred devices. SYSORD is a required macro statement. It must be included in DMKSYS prior to the SYSLOCS macro statement.

The format of the SYSORD macro is:

Name	Operation	Operands
	SYSORD	[SYSFH= (devtype[,devtype,...[,(devtype,devtype,...)]])] [SYSMH= (devtype[,devtype,...[,(devtype,devtype,...)]])] [SYSTEMP=(devtype[,devtype,...[,(devtype,devtype,...)]])]

where:

devtype
is a supported DASD type

SYSFH=
designates the priority order for searching device types that have preferred fixed-head paging (PAGE) cylinders or extents defined. A specified device type indicates a single device or several devices of the same device type in a chain. Supported fixed-head device types (in order of decreasing performance) are: 2305, 3350, 3330, and 3340. A device type may appear only once within this operand.

Note: The 3330 specification for SYSFH is included only for fixed-head 3350s running in 3330 compatibility mode.

SYSMH=
designates the priority order for searching device types that have preferred moveable-head paging (PAGE) cylinders or extents defined. A specified device type indicates a single moveable-head device or several moveable-head devices of the same device type in a chain. Supported moveable-head device types (in order of decreasing performance) are: 3380, 3375, 3370, 3310, 3350, 3330, 3340, and 2314. A device type may appear only once within this operand.

SYSTEMP=
designates the priority order for searching device types that have nonpreferred (TEMP) cylinders or extents defined. TEMP space is used for overflow paging operations and all spooling operations. A specified device type indicates a single fixed or moveable-head device or several devices of the same device type in a chain. Supported device types (in order of decreasing performance) are: 2305, 3380, 3375, 3370, 3310, 3350, 3330, 3340, and 2314. A device type may appear only once within this operand.

The brackets in the format of the SYSORD MACRO represent options in specifying the operands. For *each* operand, the three options indicated are:

1. The option to specify the operand. If specified, at least one device type is required.

   Example:

   ```
   SYSFH=(devtype)
   SYSMH=(devtype)
   SYSTEMP=(devtype)
   ```

2. The option to specify more than one device type.

   Example:

   ```
   SYSFH=(devtype,devtype,...)
   SYSMH=(devtype,devtype,...)
   SYSTEMP=(devtype,devtype,...)
   ```

3. The option to specify for devtype, a list of device types indicating that the device types in the inner parentheses are to have equal priority with each other.

   Example:

   ```
   SYSFH=(devtype,(devtype,devtype,...),...)
   SYSMH=(devtype,(devtype,devtype,...),...)
   SYSTEMP=(devtype,(devtype,devtype,...),...)
   ```

A request for a page of external page space is satisfied by allocating space on a preferred paging cylinder or extent, provided that one exists on the system and that it has a page slot available. You specify preferred and nonpreferred paging space by using the PAGE and TEMP operands of the CP Format/Allocate service program.

Within the class of CP-owned volumes with preferred space, pages are first allocated from volumes with fixed-head cylinders or extents in the order specified by the SYSFH operand. (If SYSFH is not specified, the default order is used.) If no fixed-head space is available, then pages are allocated from volumes with moveable-head cylinders or extents in the order specified by the SYSMH operand. (If SYSMH is not specified, the default order is used.) Finally, if no preferred space is available, pages are allocated from volumes with nonpreferred space in the order specified by the SYSTEMP operand. (If SYSTEMP is not is specified, the default order is used.)

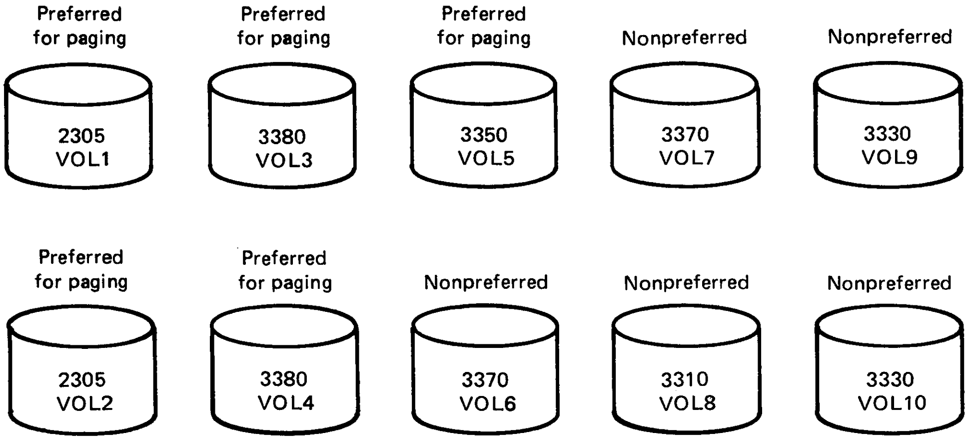
Spooling space is not allocated from preferred space. Spooling space is allocated from volumes with nonpreferred space in the order specified by the SYSTEMP operand. (If SYSTEMP is not specified, the default order is used.)

A rotary paging scheme is used to evenly distribute external DASD pages across all volumes of a specific device type and to allow concurrent paging operations. For *each* SYSFH/SYSMH/SYSTEMP operand, a device type order specification of (devtype,devtype,devtype,...) indicates a series of unique device types, where each device type may indicate a single device or a chain of devices of the same type. If a chain exists, pages are allocated on all devices within the chain on a rotating basis. When all available pages on the device chain are used, the next specified device type is examined for available pages. Allocation of DASD pages occurs in this manner until all device types specified within the parentheses are used. A device type order specification of (devtype,devtype,(devtype,devtype,...),...) indicates that the device types in the inner parentheses are to have equal priority with each other.

The same rotary scheme is used for the allocation of spooling space from nonpreferred devices.

The following example shows the sequence in which paging or spooling space would be allocated from all CP-owned volumes having cylinders or extents defined as specified in the SYSORD macro below. The example is meant to illustrate the allocation technique, not necessarily the desirable allocation of paging or spooling space.

```
SYSORD SYSMH=(3380,3350) SYSTEMP=((3370,3310),3330)
```



Using the previous example, paging cylinders or extents would be allocated in the following sequence:

1. On an alternating basis on VOL1 and VOL2 until all fixed-head PAGE space is used on these volumes.

2. On VOL5 until all fixed-head PAGE space is used on that volume.

3. On an alternating basis on VOL3 and VOL4 until all moveable-head PAGE space is used on these volumes.

4. On VOL5 until all moveable-head PAGE space is used on that volume.

5. On an alternating basis on VOL6, VOL7, and VOL8 until all TEMP space is used on these volumes.

6. On an alternating basis on VOL9 and VOL10 until all TEMP space is used on these volumes.

Spooling cylinders or extents would be allocated in the following sequence:

1. On an alternating basis on VOL6, VOL7, and VOL8 until all TEMP space is used on these volumes.

2. On an alternating basis on VOL9 and VOL10 until all TEMP space is used on these volumes.

SYSORD is a required macro statement. Though the statement is required, the operands are optional. If SYSORD is specified without operands, the following default priority orders apply:

```
SYSFH=  (2305, 3350, 3330, 3340)
SYSMH=  (3380, 3375, 3370, 3310, 3350, 3330, 3340, 2314)
SYSTEMP=(2305, 3380, 3375, 3370, 3310, 3350, 3330, 3340, 2314)
```

A device type may appear only once within each SYSFH/SYSMH/SYSTEMP statement. If duplicate device types are specified within an operand, the following MNOTE is issued:

```
   'SAME DEVICE TYPE SPECIFIED MULTIPLE TIMES'
```

You should eliminate the multiple specification and reassemble DMKSYS.

Specifying an operand other than SYSFH/SYSMH/SYSTEMP produces the following:

```
   'KEYWORD PARAMETER 'parameter' UNDEFINED IN MACRO DEFINITION'
```

You have specified an invalid keyword. Only SYSFH, SYSMH, or SYSTEMP are allowed.

Specifying an invalid device type within any operand generates one of these MNOTES:

```
   'INVALID DEVICE TYPE FOR SYSFH'
   'INVALID DEVICE TYPE FOR SYSMH'
   'INVALID DEVICE TYPE FOR SYSTEMP'
```

Missing parentheses or parentheses without following operands generate the MNOTE:

```
   'POSITIONAL OR EMPTY PARMS NOT ALLOWED'
```

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 is:

Name	Operation	Operands
	SYSMIH	$\left[\underline{\text{DASD}}\text{=}\begin{Bmatrix}\underline{\text{0:15}}\\ \text{mm:ss}\end{Bmatrix}\right]\left[\text{,}\underline{\text{GRAF}}\text{=}\begin{Bmatrix}\underline{\text{0:30}}\\ \text{mm:ss}\end{Bmatrix}\right]$ $\left[\text{,}\underline{\text{TAPE}}\text{=}\begin{Bmatrix}\underline{\text{10:00}}\\ \text{mm:ss}\end{Bmatrix}\right]\left[\text{,}\underline{\text{UR}}\text{ =}\begin{Bmatrix}\underline{\text{1:00}}\\ \text{mm:ss}\end{Bmatrix}\right]$ $\left[\text{,}\underline{\text{MISC}}\text{=}\begin{Bmatrix}\underline{\text{12:00}}\\ \text{mm:ss}\end{Bmatrix}\right]$

where:

$\underline{\text{DASD}}\text{=}\begin{Bmatrix}\underline{\text{0:15}}\\ \text{mm:ss}\end{Bmatrix}$

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.

$\underline{\text{GRAF}}\text{=}\begin{Bmatrix}\underline{\text{0:30}}\\ \text{mm:ss}\end{Bmatrix}$

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.

$\underline{\text{TAPE}}\text{=}\begin{Bmatrix}\underline{\text{10:00}}\\ \text{mm:ss}\end{Bmatrix}$

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, 10 minutes.

$\underline{\text{UR}}\text{=}\begin{Bmatrix}\underline{\text{1:00}}\\ \text{mm:ss}\end{Bmatrix}$

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, 1 minute.

$$\underline{\texttt{MISC}}=\left\{\begin{matrix}\underline{\texttt{12:00}}\\ \texttt{mm:ss}\end{matrix}\right\}$$

specifies the time interval value used for miscellaneous devices. Miscellaneous devices include MSS devices, CLASGRAF TYP328x 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 (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, 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 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 processing for devices that are functioning properly.

5. If you specify more that 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. If a 3800 is installed, the time interval for unit record devices should be increased to 5 minutes because of the warm-up time required by the printer.

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 processing
- Disable I/O monitoring for Miscellaneous devices

```
SYSMIH GRAF=00:20,TAPE=00:00,MISC=00:00
```

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 file.

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	

Chapter 22. Additional System Definition Files

The Forms Control Buffer Load (DMKFCB)

The DMKFCB module is supplied with the product tape. This module defines a 3211, 3203-4, 3203-5, 3262-1, 3262-5, 3262-11, 3289 Model 4, or 4245 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 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 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 *VM/SP System Programmer's Guide* for directions.

The Universal Character Set and Font Offset Buffer

The DMKUCS, DMKUCB, DMKUCC, DMKPIA, and DMKPIB modules are supplied with the product tape. 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 for a particular device, see the corresponding module's prologue.

Appendixes

Appendix A. Licensed Programs and Integrated Emulators

The Conversational Monitor System (CMS), and the Control Program (CP) are distributed as components of VM/SP. Certain other facilities mentioned in this publication are not part of VM/SP, 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.

VM/370 Assembler

The VM/370 Assembler is distributed as a part of the VM/SP system and is required for installation and further support of the system. All necessary installation and support macros are provided in CMS libraries.

Licensed Programs

The following is a list of IBM Licensed Programs and their respective program numbers that VM/SP users have found useful.

Program Products are listed first in each area, followed by Program Offerings (Field Developed Programs(FDPs) and Installed User Programs(IUPs)), and Programming Requests for Price Quotation(PRPQs).

More information may be found in the IBM DPD Software Directory, form number GB21-9949.

General Business Applications:

Interactive Instructional Authoring System (IIAS)	5668-001
Interactive Instructional Presentation System (IIPS)	5668-012
General Purpose Simulation System V OS (GPSS-V)	5734-XS2
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
Applicant Tracking System	5796-PLT
APL Data Interface - Version II (APL/DI-II)	5796-PNG
Alpha Search Inquiry System Online Update	5798-CFJ
Braille Utilities	5798-CRZ
Financial Planning System II (FPS-II)	5798-DCN

Scientific and Engineering Applications:

MATH/BASIC	7534-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 Forcasting 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
List Processor 370 (LISP/370)	5796-PKL

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:

System Productivity Facility (SPF)	5668-009
COBOL Interactive Debug	5734-CB4
Interactive Productivity Facility (IPF)	5748-MS1
Display Management System/370 (DMS/370)	5748-XC3
Development Management System/DPCX (DMS/DPCX)	5748-XC4
Display Management System/CMS (DMS/CMS)	5748-XXB
Entry Level Interactive Application System I (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/Interactive Problem Control System (IPCS)	5748-SA1

Automated Project Planning and Evaluation (APPLES)	5796-AZR
VS Memory Analysis (VS/REPCAK)	5796-PDZ
VM/CMS Performance Monitor Analysis (VMAP)	5798-CPX

Auditor and Security Support Aids:

Display Management System/370 (DMS/370)	5748-XC3
VM Directory Maintenance (DIRMAINT)	5748-XE4
Display Management System/CMS (DMS/CMS)	5748-XXB
VM Interactive File Sharing (VM/IFS)	5748-XXC

JES2 Informational Retrieval System for CMS	5796-AYD
APL Statistical Library for VSAPL	5796-PHW
APL Decision Table Processor and Code Generator	5796-PJB
VSAPL Advanced Statistics Library	5796-PJT
A Departmental Reporting System II (ADRS-II)	5796-PLN
Source Compare Audit Utility (SUPER-C)	5796-PLZ
APL Data Interface II (APL/DI-II)	5796-PNG

Text and Office Systems Applications:

Graphics Attachment Support	5799-AXX
Document Composition Facility (DCF)	5748-XX9

Storage and Informational Retrieval System (STAIRS/CMS)	5785-CAH
Document Composition Facility for 6670	5798-DBR
Professional Office System (PROFS)	5799-BEX

Graphics Applications:

Digital Interactive Graphics Interpretive Mapping	5668-959
GAM/SP	5668-978
Graphic 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 Business 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
McGill University System for Interactive Computing (MUSIC)	5796-ATL
Applicant Tracking System	5796-PLT

APL Support and Applications:

A Programming Language (VSAPL)	5748-AP1
APL Continuous System Modeling (CSMP)	5785-KAE
APL System Extensions	5796-AZT
APL Forcasting and Time Series Analysis	5796-PFX
APL Function Editor for VSAPL	5796-PGY
APL Statistical Library	5796-PHW
APL Decision Table Processor	5796-PJB
General Purpose Simulation System VSAPL (GPSS-APL)	5796-PJG
VSAPL Advanced Statistics Library (STATLIB2)	5796-PJT
APL Interactive Training Course	5796-PJW
General Cross Assembler Generator	5796-PKD
APL/DI File Create using PL/I	5796-PKP
IMS APL Data Link for VM/CMS	5796-PLE
A Departmental Reporting System II (ADRS-II)	5796-PLN
APL Handbook of Technical Workspace	5796-PLP
Interactive Circuit Design	5796-PLR
APL Multivariate Time Series Analysis	5796-PLX
Extended Editor and Full Screen Manager	5796-PLY
APL Workspace Structure Analyzer	5796-PNB
APL Data Interface Version II (APLDI-II)	5796-PNG
APL Data Language	5798-CHR

3277 Graphics Attachment Plotter Tablet	5798-DCE
Financial Planning System II (FPS-II)	5798-DCN
VSAPL Variable Conversion Processor	5798-DEH
3277 APL Graphics Attachment Support	5799-AXW

Query Programs and Support:

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 COBOL Compiler and Library	5736-CB1
DOS RPG II	5746-RG1
DOS/VS Sort/Merge Version 2	5746-SM2

OS/VS SCP, Compilers, Utilities, and Aids:

OS Assembler H	5734-AS1
COBOL Interactive Debug	5734-CB4
OS FORTRAN IV G	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 Library	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
FORTRAN Utilities for VM/370	5798-DFH

MVS SCP, Compilers, Utilities, and Aids:

FORTRAN Interactive Debug	5734-FO5
VS Memory Anaylsis (VS)	5796-PDZ

VM SCP, Aids, Utilities:

System Productivity Facility	5668-009
GAM/SP	5668-978
FORTRAN Interactive Debug	5734-FO5
VM/VTAM Communication Network Application (VM/VCNA)	5734-RC5
EP/VS for OS/VS and VM/370	5744-AN1
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/Interactive Problem Control System (VM/IPCS)	5748-SA1
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 (ELIAS-I/VM)	5748-XXK
Document Composition Facility (SCRIPT/VS)	5748-XX9
VM+DOS/VSE System IPO/E DB/DC (SIPO/E)	5750-AAE
VM+DOS/VSE System IPO/E DC (SIPO/E)	5750-AAF
VM+DOS/VSE System IPO/E Batch/Interactive (SIPO/E)	5750-AAJ
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
APL System Extensions	5796-AZT
Assembler H CMS Interface	5796-PEJ
FORTRAN Conversion Aid	5796-PFG
Batch Monitor for VM/CMS	5796-PGZ
List Processor/370 (LISP/370)	5796-PKL
Query by Example (QBE)	5796-PKT
IMS APL Data Link for VM/CMS	5796-PLE
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
VM Diskette Copy	5796-PNT
Virtual Librarian	5796-PNZ
CMS Sort for VM/CMS plus extensions	5798-BDW
VM Performance Monitor Analysis (VMAP)	5798-CPX
VM/CMS Library and SPMOL-II Simulator	5798-CYA
Application Enabling Facility	5798-DBF
Airline Control Program Testing in VM	5798-DEP
Professional Office System (PROFS)	5799-BEX

Integrated Emulators

Emulator-dependent programs (except for DOS emulation under OS or OS/VS) that run on a particular processor equipped with the appropriate compatibility features can run on that processor in DOS or OS virtual machines under VM/SP.

Figure 25 shows, by processor model number, which integrated emulators can run under VM/SP and the compatibility feature numbers (#xxxx) that are required.

No changes are required to the emulators, to DOS or OS, or to VM/SP to allow emulator-dependent programs to run 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.

Processor Model	S/360 Model 20	1401 1440 1460	1401 1440 1460 1410 7010	7070 7074	7080	709 7090 7094 7094II
135,135-3,138	#7520	#4457				
145,145-3,148		#4457	#4458			
155 II,158			#3950	#7117		
165 II				#7117	#7118	#7119
168				#7127	#7128	#7129
4331		#3950				

Figure 25. Integrated Emulators that Run under VM/SP

Appendix B. Configuration Aid

Appendix B lists the devices and control units that can be specified in a VM/SP system generation, grouped by use. The maximum number of devices that can be specified in the FEATURE= operand of the RCTLUNIT macro is listed, along with whether or not the control units can operate on a shared subchannel.

Listed are the devices that can be attached to each control unit, and the operands that can be specified for each device in the RDEVICE macro.

The control units and devices 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		Shared Sub-channel	RDEVICE	
	CUTYPE=	Maximum FEATURE=		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

Type of Device	RCTLUNIT		Shared Sub-channel	RDEVICE	
	CUTYPE=	Maximum FEATURE=		DEVTYPE=	Other Operands
Trans-mission Control Units (cont.)	3704 3705	16-DEVICE 256-DEVICE	— —	3704 3705	ADAPTER=BSCA, IBM1, TELE2, TYPE1, TYPE2, TYPE3, or TYPE4 MODEL=A1 through H8 SETADDR=0, 1, 2, or 3 CPTYPE=EP CPNAME=ncpname BASEADD=cuu
	ICA	16-DEVICE	—	ICA	ADAPTER=BSCA, IBM1, TELE2, or SDLC
	2955	—	—	2955	—
Display Devices (Local Attach.)	2848	32-DEVICE	yes	2260 1052	—
	2845	—	yes	2265	—
	2840	—	—	2250	—
	3272[1]	32-DEVICE	yes	3277 3284 3286 3288	FEATURE=OPRDR
	3274	32-DEVICE	yes	3277 3278 3279 3284 3286 3287 3288 3289 4250	FEATURE=OPRDR MODEL=2, 3, 4, or 5 MODEL=2, or 3
	DPA	—	—	3230 3268 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
	ICA	—	—	ICA	ADDRESS=cuu (line address) ADAPTER=BSCA CLUSTER=label

[1] If a 3287 is attached to a 3272, the 3287 is specified as a 3284 or 3286.

Type of Device	RCTLUNIT		Shared Sub-channel	RDEVICE	
	CUTYPE=	Maximum FEATURE=		DEVTYPE=	Other Operands
Remote 3270 Display Devices (cont.)	3704 3705	— —	— —	3704 3705	ADDRESS=cuu (line address) ADAPTER=BSCA CPTYPE=EP BASEADD=cuu CLUSTER=label
Direct Access Storage Devices	2841	—	yes	2311 2321 2303	
	2314	—	yes	2314	
	2319	—	—	2319	
	IFA	16-DEVICE	—		
	3830	32-DEVICE	—	3330	MODEL=1, 2, or 11
	3830	160-DEVICE	—	3330	FEATURE=SYSVIRT, FEATURE=VIRTUAL
	3880	32-DEVICE	—	3330	MODEL=1, 2, or 11
	3345	16-DEVICE	—	3333	MODEL=1 or 11
	3880	16-DEVICE	—	3333	MODEL=1 or 11
	ISC	64-DEVICE	—		
	3830	64-DEVICE	—	3340	
	3880	64-DEVICE	—	3340	
	3345	16-DEVICE	—		
	ISC	160-DEVICE	—		
	IFA	160-DEVICE	—		
	3830	64-DEVICE	—	3350	FEATURE=FH
	3880	64-DEVICE	—	3350	FEATURE=FH
	ISC	64-DEVICE	—		
	IFA	16-DEVICE	—		
	3880	64-DEVICE	—	3375	
	3880	64-DEVICE	—	3380	
	3880	16-DEVICE	—	FB-512	
		32-DEVICE	—		
		64-DEVICE	—		
				3310	
				3370	
	2820	—	yes	2301	
	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 FEATURE=7-TRACK, DUALDENS
	2804	16-DEVICE			
				2402	MODEL=1, 2, 3, 4, 5 or 6 FEATURE=7-TRACK, CONV, DUALDENS
				2420	MODEL=5 or 7

Type of Device	RCTLUNIT		Shared Sub-channel	RDEVICE	
	CUTYPE=	Maximum FEATURE=		DEVTYPE=	Other Operands
Tape Devices (cont.)	3411	—	yes	3410 3411	MODEL=1, 2, or 3 FEATURE=7-TRACK, DUALDENS
	2403 2404	16-DEVICE	yes	2403 2404	MODEL=1, 2, 3, 4, 5 or 6 FEATURE=7-TRACK, CONV, DUALDENS
	2415	—	yes	2415	MODEL=1, 2, 3, 4, 5 or 6 FEATURE=7-TRACK, CONV
	FTA	16-DEVICE	—	8809	
	3411	—	yes	3410 3411	MODEL=1, 2, or 3 FEATURE=7-TRACK, DUALDENS
	3430	—	yes	3430	FEATURE=DUALDENS
	3803	16-DEVICE[1]	yes	3420	MODEL=3, 4, 5, 6, 7 or 8 FEATURE=7-TRACK, DUALDENS
Unit Record Output Devices	2821	—	—	1403	CLASS=(class[,class...]) FEATURE=UNVCHSET
				2540P	CLASS=(class[,class...])
	1443	—	—	1443	CLASS=(class[,class...])
	3811	—	—	3211	CLASS=(class[,class...])
	3262	—	—	3262[2]	MODEL=5 CLASS=(class[,class...])
	2826	—	—	1018	
	2520	—	—	2520P	CLASS=(class,[class...])
	SVPC	—	—	3262	MODEL=1 or 11
	SVPC	—	—	3289	MODEL=4
	3203	—	—	3203	MODEL=4 or 5 CLASS=(class,[class...])
	3505	—	—	3525	CLASS=(class,[class...])
	3800	—	—	3800	MODEL=3 or 8 FEATURE=4WCGMS, IMAGE=imagelib, CHARS=ffff,FCB=lpi, DPMSIZE=n CLASS=(class[,class...])
	4245	—	—	4245	MODEL=1 CLASS=(class[,class...])

[1] 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.

[2] The RCTLUNIT, when coded as 3262, is valid for DEVTYPE=3262, MODEL=5.

Type of Device	RCTLUNIT		Shared Subchannel	RDEVICE	
	CUTYPE=	Maximum FEATURE=		DEVTYPE=	Other Operands
Unit Record Input Devices	2821	—	—	2540R	
	2520	—	—	2520R	
	3505	—	—	3505	
	2495	—	—	2495	
	2822	—	—	2671	
	2826	—	—	1017	
	2501	—	—	2501	
Special Devices	CTCA	—	yes	CTCA	
	3088	64-DEVICE	—	3088	
	7443	—	—	7443	

Appendix C. Compatible Devices

The devices listed below are functionally equivalent to the 2770 Communication System. Details on the feature requirements for operational control of such devices are not contained in VM/SP publications, but in the programming and operating publications that support these devices.

IBM 6640 Document Printer-Communicating

- 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 - Communicating and IBM 6240 Mag Card Typewriter - Communicating

- 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 Restrictions

A virtual machine created by VM/SP is capable of running an IBM System/360 or System/370 operating system as long as certain VM/SP restrictions are not violated. Virtual machine restrictions and certain execution characteristics are stated in this appendix.

VM/SP

Two components, CP and CMS, have been extensively modified and integrated into a VM/370 Release 6 base. This collective package (CP and CMS) is referred to as VM/SP. However, there are recommended program products (Remote Spooling Communication Subsystem (RSCS) Networking, program number 5748-XP1) and Interactive Problem Control System (IPCS) Extension, program number 5748-SA1) available that have been technically advanced to provide supportive function to VM/SP.

Restrictions - Channel Program

Looping channel programs should be avoided. Execution of a backward transfer in channel CCW to an I/O CCW that will present channel end and device end at the same time could 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 state, the CP HALT command must be issued to the device or have the operator issue a system reset.

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 Programs

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 input/output 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/SP option
- Those containing polling sequences

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 VM/SP directory entry. There is no such restriction for DOS

ISAM, or for ISAM if it is running in an OS/VS virtual=virtual partition. If ISAM is to run in an OS/VS virtual=real partition, you must specify the ISAM option in the VM/SP directory entry for the OS/VS virtual machine.

Virtual machines using OS/VS TCAM (Level 5, generated or invoked with the VM/SP option) issue a DIAGNOSE instruction when the channel program is modified. This instruction causes CP to reflect the change 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.

When a virtual machine starts a channel program containing a polling sequence, the CCW translation sets a 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 restriction against dynamically modified channel programs does not apply if the virtual machine has the virtual=real performance option and the NOTRANS option has been set on.

Minidisk Restrictions

The following restrictions exist for minidisks:

1. In the case of read home address with the skip bit off, VM/SP 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 FB-512 device with the skip bit off, VM/SP 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: The user 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, further 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 a VS1 system using VM handshaking does no paging, any ISAM programs run under VS1 are treated by VM/SP as though they are running in an ADDRSPC=REAL partition.

5. VM/SP 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 your channel program for a count-key-data minidisk does not perform a seek operation, VM/SP inserts a positioning seek operation into the program to prevent accidental accessing. Thus, certain channel programs may generate a condition code (CC) of 0 on a SIO instead of an expected CC of 1, which is reflected to the virtual machine. The final status is reflected to the virtual machine as an interrupt.

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 non-zero relocation factors if the channel program includes multiple-track operations and depends on a search ID high or a search ID equal or high to end the program. This is because the record 0 count fields on these 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 end prematurely if a real record 0 is encountered.

Notes:

a. Minidisks with non-zero relocation factors on 3330, 3340, 3350, 3375, or 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 high CCW to find the end of the VTOC.

b. If the 'R' byte field of 'CCHHR' = 0 at the time 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. If the DASD channel programs directed to 3330/3340/3350/3375/3380 devices include a write record R(0), results differ depending on whether the 3330/3340/3350/3375/3380 is dedicated or nondedicated. A full-pack minidisk is treated the same as any nondedicated device. For a dedicated 3330/3340/3350/3375/3380, a write R(0) is allowed, but you must be aware that the track descriptor record may not be the same from one 3330/3340/3350/3375/3380 to another. For a nondedicated 3330/3340/3350/3375/3380, 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.

9. When performing DASD I/O, if the record field of a search ID argument is zero when a virtual Start I/O is issued, but the search ID argument is dynamically read by the channel program before the search ID CCW is executed, then the real search ID uses the relocated search argument instead of the argument that was read dynamically. To avoid this problem, the record field of a search ID argument 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 wanted.

10. On FB-512 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.

11. FB-512 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.

12. DIAGNOSTIC READ HOME ADDRESS and DIAGNOSTIC WRITE HOME ADDRESS commands are supported only for:

    - Dedicated devices
    - Minidisks that start at cylinder 0 (real)

13. Refer to *Device Support Facilities,* GC35-0033, for procedures on formatting 3375 and 3380 direct access storage for use in an OS/VS operating system running in a virtual machine.

14. 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 referenced differently by 3330 Models 1 and 11, the results will be unexpected.

Timing Dependencies

Timing dependencies in input/output devices or programming do not function normally under VM/SP:

1. The following telecommunication access methods (or the designated option) violate the restriction on timing dependency by using program-controlled interrupt techniques and/or violate 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 TCAM is not generated or invoked with the VM/SP 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 dialed 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 if issued from third level storage. For example, a reset poll macro has a NOP effect if executed from virtual=virtual storage in a VS1 system that is running under VM/SP.

2. Programming that makes use of the PCI channel interrupt for channel program modification or processor signalling must be written so that processing can continue normally if the PCI 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 interrupt within a fixed period of time may not function correctly because of execution delays caused by normal VM/SP 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 timing dependent. For this reason, the channel appears available to the virtual machine operating system, and channel available interrupts are not observed. However, operations on virtual block multiplexing devices should use the available features like Rotational Position Sensing to heighten use of the real channels.

5. Devices that experience extreme performance penalties if not reinstructed within a fixed interval may experience this penalty during every I/O operation. An example is the 8809 tape drive. Setting the mode to "streaming" may actually result in a slower data rate than running in nonstreaming mode. Execution delays, caused by normal VM/SP processing, prevent a timely reinstruct and the 8809 tape drive may sustain a 1.2 second delay on every I/O operation. You must decide (based mainly on the size of the I/O buffers) between running at 100 IPS with continuous delays and running at 12.5 IPS, and set the mode accordingly.

Processor Model-Dependent Functions

On the System/370 Model 158 only, the virtual machine assist feature cannot operate at the same time with the 7070/7074 compatibility feature (#7117).

Programs written for processor model-dependent functions may not run properly in the virtual machine under VM/SP. The following points should be noted:

1. Programs written to examine the machine logout area do not have meaningful data since VM/SP does not reflect the machine logout data to a virtual machine.

2. Programs written to obtain processor identification (via the Store CPUID instruction, STIDP) receive the real machine 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.

4. Since an operating system's channel error recovery procedures may be processor model- and channel model-dependent, operating systems that will run in a virtual machine may have to be generated for the same model of processor that VM/SP will be running on.

Channel Model-Dependent Functions

Channel checks (channel data check, channel control check and interface control check) no longer cause the virtual machine to be reset. They are reflected to the virtual machine as other I/O errors are. 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 abide by 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 may be inconsistent with the data provided by the store channel id (STIDC) instruction.

Note: Virtual machines running CMS do not need to abide by 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 a random channel, one of several the specified virtual channel may map to. 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 model-dependent error recovery procedures can use STIDC to identify the channel.

Channel extended logouts are reflected to the virtual machine in a manner that is processor model- and channel model-dependent and constant with the data returned by STIDC (provided that the virtual-to-real channel mapping abides by the requirement stated previously).

A difference in the handling of channel extended logouts occurs if the virtual machine uses the bit in control register 14 to mask out channel extended logouts. In a virtual machine, any channel extended logouts that are masked out by control register 14 are lost rather than kept pending, and the logout pending bit (bit 5) in the CSW is never set. However, channel extended logouts will not be lost when they are kept pending along with their associated I/O interrupts by the channel masks in control register 2 and the PSW. Regardless of whether or not the setting of the virtual machine's control register 14 causes it to lose the channel extended logout, CP will still successfully record the logout in its own error recording cylinders.

Virtual Machine Characteristics

Other characteristics that exist for a virtual machine under VM/SP are as follows:

1. 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-protect-key mechanism of the processor and channels operates in these cases, but is unable to provide predictable protection to other virtual machines. In addition, violation of this restriction may compromise the soundness of the system. The results are unpredictable.

2. A two-channel switch can be used between the processor running a virtual machine under VM/SP and another processor.

3. The DIAGNOSE instruction cannot be issued by the virtual machine for its normal function. VM/SP uses this instruction to allow the virtual machine to communicate system services requests. The Diagnose interface requires the operand storage addresses passed to it to be real to the virtual machine issuing the DIAGNOSE instruction. For more information about the DIAGNOSE instruction in a virtual machine, see the *VM/SP System Programmer's Guide.*

4. A control unit, normally, never appears 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 IPL command cannot simulate self-modifying IPL sequences of dedicated unit record devices or certain self-modifying IPL sequences of tape devices.

6. VM/SP spooling does 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 does not support count control on the virtual 1052 operator's console.

8. Programs that use the integrated emulators function only if the real system has the appropriate compatibility feature. VM/SP does not attempt simulation. The DOS emulator running under OS or OS/VS is not supported under VM/SP.

9. The READ DIRECT and WRITE DIRECT instructions are not supported for a virtual machine.

10. The SET CLOCK instruction cannot be simulated and is ignored if issued by a virtual machine. The STORE CLOCK instruction is a nonprivileged instruction and cannot be trapped by VM/SP; 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 interrupt 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 following formula:

    $$\frac{\texttt{virtual machine size}}{\texttt{2}} \texttt{ = starting address of the overlaid page}$$

    or the hexadecimal value 20000, whichever is smaller.

14. To maintain data 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 ended with an interrupt whose CSW indicates incorrect length and a channel program check.

    Notes:

    a. A data transfer sequence is defined as one or more read or write CCWs connected via chain data. 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 not the same as those used by the data security of VM/SP and will give an error when used.

15. When an I/O error occurs on a device, the hardware maintains a conditional connection for that device until a SENSE channel command is executed and sense data is recorded. That is, no other I/O activity can occur on the device during this time. Under VM/SP, the conditional connection is maintained until the SENSE command is executed, but I/O activity from other virtual machines can begin on the device while the sense data is being reflected to the virtual machine. Therefore, you should be aware that on a shared disk, the access mechanism may have moved during this time.

16. 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, thus changing the mode setting of all 7-track tape devices attached to that control unit.

17. A shared system or one that uses discontiguous saved segments cannot be loaded (IPL) into a virtual machine running in the virtual=real area.

18. 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.

19. The 3066 is supported as a 3215. It is not supported as a graphics editor; therefore, it is recommended that the NODISP option of the EDIT command be used when editing in a 3066.

20. The Program Controlled Interruption (PCI) FETCH option for load module calling is not supported for OS/MFT or VS1.

21. 3081 processors do not permit use of one megabyte segments for virtual machines. Any attempt by a relocatable virtual machine using 1Mb segments to use the DAT facility for address translation, results in a translation exception.

22. The Input/Output Configuration Program must not be run while single processor mode is active on the system. Objectionable results may occur.

23. OS/VS2 is supported in uniprocessor mode only.

MSS Restrictions

1. There are two OS/VS system data sets associated with a 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 in either 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 the same: the definition of 3330V addresses in the MSC tables, the DMKRIO module, and the IOGEN for any OS/VS system running in a virtual machine with a dedicated MSC port. The reason for this, and the way to ensure it, is explained in the *VM/SP System Programmer's Guide.*

4. Each active volume in the MSS must have a different 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 VM/SP 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. CP commands that require action from the central server must not be issued from the central server virtual system.

8. If virtual volumes are to be shared between processors, the virtual machine must handle cylinder faulting.

CMS Restrictions

The following restrictions apply to CMS, the conversational subsystem of VM/SP:

1. CMS runs only on a virtual processor provided by VM/SP.

2. The maximum sizes (in cylinders or blocks) of CMS minidisks are as follows:

Device Type	Model(s)	CMS/VSAM	CMS 800-byte Format	CMS 512, 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	-	558,000 blocks	not supported	558,000 blocks
3375	-	959 cyls.	182 cyls.	959 cyls.
3380	-	885 cyls.	121 cyls.	885 cyls.

3. If CMS cannot calculate a true time, it will display *.** in place of n.nn or x.xx.

4. Programs that operate under CMS are encouraged to use documented interfaces. Those programs that 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.

5. CMS uses VM/SP spooling to perform unit record I/O. However, a program running under CMS can issue its own SIOs to attached dedicated unit record devices.

6. Only those OS and VSE tasks that are simulated by CMS can be used to run OS and VSE programs produced by language processors under CMS.

7. Many types of object programs produced by CMS (and OS) languages can be run 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.

8. CMS can read sequential and partitioned OS data sets and sequential VSE files, by simulating certain OS and VSE system services.

   The following restrictions apply when CMS reads OS data sets that reside on OS disks:

   - Read-password-protected data sets are not read unless they are VSAM data sets.

   - 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.

- Keys in data sets with keys are ignored and only the data is read, except for VSAM.

- User labels in user labeled data sets are ignored.

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 CONCAT options of FILEDEF and the PDS option of MOVEFILE) also do not apply to VSE sequential files.

- The following types of VSE 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 and there is no end-of-volume switching.

- User labels in user labeled files are ignored.

- Since VSE files do not contain BLKSIZE, RECFM, or LRECL parameters, these parameters must be specified via FILEDEF or DCB parameters; otherwise, BLKSIZE=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, since the CMS/DOS implementation of the DUMMY statement corresponds to VSE implementation. Specifying the DUMMY option with the DLBL command will cause an execution-time error.

9. Assembler program usage of the ISAM Interface Program (IIP) is not supported.

10. CMS/DOS support is based on the VSE/Advanced Functions program product. With VSE, prior releases of VSAM are not supported under CMS/DOS.

11. System logical units (SYSIN, SYSRDR, SYSIPT, SYSLST, and SYSPCH), are not supported for VSE formatted FB-512 devices because the SYSFIL function (SVC 103) of VSE is not supported under CMS/DOS.

12. Programs created using CMS/DOS are not recommended for transfer directly to a VSE machine because:

- The CMS/DOS VSE linkage editor is designed to link edit VSE programs under CMS/DOS only.

- Programs created using the CMS/DOS assembler may have incorrect ESD's. In this case, 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.

13. Setting the PSW EC mode bit on is not recommended because CMS handles interrupts in BC mode only.

14. 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 block 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[11]. 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.

Miscellaneous Restrictions

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 system to be put into an enabled wait state.

2. If you plan to define more than 64 virtual devices for a single virtual machine, be aware that any single request for free storage in excess of 512 doublewords (a full page) can cause an error message to be issued if storage cannot be obtained. Tables for virtual devices for a virtual machine must reside in contiguous storage. Therefore, two contiguous pages of free storage must be available in order to logon a virtual machine with more than 64 virtual devices, (three contiguous pages for a virtual machine with more than 128 virtual devices, etc.). Contiguous pages of free storage are sure to be available only immediately after IPL, before other virtual machines have logged on. Therefore, a virtual machine with more than 64 devices should be the first to logon after IPL. The larger the real machine size, the lesser the possibility of this occurring.

3. For remote 3270s, VM/SP supports a maximum of 256 binary synchronous lines minus the number of 3704/3705 Communications Controllers.

4. If an I/O device (such as a disk or tape drive) drops ready 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 LOGOFF and

---

[11] The DASD address of the Y-DISK will be whatever was specified when CMS was generated. For the standard system this will be 19E.

FORCE commands are not effective because they do not complete until all waiting I/O is finished. The system operator should determine which I/O device is involved and make that device ready once more.

5. Any modifications to local OPTIONS COPYFILE, unless otherwise specified in existing documentation, is not supported.

6. If you are using an IBM 3031, 3032, or 3033 processor, you must dedicate the service record file (SRF) device to VM/SP. Thus, the channel on which the SRF is located cannot be dedicated to any virtual machine.

7. When using the SPOOL, DEDICATE, and SPECIAL directory control statements to define virtual devices, specify virtual addresses that do not conflict or compete with the virtual control unit interface. This conflict or competition occurs because devices can require special I/O interface protocol from control units such as shared and nonshared subchannel operations. Putting devices that require different real control units on the same virtual control unit can result in a hung or busy condition. To avoid this problem, users must define (and separate) devices within their own control unit range. For example, if the directory entry specifies:

   ```
   SPOOL 102 3211
   SPECIAL 103 3270
   ```

   the control unit 0 on channel 1 controls both a nonshared device (the 3211 printer) and a shared device (the 3270 display unit). Processing of channel programs involving these two devices can result in a hung or busy condition.

8. If you are using an 8809 tape device, it is required to have a tape mounted with the drive ready before issuing a CP DETACH command. This allows the tape drive mode to be returned to the default mode when execution of the command completes.

9. 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 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).

10. When using two channel-to-channel adapters (dedicated to virtual machines), and the CTCAs are operating on the same channels on each CPU, then the virtual machines should use the control CCW to prevent locking out the channel.

11. If using conmode 3270 with a guest SCP such as MVS, SCRNSAVE ON must be specified; otherwise, unpredictable results may occur.

12. If the number of virtual devices exceeds the formula (7FFF divided by VDEVBLOK size) unpredictable results may occur. This is due to the design usage of the virtual control block structure.

13. When TERMINAL CONMODE 3270 is in effect, tracing should not be done at the same console, as unpredictable results may occur.

14. When using the 3081 processor, V=V users can no longer use 1 Mb (megabyte) segments for constructing shadow tables.

15. If a terminal has an inhibited (non-display) 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, therefore displaying any data typed in at that point. The clear key can be used following the PF key to rewrite the inhibited read.

16. 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.

17. 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.)

18. If a 3278 Model 4 is switched to alternate mode (43 line screen) and the terminal is then dialed to a virtual machine, the terminal will not be reset to default mode (24 line screen). The 3278 Model 4 will remain in alternate mode if alternate mode is started after the logo has been written and an erase write alternate has been issued to the screen.

19. VM support of the 3800 printer as a non-dedicated 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 hex 63 (LOAD FCB) CCW must be done to set up any other FCB for the spool file. This support is different from VM support of other virtual spooling devices, such as a 3211 printer.

20. In Single Processor Mode, CP-owned volumes must be on strings and control units that are not online to the MVS native side.

21. Users with the cross memory feature(#6850) installed and MVS Release 2 or 3, cannot use the single processor mode (SPM) or non-disruptive transition (QVM) functions of VM/SP.

Index

SC19-6201-3

VM/SP Planning Guide and Reference (File No. S370/4300-34) Printed in U.S.A. SC19-6201-3

VM/SP Planning
Guide and Reference
SC19-6201-3

READER'S COMMENT FORM

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate. Comments may be written in your own language; English is not required.

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

	Yes	*No*
• Does the publication meet your needs?	☐	☐
• Did you find the material:		
Easy to read and understand?	☐	☐
Organized for convenient use?	☐	☐
Complete?	☐	☐
Well illustrated?	☐	☐
Written for your technical level?	☐	☐

- What is your occupation? ____________________

- How do you use this publication:

As an introduction to the subject?	☐	As an instructor in class?	☐
For advanced knowledge of the subject?	☐	As a student in class?	☐
To learn about operating procedures?	☐	As a reference manual?	☐

Your comments:

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

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

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

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

Fold Fold

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

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

IBM®

Cut or Fold Along Line

SC19-6201-3

VM/SP Planning Guide and Reference (File No. S370/4300-34) Printed in U.S.A. SC19-6201-3