IBM

SC24-5237-04

Virtual Machine/System Product

Installation Guide

Release 6

IBM

Virtual Machine/System Product

SC24-5237-04

Installation Guide

Release 6

Fifth Edition (August 1988)

This edition, SC24-5237-04, is a major revision of SC24-5237-03, and applies to Release 6 of Virtual Machine/System Product (VM/SP), Program Number 5664-167, and to all subsequent releases of this product until 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, 30xx, 4300, and 9370 Processors Bibliography*, GC20-0001, for the editions that are applicable and current.

Summary of Changes

For a list of changes, see "Summary of Changes" on page X-1.

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

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM licensed program in this publication is not intended to state or imply that only IBM's licensed program may be used. Any functionally equivalent program may be used instead.

Ordering Publications

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

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

Preface

The *VM/SP Installation Guide* describes how to install the IBM Virtual Machine/System Product (VM/SP).

If you are installing VM/SP HPO (High Performance Option), use the *VM/SP HPO Installation Guide.*

Information on servicing VM/SP, formerly contained in the *VM/SP Installation Guide*, has been moved to a new book, the *VM/SP Service Guide.*

For a list of the major changes from the previous edition of the *VM/SP Installation Guide*, see the "Summary of Changes" on page X-1.

Who Should Use This Book

This book is intended for system programmers and anyone responsible for planning, installing, and updating a VM/SP system. Whether you are a new or current VM user, you will find this book an important installation aid. A general understanding of System/370 data processing and teleprocessing techniques is assumed.

Before you use this book, review the contents of the *VM/SP Introduction*, the *VM/SP CMS Primer*, and the *VM/SP System Product Editor User's Guide.* When you use this book to install VM/SP, the *VM/SP Planning Guide and Reference* is a co-requisite manual. In addition, you might want to refer to some of the reference material in the *VM/SP Service Guide.* For a list of other related publications, see the "Bibliography" on page X-17.

How This Book Is Organized

This book has eight chapters and ten appendixes:

- Chapter 1, "Introduction," presents an overview of VM/SP installation, including discussions of:
  - VM/SP system components
  - Real and virtual machines
  - Tools and profiles for installation and system generation
  - First level and second level installation
  - The format used in the procedures in this book.
- Chapter 2, "Planning Your VM/SP Installation," outlines some of the areas you might need to plan before you begin your VM/SP installation, such as:
  - Defining your Starter System configuration
  - Planning for the CMS Shared File System (SFS)
  - Planning for the Group Control System (GCS)
  - Planning for the Transparent Services Access Facility (TSAF)
  - Planning for APPC/VM VTAM Support (AVS)
  - Tailoring the system definition files to meet your specific needs.

  An Installation Reference Worksheet is supplied to assist you.
- Chapter 3, "Installing VM/SP Using the Starter System," provides the step-by-step procedure for using the Starter System to install VM/SP from the VM/SP Product Tape. This procedure is intended for those who are installing VM/SP for the first time. This procedure is also recommended for all users who are installing VM/SP Release 6 to update a previous VM/SP release.

- Chapter 4, "Installing VM/SP Using an Existing VM/SP System," provides a step-by-step procedure for using an existing VM/SP system to install VM/SP from the VM/SP Product Tape. This procedure is provided for current VM/SP users who do not have enough available system storage to use the Starter System procedure, or whose present system configuration differs substantially from the configuration defined in the sample files provided on the VM/SP Product Tape.

  Note: If you are migrating from a release prior to VM/SP Release 4, you must use the Starter System procedure.

- Chapter 5, "Verifying Your Installed VM/SP System," provides step-by-step instructions for using the Installation Verification Procedure (IVP) to verify that your installed VM/SP system is working properly.

- Chapter 6, "Installing Saved Segments," presents a general discussion of saved segments and provides step-by-step procedures for installing the CMSDOS, CMSBAM, CMSVSAM, CMSAMS, CMSINST, CMSVMLIB, and CMSFILES saved segments.

- Chapter 7, "Installing a New System National Language," provides the step-by-step procedure for installing a national language from a national language feature tape to replace your current system national language.

- Chapter 8, "Tools," describes installation-related execs and commands.

- Appendixes provide information about:

  - Minidisks and SFS directories reserved for MAINT
  - Restricted logon passwords
  - Sample SPLOAD PROFILE
  - Special options for CP
  - Migrating spool files using SHUTDOWN/WARM IPL
  - Enlarging the CMS nucleus to contain the Y minidisk directory
  - Alternate CMS nucleus placement
  - Alternate GCS nucleus placement
  - Considerations for placement of the DOS-related saved segments
  - Sample EXEC procedure for copying VSE macros into a CMS MACLIB

Contents

Chapter 1. Introduction

Since you placed your VM/SP system order, you have received letters, memoranda, several tapes, and a number of books. Your shipment might have arrived all at once or in several mailings. Each tape or document might include instructions stating or implying that it should be read first. Some might even provide installation procedures. By now, you might not be sure how to get started or where each piece fits.

THIS BOOK IS YOUR PRIMARY VM/SP INSTALLATION DOCUMENT. From here, you will be directed to other documents when necessary.

Pre-Installation Checklist

Do not begin the installation procedure until you complete the pre-installation requirements in this checklist:

__ 1. Read Chapter 1 of the *VM/SP Installation Guide*. This chapter introduces the VM/SP installation process and discusses the following topics:

- Organizing your materials
- VM/SP system components
- Real and virtual machines
- Initial installation and migration
- The Starter System
- The VM/SP Product Tape
- Feature tapes
- System definition files
- Tools and profiles for installation and system generation
- First level and second level installation
- The format used in the procedures in this book.

If you are new to VM/SP, you might find these general discussions helpful. If you have previous VM/SP experience, you might not need some of this information. **Make sure that you understand the concepts of VM/SP installation before you begin the installation procedure.**

__ 2. Read the *VM/SP Program Directory* and any attachments. This document lists the publications that accompany the product and explains the various product packaging options. Make sure that you have received the correct tapes. The *Program Directory* could also include changes, corrections, or updates to the procedures described in the *VM/SP Installation Guide*.

__ 3. Contact your IBM Level One Support Center and request the latest VM/SP installation updates. Do not begin the installation procedure until you obtain this additional information.

If you have received a service tape (called a program update tape, or PUT), and you plan to apply service after you install VM/SP, also request the latest PUT updates.

If you obtain the updates well in advance of the actual installation, you might want to contact the Level One Support Center again just before you begin the installation to make sure that you have the latest possible information.

__ 4. Read Chapter 2 of the *VM/SP Installation Guide.* This chapter discusses how to plan your installation and directs you to the proper installation procedure. Make sure that you identify and satisfy all of the installation requirements.

After you complete these steps, you are ready to install VM/SP.

How to Organize Your Materials

To install the tapes that you have received, you need to use specific installation documents. You might also want to have other documents available for reference. To make the process easier, we suggest that you organize your materials as shown in Table 1-1. These lists generally correspond to the sequence in which you use the materials during the installation procedure. However, keep in mind that:

- You might not have ordered all of the materials shown.
- You might not need all of the materials shown.
- You might have ordered and received materials that are not shown.

Table 1-1. Organizing Your Materials

Tapes	Starter System Tape
	VM/SP Product Tape
	Source feature tape
	EREP distribution tape
	National language feature tape(s)
Installation Documents	*VM/SP Installation Guide*
	VM/SP Program Directory and attachments
Reference Materials	*Memo to Licensees*
	VM/SP Licensed Program Specifications
	VM/SP Planning Guide and Reference
	VM/SP Service Guide
	VM/SP Connectivity Programming Guide and Reference
	VM/SP Connectivity Planning, Administration, and Operation
	VM/SP CMS Shared File System Administration
	VM/SP Administration
	VM System Facilities for Programming
	Device Support Facility User's Guide and Reference
	VM/SP CP General User Command Reference
	VM/SP CP System Command Reference
	VM/SP System Messages and Codes
	VM/SP CMS Command Reference
	EREP User's Guide and Reference

VM/SP Overview

This sections presents a brief overview of the components of VM/SP and the concept of real and virtual machines. For more information about these topics, see the *VM/SP Introduction.*

VM/SP System Components

VM/SP (Virtual Machine/System Product) is an interactive multiple-access operating system that consists of the following components:

- **CP** (Control Program) manages the real system resources, including processor functions, processor storage, and I/O (input/output) devices. It provides an individual working environment for each user on the system. This individual working environment, which simulates the operation of the real machine, is called a **virtual machine**.

- **CMS** (Conversational Monitor System), which runs only on CP, is a virtual machine operating system. It helps the user perform a wide variety of tasks involving application programs, information files, and communication with other users.

- **Procedures Language/VM** is the component containing the VM/SP System Product Interpreter, which processes the Restructured Extended Executor (REXX) programming language.

  Note: Tape files for this component on the VM/SP Product Tape have the file name REXX.

- **IPCS** (Interactive Problem Control System) provides VM/SP installations with an interactive online facility for reporting and diagnosing software failures, and for managing problem information and status.

- **GCS** (Group Control System) is an optional component that supports a virtual machine group operating environment. Members of the group share common storage space, a common virtual machine supervisor, and the ability to communicate with each other. GCS is required if you plan to install Remote Spooling Communications Subsystem (RSCS) Version 2 or Systems Network Architecture (SNA) products.

- **TSAF** (Transparent Services Access Facility) is an optional component that lets a user connect to a resource (such as a data base) within a group of connected systems without knowing the actual user ID and node ID of where that resource resides.

- **AVS** (APPC/VM VTAM Support) is an optional component that provides the services that let a program communicate with other programs on VM and non-VM systems in an SNA network.

Real and Virtual Machines

When installing VM/SP, it is important that you understand the concept of real and virtual machines.

The **real machine** consists of physical devices: the processors, channels, controllers, and I/O devices (disk drives, tape drives, printers, and terminals). Each I/O device has a unique **real (physical) address** to identify it to your processor. In most instances, real addresses are determined when your hardware is installed.

A **virtual machine** is the individual working environment that CP creates for each user ID listed in the CP directory. The CP directory is a file that contains entries that define the virtual machines permitted to log on to the VM/SP system. Each virtual machine simulates the operation of the real machine, including processor functions, I/O devices, and storage.

When a real I/O device is "attached" to a virtual machine, the device is assigned a **virtual address** in the virtual machine. The virtual address is a three-digit hexadecimal number that identifies the device in the virtual machine, just as the real address identifies the device in the real machine. The virtual address could be the same number as the real address, or it could be different, but it must be unique within the virtual machine. CP then maps the virtual address to the real address of the device.

CP divides real direct access storage (DASD) volumes into areas called **minidisks** that function as virtual direct access storage devices (virtual DASD). A minidisk is defined in the CP directory and assigned to a specific virtual machine. Virtual machines can access minidisks to obtain virtual direct access storage. Because it simulates a real DASD in a virtual machine, a minidisk has a virtual address. A minidisk that occupies an entire real DASD volume is called a **full-pack minidisk**.

CMS is the operating system that runs in a virtual machine. To obtain virtual direct access storage, a virtual machine can access minidisks and/or CMS Shared File System (SFS) **directories**. In SFS, collections of minidisks called **file pools** are assigned to **server** virtual machines. A server is a set of programs that execute in a virtual machine to manage a resource, such as a file pool. The VM/SP installation procedure includes steps for the generation of a system file pool and the optional generation of a user file pool. A virtual machine enrolled in a file pool can be allocated an amount of **file space** that can be organized into a series of hierarchical SFS directories. These SFS directories can be accessed in the same manner as minidisks.

For most operations in the installation procedure, you must be logged on to the MAINT virtual machine. MAINT is the "system owner," responsible for installing and maintaining the system. Files are loaded to specific minidisks and SFS directories reserved for MAINT. For more information, see Appendix A, "Minidisks and SFS Directories Reserved for MAINT" on page A-1.

Installation Overview

VM/SP installation consists of the following basic processes (not necessarily in this order):

- Formatting DASD volumes and minidisks
- Loading files from tapes to minidisks and SFS directories
- Editing files
- Building system nuclei
- Building file pools.

These processes require you to enter commands and respond to system prompts to define the unique configuration of your VM/SP system.

Types of Installations

There are two types of VM/SP installations:

- **Initial installation** is the generation of a VM/SP system where no VM/SP system presently exists.

- **Migration** is the generation of a VM/SP system as an upgrade of an existing previous-release VM/SP system.

The type of installation determines the procedure that you use to install VM/SP. To do an initial installation, you must use the Starter System procedure. To do a migration, you can use the Starter System procedure (recommended) or you can use the existing VM/SP system procedure. Both installation procedures are provided in this book.

The Starter System

The Starter System is an abbreviated VM/SP system that contains enough CP and CMS function to help you load and build the CP and CMS components of your new system. Once the new CP and CMS are installed and running, you can use them to install the rest of your system.

You can use the Starter System to do an initial installation or a migration. **However, the Starter System is not a production system. Do not try to use the Starter System for any functions not indicated in the Starter System installation procedure.**

When you use the Starter System to install VM/SP, the Starter System determines the allocation of space on the DASD volume where CP and CMS reside (called the system residence volume, or VMSRES). Therefore, the Starter System used to install VM/SP must match the device type of the DASD used for VMSRES. The following Starter System versions are available:

- 3350
- 3375
- 3380
- 9313/9332
- 9335/3370.

The Starter System is distributed in three tape formats:

- 9-track 1600 bpi (bytes per inch) reel
- 9-track 6250 bpi reel
- 18-track 38K bpi cartridge.

As shown in Figure 1-1, the Starter System Tape contains three tape files:

- File 1 contains:
  - Device Support Facility, a stand-alone program that you can use to initialize new DASD and perform other DASD-related functions
  - Format/Allocate, a stand-alone program that you use to format your DASD volumes before you load any files to them.
- File 2 contains DASD Dump/Restore (DDR), a stand-alone program that you use to load the Starter System from the Starter System Tape to your system residence DASD.
- File 3 contains the Starter System.

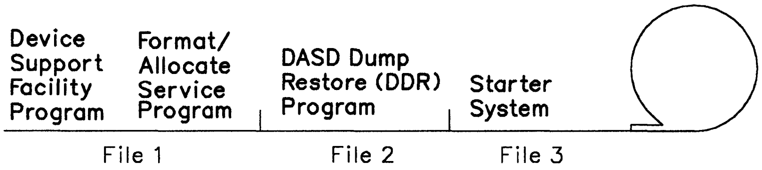

Device Support Facility Program
Format/ Allocate Service Program
DASD Dump Restore (DDR) Program
Starter System
File 1
File 2
File 3

Figure 1-1. VM/SP Starter System Tape File Layout

The VM/SP Product Tape

The VM/SP Product Tape contains:

- Tools and profiles for installation and system generation
- Sample files
- Object code and service code for all components
- HELP files
- Source code for CP, IPCS, and some CMS modules
- Tuning and accounting modules for AVS.

Object code files contain data that is in machine-readable form. Service code consists of delta files (update files and serviced text decks) and apply files (auxiliary control files) that are supplied with the VM/SP base to establish a service history, which is required for applying future service. Source code files contain data that is not in machine-readable form, but which can be assembled or compiled into object code.

The VM/SP Product Tape is distributed in the following tape formats:

- 9-track 1600 bpi reel
- 9-track 6250 bpi reel
- 18-track 38K bpi cartridge.

The layouts of the VM/SP Product Tape volumes are shown in Table 1-2. Throughout the rest of this book, the term *product tape* refers collectively to all the volumes of the VM/SP Product Tape.

Feature Tapes

Additional features that you order to install on VM/SP are provided on feature tapes. For example:

- Source code for GCS, TSAF, and selected CMS modules
- National languages
- Optional feature program products.

Feature tapes are distributed on 9-track 1600 or 6250 bpi tape reels, or on 18-track 38K bpi tape cartridges.

System Definition Files

The system definition files provide information that is used during system generation to define various VM/SP system parameters. Sample system definition files are shipped on the VM/SP Product Tape. However, because these files contain default parameters and other data that might not meet your requirements, you must edit some of the files during installation and tailor the data to define the unique configuration of your system.

VM/SP Product Parameter File

The VM/SP product parameter file (5664167E $PPF) is a multipart file that contains installation and service parameters for each VM/SP component, including control options, minidisk and SFS directory assignments, and a part type/function list. For more information about the structure and function of this file, see the *VM/SP Service Guide*.

Table 1-2. VM/SP Product Tape File Layout

1600 BPI	6250/38K BPI
Tape 1 File 1: Header File 2: Installation tools File 3: System generation tools File 4: System tools File 5: Sample system files (not DASD specific) File 6: Sample system files (DASD model specific) File 7: Sample system files (DASD type specific) File 8: CP delta code File 9: CP apply code File 10: CP object code File 11: CMS base code Tape 2 File 1: Header File 2: Procedures Language/VM delta code File 3: Procedures Language/VM apply code File 4: Procedures Language/VM object code File 5: CMS delta code File 6: CMS apply code File 7: CMS system code Tape 3 File 1: Header File 2: IPCS delta code File 3: IPCS apply code File 4: IPCS object code File 5: HELP files Tape 4 File 1: Header File 2: GCS delta code File 3: GCS apply code File 4: GCS interface files File 5: GCS object code File 6: TSAF delta code File 7: TSAF apply code File 8: TSAF interface files File 9: TSAF object code File 10: AVS delta code File 11: AVS apply code File 12: AVS interface files File 13: AVS object code Tape 5 File 1: Header File 2: CP source code Tape 6 File 1: Header File 2: IPCS source code File 3: CMS source code	Tape 1 File 1: Header File 2: Installation tools File 3: System generation tools File 4: System tools File 5: Sample system files (not DASD specific) File 6: Sample system files (DASD model specific) File 7: Sample system files (DASD type specific) File 8: CP delta code File 9: CP apply code File 10: CP object code File 11: CMS base code File 12: Procedures Language/VM delta code File 13: Procedures Language/VM apply code File 14: Procedures Language/VM object code File 15: CMS delta code File 16: CMS apply code File 17: CMS system code File 18: IPCS delta code File 19: IPCS apply code File 20: IPCS object code File 21: HELP files File 22: GCS delta code File 23: GCS apply code File 24: GCS interface files File 25: GCS object code File 26: TSAF delta code File 27: TSAF apply code File 28: TSAF interface files File 29: TSAF object code File 30: AVS delta code File 31: AVS apply code File 32: AVS interface files File 33: AVS object code Tape 2 File 1: Header File 2: IPCS source code File 3: CP source code File 4: CMS source code

CP and CMS Nucleus Definition Files

These files provide information required when building CP and CMS nuclei:

- The **CP directory (VMUSERS DIRECT)** identifies the virtual machines (users) permitted to log on to your system. A directory entry for a virtual machine defines the user ID of the machine, the password that enables that user ID to log on, the virtual processor size, the default file pool, the virtual input/output device configuration, the minidisks assigned or linked to the virtual machine, and other virtual machine parameters.
- The **CP system control file (DMKSYS ASSEMBLE)** describes the CP system residence device (which contains the CP nucleus), system storage size, CP-owned DASD, and other system parameters.
- The **real I/O configuration file (DMKRIO ASSEMBLE)** describes the I/O (input/output) devices, control units, and channels attached to the processor.
- The **system name table (DMKSNT ASSEMBLE)** defines the name, size, and virtual storage location of named saved systems (such as CMS and GCS) and physical saved segments.
- The **forms control buffer file (DMKFCB ASSEMBLE)** contains real and virtual printer parameters.
- The **CMS nucleus generation profile (DMSNGP ASSEMBLE)** defines responses to the system prompts that are generated when you build a CMS nucleus.

For additional information about the purpose and structure of these files, see the *VM/SP Planning Guide and Reference.*

CMS File Pool Definition Files

These files provide information required for building a system file pool and a user file pool:

- The **VMSYS POOLDEF** file defines the configuration of the system file pool (named VMSYS).
- The **VMSERVS DMSPARMS** file contains start-up parameters for the system file pool server machine (VMSERVS).
- The **VMSYSU POOLDEF** file defines the configuration of the user file pool (named VMSYSU).
- The **VMSERVU DMSPARMS** file contains start-up parameters for the user file pool server machine (VMSERVU).

For additional information about the purpose and structure of these files, see *VM/SP CMS Shared File System Administration.*

Other System Definition Files

The following files also affect the configuration of your system:

- The **DMKBOX ASSEMBLE** file defines system logos, such as the logo that appears on your screen when you log on to your system, or the logo that appears on printer separator pages. If you want to change the design and/or contents of these logos, you must tailor this file.
- The **RPWLIST DATA** file contains a list of restricted logon passwords. These passwords, which were used in the sample CP directories supplied in previous VM/SP releases, are no longer allowed because the use of standard passwords

violates system security. For maximum system security, you are strongly encouraged to define a unique logon password for each user in your system. A VM/SP feature called Auto-Deactivation of Restricted Passwords (ADRP) uses the RPWLIST DATA file to purge the system of all restricted passwords.

- The following files define the **universal character set** and **font offset buffer** for various printer types:

  - **DMKUCS ASSEMBLE** (1403)
  - **DMKUCB ASSEMBLE** (3211)
  - **DMKUCC ASSEMBLE** (3203)
  - **DMKPIA ASSEMBLE** (3289)
  - **DMKPIB ASSEMBLE** (3262).

Tools and Profiles for Installation and System Generation

VM/SP provides tools and profiles to simplify and automate many of the installation and system generation operations. These tools and profiles include:

ITASK EXEC This tool, used primarily in the Starter System installation procedure, calls other EXECs and commands to do most installation and system generation tasks.

SPLOAD EXEC This tool directs the loading of tape files from the product tape and feature tapes, using the information contained in the SPLOAD PROFILE. SPLOAD EXEC is called by ITASK, or you can invoke it directly.

SPLOAD PROFILE This file indicates the name and location of each tape file on the product tape, on the source feature tape, and on the national language feature tape; it also specifies the minidisk or SFS directory where SPLOAD EXEC loads each tape file.

VMFBLD EXEC This tool builds CP, CMS, and GCS nuclei, using the information supplied in the VM/SP product parameter file (5664167E $PPF). VMFBLD EXEC is called by ITASK, or you can invoke it directly.

UTILITY EXEC This tool provides five utility functions:

- Printing the following system definition files:

  - VMUSERS DIRECT
  - DMKSYS ASSEMBLE
  - DMKRIO ASSEMBLE
  - DMKSNT ASSEMBLE
  - DMSNGP ASSEMBLE

- Creating a stand-alone service utility tape containing any or all of the following programs:

  - Device Support Facility
  - DASD Dump/Restore program (DDR)
  - Format/Allocate program (FMT)
  - CP Directory program (DIR)

- Creating any or all of the following stand-alone service programs on minidisk:

  - CP Directory program (DIR)

  - Format/Allocate program (FMT)
  - DASD Dump/Restore program (DDR)

- Writing a backup IPLable CP nucleus to tape

- Obtaining DASD-specific information from the $DASD$ CONSTS file.

DIRECGEN EXEC	This tool sets up CP directory entries for optional feature program products.
PASSMOD EXEC	This tool changes logon passwords for user IDs in the CP directory when installing optional feature program products.
INSTFPP EXEC	This tool installs optional feature program products.

For additional information about ITASK EXEC, SPLOAD EXEC, and SPLOAD PROFILE, see Chapter 8, "Tools" on page 8-1. For additional information about VMFBLD EXEC and UTILITY EXEC, see the *VM/SP Service Guide*. For more information about DIRECGEN EXEC, PASSMOD EXEC, and INSTFPP EXEC, see the optional feature program product documentation.

First Level and Second Level Installation

The virtual machine concept of VM/SP allows you to install VM/SP as the operating system of a real machine or as the operating system of a virtual machine.

Installation of VM/SP as the operating system of a real machine is called **first level** installation. At first level, you use the system operator's console to control system functions.

The operating system usually run in a VM/SP virtual machine is CMS. However, other operating systems that can run in a virtual machine include VSE/SP, MVS/SP, and VM/SP. Installation of VM/SP as the operating system of a virtual machine is called **second level** installation. At second level, your "console" is a terminal logged on to the first level VM/SP system through a valid user ID.

When you do a first level installation, you must have access to the real machine. You mount and ready the required DASD volumes and tapes, perform system IPLs (initial program loads), and display PSWs (program status words). This book assumes that you already know how to operate the computer and all of its associated hardware devices; no assistance is provided for those tasks.

When you do a second level installation, you might have access to only your own virtual machine. The first level computer operator may have to mount and ready your DASD and tape volumes and attach them to your first level user ID.

An important difference between first level and second level installation concerns system addresses. At first level, when you IPL your system residence volume, you IPL the real address of the DASD. When you IPL your system residence volume at second level, you IPL the virtual address of a minidisk or DASD attached to your first level user ID. This first level minidisk (which could be a full-pack minidisk) or DASD functions as a "real" DASD volume at second level.

At second level, it is also important to keep in mind that you can communicate with two levels of CP. First level CP provides the virtual machine in which you are installing the VM/SP system that contains your second level CP. At times, you might want to issue commands to one CP or the other. To distinguish first level CP commands from second level CP commands, you use different logical line end characters preceding the commands. Setting and using line end characters is explained in the installation procedure. **First level and second level operations are clearly identified in the appropriate steps of the installation procedures in this book.** Figure 1-2 shows the relationship between first level and second level operation.

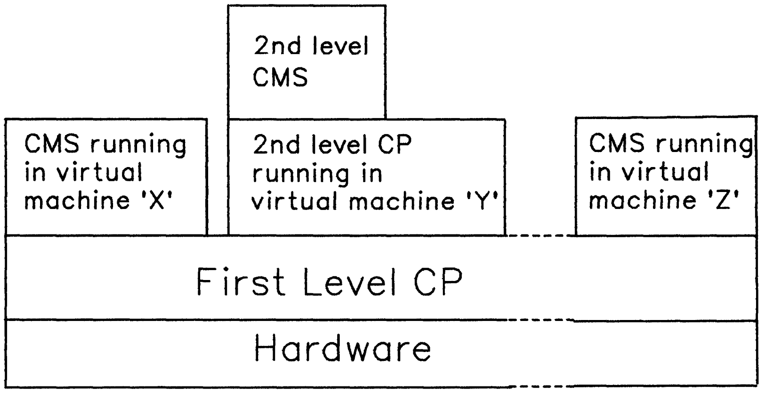

2nd level CMS
CMS running in virtual machine 'X'
2nd level CP running in virtual machine 'Y'
CMS running in virtual machine 'Z'
First Level CP
Hardware

Figure 1-2. Relationship between First Level and Second Level Operation

The Format Used in the Procedures in This Book

This book contains a series of installation procedures. The page layout and text formats used in these procedures are designed to help you identify the following types of information:

- System messages, responses, and prompts

  Note: VM/SP messages are shown with their associated message identifiers. A message identifier consists of a component prefix, a module code, a message number, and a severity code. VM/SP messages are explained in *VM/SP System Messages and Codes.* A message, response, or prompt generated by the VM/SP system, by a program running on the system, or by a stand-alone program is generally referred to by the generic term "message."

- User entries and responses
- Console and terminal keys that signal the system
- Comments about specific actions in the procedure.

A procedure consists of a series of steps; these steps usually correspond to natural break points in the procedure. Each step in the procedure describes one or more operations. Some procedures in this book assume a higher level of experience, and therefore do not provide as much detail.

The number of a step in a procedure is indicated by a heading at the top of the page. The description of each step contains two sections:

- The **Overview** outlines the operations that take place in the step. It could also contain additional information about the function of the step and describe the operation of EXECs and commands.

- The **Procedure** shows the sequence of system messages and user entries/responses required to complete the step. An operation is indicated with a ■ symbol.

Under the ■ symbol, the page is set in a double column layout. The left column displays the **representative** flow of system messages and user entries/responses **when installing VM/SP on a 3380 DASD**. Depending on the actual system that you are using, you might receive additional messages or fewer messages or ones that are slightly different from those shown. The right column contains comments about the sequence shown in the left column.

For example,

■ This line tells you to do a certain operation. System messages and user entries/responses (if any) are shown below in the left column. Different text formats are used to distinguish between the different types of information.

```
System messages, responses, and prompts
are shown in this column in this typeface.
```

This column contains comments.

Messages, responses, and prompts generated by the system are shown in black type on a grey shaded background; variables are shown in italics.

⋮

The three vertical dots (which **do not** actually appear on your screen) represent other messages that **could** appear at the indicated point in the procedure.

literal user entry/response

A literal user entry/response is shown in bold blue type. You must enter a literal user entry/response exactly as shown. Unless the procedure indicates otherwise, you can use lowercase.

variable user entry/response

A variable user entry/response is shown in underlined blue lowercase italics. It generally indicates the *kind* of information that is required; you must enter the *specific* data that applies to your installation. The comments column explains your options.

Something That You Might Have to Do

When part of the procedure is set off from the rest by these qualification lines, the enclosed operation is necessary only if you meet the requirement stated on the opening qualification line.

End of Something That You Might Have to Do

Something That You Might Want to Do

When enclosed in a box like this, the indicated operation is not part of the required procedure, but is something extra that you might want to do. This format is also used to supply some additional nonprocedural information.

WARNING

This message alerts you to a situation that could cause loss or destruction of your data.

ATTN, ENTER, and PFn

These symbols represent the keys on your console or terminal that you press to signal the system.

ATTN interrupts whatever the system is doing, so that you can do something else.

ENTER tells the system that you have completed your input.

PFn tells the program to do the function assigned to program function key *n*, such as PF1 or PF3.

After you make an entry or response in a procedure in this book, you must press ENTER to request the system to accept your data. This use of ENTER is implied and is **not shown** in the procedure.

If you have no input to provide, you still must press ENTER to tell the system to continue the operation. This use of ENTER is called the **null entry**.

If a system prompt has a pre-defined user response (called a **default**), you can use the null entry to tell the system to take the default. You also use the null entry to tell the system to begin the next operation. The null entry is the **only** use of the ENTER symbol shown in the procedures in this book.

The specific keys that you press to obtain the ATTN and ENTER functions depend on the device that you are using, as shown in Table 1-3 on page 1-14.

Table 1-3. Console/Terminal Keys That Signal the System

Device	**ATTN function**	**ENTER function**
1052	RESET LINE	RETURN or EOB
2741	ATTN	RETURN
3101	BREAK	BREAK, SEND, new-line
3210, 3215	REQUEST	END
3277	ENTER or DUP/PA1	ENTER
3278, 3279	ENTER or PA1	ENTER
3767	ATTN	EOB or EOM
System/370 138, 148, 158	ENTER	ENTER

Chapter 2. Planning Your VM/SP Installation

> IMPORTANT
>
> **READ THIS CHAPTER BEFORE YOU BEGIN YOUR INSTALLATION.**

Careful planning helps you achieve a smooth and successful installation. This chapter outlines some of the areas that you may need to plan:

- Selecting the installation procedure
- Identifying your DASD requirements
- Defining your Starter System configuration
- Planning for the 308x Processor Complex
- Planning for the CMS Shared File System (SFS)
- Planning for the Group Control System (GCS)
- Planning for the Transparent Services Access Facility (TSAF)
- Planning for APPC/VM VTAM Support (AVS)
- Tailoring the system definition files
- Migrating spool files
- Installing national languages on your system.

This chapter also indicates where you can find additional information on these topics. Of course, you only need to consider the topics that apply to your installation.

To help you plan your installation, a sample Installation Reference Worksheet is provided on pages 2-13 and 2-14. Make a copy of the Worksheet and use it as directed in this chapter to record essential information about your system. During the installation procedure, you need to enter this information in response to various system prompts.

Selecting the Installation Procedure

Your first planning decision is to select one of the following VM/SP installation procedures:

- Starter System procedure
- Existing VM system procedure.

The procedure that you select depends on the *type* of VM/SP installation:

- Initial installation
- Migration.

Initial Installation

Initial installation is the generation of a completely new first level or second level VM/SP system that does not use portions of any existing VM/SP system. To do an initial installation, you must use the Starter System procedure (Chapter 3).

In the Starter System procedure, the ITASK EXEC installation tool simplifies and automates many of the installation and system generation operations by calling other tools, such as the SPLOAD EXEC and the VMFBLD EXEC. These tools use data contained in other files (SPLOAD PROFILE, 5664167E $PPF, and DMSNGP ASSEMBLE). If you accept the sample values in the files supplied by IBM, you can complete the installation with fewer entries and decisions. If you do not want to accept the sample values, you can modify the files to substitute your own values, as long as you understand the implications of your changes.

Migration

Migration is the generation of a new first level or second level VM/SP system as an upgrade from a previous VM/SP system release. To migrate from a release prior to VM/SP Release 4, you must use the Starter System procedure (Chapter 3). To migrate from VM/SP Release 4 or Release 5, the Starter System procedure is recommended. However, if you do not have enough DASD available to use the Starter System (see "Identifying Your DASD Requirements"), or if your system configuration differs substantially from the configuration defined in the sample system files, you can use the existing VM/SP system procedure (Chapter 4). This procedure assumes a greater working knowledge of VM/SP and the installation process.

Note: To use the existing VM/SP system procedure, you must first load and apply specific CP and CMS service. See "Migration Considerations" on page 4-2.

Identifying Your DASD Requirements

To install the VM/SP base (initial installation or migration), you need the following DASD volumes:

Label	*Function*
VMSRES	System residence volume
VMPK01	Additional system volume
VMPK02	Additional system volume
VMPK04	Source code volume

Notes:

1. VMPK02 is required only for 9313 DASD.
2. VMPK04 is required only if you plan to load source code.
3. If you are installing VM/SP at second level, these volumes could be minidisks or real DASD attached to your first level user ID.
4. If you are using the Starter System procedure, the VMSRES volume must be on a DASD type that matches the Starter System version: 3350, 3375, 3380, 9313/9332, or 9335/3370.
5. Because it is a limited function system, the Starter System accepts only 4-bit addresses (X'*nn*7' or less) for the VMSRES and VMPK01 volumes (and VMPK02, if used).
6. If you are using the existing VM system procedure, you can install certain parts of your system on additional DASD volumes (for example, you can install the GCS component on its own volume).

Record your DASD volume numbers (if you are using volume numbers), device types, and volume addresses in the Installation Reference Worksheet:

- If you are installing at **first level**, record the **real** address of each volume.

- If you are installing at **second level**, record the first level **virtual** address of each volume, which functions as a "real" volume at second level. You might also want to record the real address of the volume; however, you might have to obtain this information from the system operator.

Defining Your Starter System Configuration

This information is required only if you are using the Starter System procedure to install VM/SP.

In Step 3 of the Starter System procedure you are prompted for the addresses and device types of certain devices in your system. The Starter System provides default values, which are indicated on the sample Installation Reference Worksheet.

If your address or device type does not match the default, you must respond with the correct value. You might find it helpful to record this information now in your Worksheet:

- If you are installing at **first level**, record the **real** device address and device type.
- If you are installing at **second level**, record the **first** level **virtual** device address and device type, which function as "real" at second level. You might have to obtain this information from the system operator.

If you do not have the prompted device in your system, you can accept the address and device type defaults indicated on the sample Worksheet.

Planning for the 308x Processor Complex

In 308x Processor Complex installations, device addresses defined in the input/output configuration data set (IOCDS) and stored in the processor controller must match the addresses that you define in the real I/O configuration file (DMKRIO) in the CP nucleus. For more information, see the *VM/SP Planning Guide and Reference.*

Planning for the CMS Shared File System (SFS)

The CMS Shared File System (SFS) is an extension of the CMS file system that offers users an alternative to personal minidisks and shared minidisks for file management and file sharing. In SFS, file space is allocated to users from collections of minidisks called file pools. A file space can be organized into hierarchical directories that can be accessed in the same manner as minidisks (with the ACCESS command). Files can be stored at any level. For more information about SFS, see the *VM/SP Introduction* or the *VM/SP CMS User's Guide.*

The VM/SP installation procedure includes a step for building a system file pool (VMSYS) and an optional step for building a user file pool (VMSYSU). In later installation steps, GCS, TSAF, and AVS files are loaded to directories defined in the system file pool.

The system file pool and the user file pool are each managed by a server virtual machine: VMSERVS and VMSERVU respectively. These server machines are defined in the sample CP directory. In addition, file pool definition files (VMSYS POOLDEF and VMSERVS DMSPARMS for the system file pool, VMSYSU

POOLDEF and VMSERVU DMSPARMS for the user file pool) are shipped on the product tape. These files provide data required for building the file pools.

Before you build the file pools, you should evaluate your system DASD requirements. You might want to tailor some of IBM-supplied file pool parameters. However, you should not change any of the supplied data unless you have a thorough working knowledge of the CMS Shared File System. For information about CMS DASD requirements and the VMSERVS and VMSERVU CP directory entries, see the *VM/SP Planning Guide and Reference*. For information about file pools, file pool servers, and file pool administration, see *VM/SP CMS Shared File System Administration*.

Planning for the Group Control System (GCS)

If you plan to install the Remote Spooling Communications Subsystem (RSCS) Version 2 or Systems Network Architecture (SNA) products, you must install the GCS component.

GCS is a named saved system that can be IPLed by a "group" of one or more virtual machines. GCS provides a variety of services to the group, such as supporting a native VM/SNA network that functions as part of your VM/SP system without help from a guest operating system. For more information about GCS concepts and facilities, see to the *VM/SP Planning Guide and Reference*.

If you plan to install GCS, you must define both the GCS saved system and the virtual machine group. A GCS saved system is already defined for you in the sample system name table (DMKSNT ASSEMBLE), but you might want to tailor it to meet your own requirements. To define the virtual machine group, you invoke the GROUP EXEC to create a group configuration file. Information in the group configuration file is used to build the GCS nucleus.

You can install more than one GCS nucleus on your VM/SP system. Planning for each GCS nucleus involves the following operations:

- **Calculating your GCS storage requirements**

  You need to calculate how much private storage and common storage GCS requires. You use these figures during the installation procedure when you:

  - Tailor the GCS entry in the CP directory
  - Tailor the GCS entry in DMKSNT
  - Build the GCS configuration file.

  The *VM/SP Planning Guide and Reference* provides a procedure that you can use to calculate your storage requirements.

- **Planning the GCS entry in the CP directory**

  The sample directory (VMUSERS DIRECT) supplied on the product tape contains an entry for a GCS recovery machine. The user ID is GCS. During the installation procedure, you **must** modify this entry to remove NOLOG from the USER statement and substitute a valid non-restricted logon password. (For a list of the restricted passwords, see Appendix B, "Restricted Logon Passwords" on page B-1.) You might also have to change the machine size if you calculated a greater storage requirement.

  In addition, you might want to create new CP directory entries for the virtual machines that will use GCS. To use GCS, a virtual machine must have a link to the GCS system minidisk (MAINT 595). This minidisk must be accessed even after the GCS saved system is built to give the user access to GCS files that are not in the GCS nucleus. A link to the GCS system minidisk extension (MAINT 59E) is optional, depending on whether or not your GCS applications use this minidisk.

The *VM/SP Planning Guide and Reference* explains the structure of CP directory entries.

- **Planning the GCS entry in the system name table**

  When you have calculated your storage requirements for GCS, you can plan the GCS entry in the system name table (DMKSNT ASSEMBLE). As supplied on the product tape, DMKSNT contains a NAMESYS entry for GCS. During the installation procedure, you can tailor this entry to define your GCS system.

  The SYSNAME that you specify in this entry (the supplied name is GCS) must be the same "system name" that you specify when you build the GCS configuration file. If you plan to install multiple GCS nuclei, you must create a new NAMESYS entry for each additional nucleus and specify a unique SYSNAME for each one.

  The *VM/SP Planning Guide and Reference* explains the structure of system name table entries.

- **Planning the GCS configuration file**

  You build the GCS configuration file during the installation procedure by invoking the GROUP EXEC, which presents a series of panels (screens) that ask you to supply certain information about your GCS configuration. You may find it helpful to record this information now in your Installation Reference Worksheet (space is provided to define up to three GCS configurations).

  Note: If you do not have a full-screen display device, you cannot use the GROUP EXEC, because you cannot display the panels. In that case, you must build the configuration file manually during the installation procedure using the build macros described in the *VM/SP Group Control System Command and Macro Reference*. You might want to look at that book now to become familiar with the operation of those macros.

  System Name This is the name of your GCS saved system. You must select a unique name for each GCS saved system that you want to install. This name must match the SYSNAME parameter in the NAMESYS entry that defines this saved system in the DMKSNT file. GROUP EXEC builds a configuration file named *systemname* GROUP that becomes the blueprint for your virtual machine group.

  Authorized User IDs These are the user IDs of the virtual machines allowed to run in supervisor state and allowed to use GCS functions. For example, these user IDs have authorization to change common storage. You must include the recovery machine and the common dump receiver (if defined). If you plan to install AVS, you must also include the AVSVM virtual machine.

System Disk vdev

This is the virtual address of the GCS system minidisk. MAINT 595 is the default, as defined in the SPLOAD PROFILE and in the CP directory. This address must match the VSYSADR parameter in the NAMESYS entry that defines this named saved system in the DMKSNT file. To use GCS, a virtual machine must have a LINK statement for this minidisk in the CP directory, or the user must enter a LINK command. The user must also access the minidisk, either through the PROFILE EXEC or by entering an ACCESS command.

System Disk Ext. vdev

This is the virtual address of the GCS system minidisk extension. MAINT 59E is the default, as defined in the CP directory. This minidisk is available for use by applications, and links to this minidisk are optional.

Common Dump Receiver

Normally, when a user's virtual storage is dumped to trace a problem, the dump file is spooled to the user's virtual reader. However, when you build the configuration file, you can select one authorized user ID to receive all GCS virtual storage dumps. If one of your GCS user IDs is the RSCS virtual machine, you **must** name a common dump receiver, because the RSCS virtual machine cannot process a dump file spooled to its own virtual reader.

Recovery Machine

You **must** specify one authorized user ID to act as the recovery machine. The sample CP directory defines a recovery machine named GCS. The recovery machine must be the first machine to join your group, and is responsible for cleaning up system resources when other machines using those resources reset.

Trace Table Size

This table contains a history of GCS supervisor events. 16K is the default size. However, increasing the number of applications that you run also increases the activity of the GCS supervisor, which might require a larger trace table.

Max. Virtual Machines

This is the **maximum** number of virtual machines allowed in the group at one time. The number that you select depends upon how much space you have available in common storage. Remember that each group member occupies one control block in common storage.

System ID

This is the message that is displayed at the console of a virtual machine that IPLs this GCS saved system. You can define up to 130 characters.

Shared Segments

These are the additional saved segments (such as VTAM) that you want to access. They are linked automatically when you IPL the GCS saved system. Before you provide this information, you must find out what requirements these saved segments have.

Note: Do not include CMSBAM or CMSVSAM here. Follow the procedures described in Chapter 6, "Installing Saved Segments" on page 6-1, to build CMSBAM and CMSVSAM after you complete your base installation.

Planning for the Transparent Services Access Facility (TSAF)

TSAF is an optional system component that enables users to connect to and communicate with local or remote virtual machines within a **collection** of systems. With TSAF, a user can connect to a **resource** located in the collection by specifying a **resource ID** instead of specifying the user ID and node ID.

The sample CP directory contains an entry for the TSAF virtual machine (user ID TSAFVM). If you plan to install the TSAF component, you may want to tailor this directory entry to meet your specific requirements. In addition, you may need to consider the following:

- Setting up the TSAF message repository
- Establishing communication links for the TSAF virtual machine
- Setting up **resource manager** (also called **server**) virtual machines
- Assigning new user IDs (all user IDs on all systems in the TSAF collection must be unique)
- Defining resource IDs (all resource IDs in the TSAF collection must be unique).

For more information about TSAF, see *VM/SP Connectivity Planning, Administration, and Operation.*

Planning for APPC/VM VTAM Support (AVS)

AVS is an optional system component that runs in a GCS virtual machine. It provides the services that let a program communicate with other programs on VM and non-VM systems in an SNA network.

If you plan to install AVS, make sure that you define the AVSVM virtual machine as an authorized user ID in the GCS configuration file when you build GCS. You should also be familiar with the following items, which are described in detail in *VM/SP Connectivity Planning, Administration, and Operation*:

- Setting up the AVSVM virtual machine:
  - Modifying the AVS entry in the CP directory
  - Preparing to install and service the AVSVM virtual machine
  - Preparing the AVS message repository (optional)
  - Preparing the AVS accounting module (optional)
  - Adjusting the tuning parameters to improve performance (optional)
  - Setting up links through which VTAM can communicate.
- Assuring secure TSAF collection-to-SNA network communications by assigning unique gateway names
- Setting up the AVSVM virtual machine to be a gateway manager.

IBM supplies a non-executable source file, AGWACI ASSEMBLE, that generates three types of accounting records:

- Initialization records
- Conversation records
- Termination records.

You can modify this file or replace it with your own accounting module.

IBM supplies another non-executable source file, AGWTUN ASSEMBLE, that contains default tuning values. When doing system administration, you can modify the defined constants (DCs) in the AGWTUN ASSEMBLE file. See *VM/SP Connectivity Planning, Administration, and Operation* for a description of these parameters.

Note: Changing the default values could improve or *degrade* performance in your installation.

To change the IBM-supplied default values in AGWACI or AGWTUN (after you load the AVS code):

1. Edit the file (loaded to the AVS object directory) and modify the selected values.
2. Use the VMFASM EXEC to reassemble the file.
3. Use the VMFLKED EXEC to re-link the resulting TEXT file with the other AGW*xxx* TEXT files in a new AGW LOADLIB.
4. Copy the new AGW LOADLIB to the AVSVM 191 minidisk.

For more information about AVS, see *VM/SP Connectivity Planning, Administration, and Operation*. For more information about the VMFASM EXEC and the VMFLKED EXEC, see the *VM/SP Service Guide*.

Tailoring the System Definition Files

The system definition files are a group of files shipped on the VM/SP Product Tape that contain parameters used to generate your VM/SP system. During VM/SP installation, you must examine these files and tailor them to define your unique system configuration. This section outlines some general tailoring considerations. Specific tailoring instructions are provided in the VM/SP installation procedures (Chapter 3 and Chapter 4).

VM/SP Product Parameter File (5664167E $PPF)

The VM/SP product parameter file contains various types of system generation parameters, such as control file names, minidisk and SFS directory assignments, and build list (loadlist) names. You can accept the values specified in this file, or you can tailor the file to satisfy your own requirements.

For example, you can build your CP nucleus with certain combinations of the following special CP options:

- Small CP
- CP FRET Trap
- Virtual=Real.

The selection of these options is determined by the control file and build list (loadlist) used to build the nucleus, as specified in the product parameter file. In the base product parameter file shipped on the product tape, these options are **not** selected. To select any of these options, you must modify the file to specify the appropriate control file and/or build list. For additional information about these CP options, see Appendix D, "Special Options for CP" on page D-1.

For additional information about the structure and function of the product parameter file, see the *VM/SP Service Guide*.

CP Directory (VMUSERS DIRECT)

The CP directory identifies the virtual machines (users) permitted to log on to your system. Each virtual machine is assigned a unique user identification (user ID). As a **system security** measure, the sample CP directory defines the logon prevention password **NOLOG** for all user IDs except MAINT and OPERATOR. You cannot log on to any user ID that has a NOLOG password. In addition, the passwords defined for MAINT and OPERATOR are **restricted passwords**. All restricted passwords, including the supplied MAINT and OPERATOR passwords, are listed in a file named **RPWLIST DATA** that is loaded with the sample files. For a list of the restricted passwords, see Appendix B, "Restricted Logon Passwords" on page B-1.

The Starter System CP nucleus and the VM/SP CP nucleus include a feature called Auto-Deactivation of Restricted Passwords (ADRP). After you IPL (initial program load) the Starter System or build the new CP nucleus (or, for current VM users, if your current CP has the ADRP feature installed), issuing the DIRECT command to process the directory causes the system to search the directory for logon passwords that match the list in the RPWLIST DATA file. All passwords that match are changed to NOLOG in the directory before the directory is placed on line.

If you are using the sample directory to create your system, you must examine and modify this file to:

- Replace the supplied MAINT and OPERATOR logon passwords with unique non-restricted passwords
- Supply unique non-restricted logon passwords for all other user IDs to remove the NOLOG condition.

In addition, you might want to:

- Modify certain sample directory entries, such as GCS or TSAFVM
- Create entries for additional user IDs, such as RSCS.

CP System Control File (DMKSYS ASSEMBLE)

DMKSYS describes the CP system residence device (VMSRES), system storage size, CP-owned DASD, and other system parameters. For example, the SYSOPER keyword in the SYSOPR macro in the DMKSYS file identifies the system operator user ID. The default user ID is OPERATOR; this user ID appears in the installation procedure. However, if you want some other user ID to be the system operator, you must change this entry.

Real I/O Configuration File (DMKRIO ASSEMBLE)

DMKRIO describes the I/O (input/output) devices, control units, and channels attached to the processor. Because VM/SP uses this information to schedule I/O operations and allocate resources, you must make sure that the entries in this file accurately represent your system configuration.

System Name Table (DMKSNT ASSEMBLE)

DMKSNT defines the name, size, and virtual storage location of named saved systems and physical saved segments. If you plan to install any programs or products that have saved system or physical saved segment requirements, you must make sure that the DMKSNT file contains properly defined entries for them.

For example, if you are installing GCS, the SYSNAME parameter in the NAMESYS macro for the GCS nε ned saved system in the DMKSNT file must match the system name that you define in your GCS configuration file.

Table 6-1 on page 6-3 shows the suggested layout of saved systems and physical saved segments in virtual storage.

Forms Control Buffer File (DMKFCB ASSEMBLE)

DMKFCB contains real and virtual printer parameters. If you want to change the supplied forms control buffer macros, see *VM/SP Administration* for directions.

CMS Nucleus Generation Profile (DMSNGP ASSEMBLE)

DMSNGP defines responses to the system prompts that are generated when you build a CMS nucleus. If a response is defined in the DEFNUC macro call in this file, the system uses that value, and the corresponding prompt does not appear when you build the nucleus. If a DEFNUC statement is missing or does not contain a value (and the DEFNUC macro does not contain a default value), or if the DEFNUC statement contains a question mark (?), then the prompt does appear when you build the nucleus.

DMSNGP is supplied with most responses pre-defined. However, you can modify the file to tailor the responses to suit your installation. For more information about DMSNGP and the DEFNUC macro, see the *VM/SP Planning Guide and Reference.*

Migrating Spool Files

If you are doing a migration, and you have spool files that you want to migrate, you have two options:

- You can use the SPTAPE command to save the spool files on tape before you begin the migration procedure. For more information about this command, see the *VM/SP CP System Command Reference.*
- You can use the procedure described in Appendix E, "Migrating Spool Files Using SHUTDOWN/WARM IPL" on page E-1 if:
  - You are **not** using the Starter System procedure to do the migration.
  - If you are migrating from VM/SP Release 4 or later.
  - Your system residence volume is **not** on a 3380-E4 or 3380-K4 DASD.

  This procedure has other restrictions and requirements that you must satisfy before you begin the VM/SP migration. The procedure is intended for experienced VM system programmers in installations that explicitly control their own minidisk layout for CP-owned DASD.

Installing National Languages on Your VM/SP System

VM/SP is shipped with mixed-case American English as the default national language. However, the product tape also contains files for uppercase English. When you install VM/SP, you can install either American English (by default) or uppercase English as the system national language. When VM/SP is completely installed, the system national language is automatically set for CP and all virtual machines on the system. When users log on, they receive messages, see panels, and enter CMS commands in the system national language.

You can also order other national languages to install on your system. National languages (except American English and uppercase English) are distributed on national language feature tapes. The files on a national language feature tape contain translated information.

There are two ways to install another national language:

- As the new system national language.

  Then, when users log on, they receive most messages, see some panels, and enter most CMS commands in the new language.

  In this type of language installation, the language files are used to create new system nuclei. The procedure is a supplement to the base installation procedure, and is described in Chapter 7, "Installing a New System National Language" on page 7-1.

  Notes:

  1. The extent of the translated information could vary between national languages. The language of some messages, panels, and commands is always American English, regardless of the language installed as the system national language.
  2. The language of the installation and system generation tools is American English and cannot be changed.

- As an option available to system users enabling them to use that language instead of the system national language.

  You can install several national languages in this manner, and you can set the languages that are available for individual users. This type of language installation uses a saved segment to hold the new language information, and must be performed **after** your base system is installed. For more information about loading language files into a saved segment, see *VM System Facilities for Programming*.

Installation Reference Worksheet

DASD REQUIREMENTS

Volume Label	Temporary Label	Volume Number	Device Type (*devtype*)	Real Address (*rdev*)	Virtual Address (*vdev*)
VMSRES	SPSYS				
VMPK01	SPSYS				
VMPK02[1]	SPSYS				
VMPK04	SPSYS				

[1] Required only for 9313 DASD

STARTER SYSTEM CONFIGURATION

Device	Default Address	Default Device Type	Real Address	Virtual Address	Device Type	Notes
Console	014	3148				You are not prompted for this device.
Printer	00E	1403				Recommended.
Punch	00D	2450P				Optional.
Reader	00C	2450R				Optional.
First Tape Drive	181	3420				Required.
Second Tape Drive	182	3420				Optional.
VMPK01 Work Pack	151	3380				Required for all DASD types.
Extra Graphic Device	020	3277				Optional.

Note: The Starter System is a limited function system and accepts only 4-bit addresses (X'*nn*7' or less) for each DASD, tape drive, and graphic device used to restore the Starter System from tape.

GCS CONFIGURATION FILE

	System #1	System #2	System #3
System Name:			
Authorized User IDs:			
System Disk vdev:	595[1]	595[1]	595[1]
System Disk Ext. vdev:	59E[1]	59E[1]	59E[1]
Common Dump Receiver:			
Recovery Machine:	GCS[2]		
Trace Table Size:	16K[1]	16K[1]	16K[1]
Max. Virtual Machines:			
System ID:			
Shared Segments:			

[1] GROUP EXEC defaults
[2] Entry supplied in sample CP directory

Chapter 3. Installing VM/SP Using the Starter System

This chapter describes the step-by-step procedure for using the Starter System to install VM/SP. If you are doing an initial installation, whether first level or second level, you must use this procedure. If you are migrating from a VM/SP release prior to VM/SP Release 4, you must use this procedure. If you are migrating from VM/SP Release 4 or VM/SP Release 5, this procedure is recommended.

The system messages, responses, and prompts shown in the following steps are intended to be **representative** of what you see on your display when installing VM/SP **on a 3380 DASD**. Depending on your actual system configuration, you might receive additional or fewer messages, responses, and prompts, or you might receive ones that are slightly different from those shown. Important variations are indicated in the text or in the comments column.

This procedure assumes that:

- You have read Chapters 1 and 2 of this book
- You understand the principles of VM/SP installation
- You understand the page layout and text formats used in this book
- You have done the necessary planning for your installation and completed the Installation Reference Worksheet.

The sequence of steps in the Starter System procedure is outlined in Table 3-1.

Notes:

1. This procedure is designed for uniprocessor (UP) installations but can also be used for attached processor (AP) and multiprocessor (MP) installations. Additional requirements for AP and MP installations are noted in the text.
2. The stand-alone programs used in this procedure might not work on processors with the Loop Adapter feature enabled.
3. Certain prompts and responses in this procedure use the term "FB-512" when referring to a fixed-block architecture (FBA) direct access storage device (DASD).

Table 3-1 (Page 1 of 2). Starter System Procedure to Install VM/SP

Step	Description
(prep)	Complete the Installation Reference Worksheet
1	Format your DASD volumes
2	Restore the Starter System from the Starter System Tape
3	IPL and define the Starter System
4	Load the installation tools and profiles from the VM/SP Product Tape
5	Load the system generation tools, system tools, CP code, and system definition files
6	Tailor the system definition files
7	Build the new CP nucleus
8	Shut down the Starter System CP and IPL the new CP

Table 3-1 (Page 2 of 2). Starter System Procedure to Install VM/SP

Step	Description
9	Allocate the remaining system volumes
10	Shut down and re-IPL
11	Load CMS and Procedures Language/VM; build and save CMS
12	Load IPCS and complete the CUSTOMER PROFILE
13	Format the remaining minidisks in the base CP directory
14	Build the system file pool
15	Build the user file pool (recommended)
16	Load and save HELP (optional)
17	Load, build, and save GCS (optional)
18	Load TSAF (optional)
19	Load AVS (optional)
20	Load source code (optional)
21	Update the SYSTEM NETID file (optional)
22	Install EREP (optional)
23	Do a full system backup (recommended)
(sup)	Verify your installed VM/SP system (recommended)
(sup)	Install saved segments (optional)
(sup)	Install a new system national language (optional)
(sup)	Install additional national languages (optional)
(sup)	Install optional feature program products (optional)
(sup)	Service your VM/SP system (optional)

Migration Considerations

If you are migrating from a VM/SP release prior to VM/SP Release 4, you **must** use this procedure. If you are migrating from VM/SP Release 4 or VM/SP Release 5, you can use this procedure (**recommended**) or the procedure described in Chapter 4, "Installing VM/SP Using an Existing VM/SP System."

This VM/SP release includes substantial changes from previous VM/SP releases. Chapters 1 and 2 of this book highlight the changes that relate to installation and system generation. For more information about the additions, enhancements, and migration considerations for this release, see the *VM/SP Release 6 Guide*.

Also consider the following, where appropriate:

- Current VM/SP Release 2 customers must not use a 3278 Model 5, 3279 Model 3, or any terminal that supports greater than 24 x 80 screen sizes.
- Current VM/SP Release 3 customers with service level less than SL304 must not use a 3278 Model 5, 3279 Model 3, or any terminal that supports greater than 24 x 80 screen sizes.

Starter System Installation Procedure Checklist

Before you begin Step 1, make sure that you have:

__ A completed Installation Reference Worksheet

__ At least 1M available real storage

__ The required type and number of DASD volumes

__ At least one tape drive

__ The Starter System Tape that matches your VMSRES device type

__ The VM/SP Product Tape

__ The source feature tape, if included with your order

__ The EREP distribution tape, if included with your order

__ A tape on which to dump the stand-alone programs

__ At least one additional tape for system backup

__ A feature tape for a new system national language, if included with your order

__ (For 308x Processor Complex users only) An adequately defined input/output configuration data set (see the *VM/SP Planning Guide and Reference*).

Step 1. Format Your DASD Volumes

Overview

In Chapter 2, "Planning Your VM/SP Installation," you determined your DASD requirements. The first step in the installation procedure is to format the DASD volumes that you are going to use.

The first tape file on the Starter System Tape contains two stand-alone utilities:

- Device Support Facility
- Format/Allocate program.

In this step, you:

- Load the Device Support Facility and use it to initialize new DASD or to perform other DASD-related operations, such as:
  - Preparing a new device or a replacement head/disk assembly (HDA) for use
  - Initializing minidisks
  - Investigating suspected drive problems
  - Checking disk surfaces
  - Assigning alternate tracks
  - Printing a track assignment map
  - Emulating a CKD device on an FBA device
  - Changing volume serial numbers and owner identification
  - Adding IPL text to volumes
  - Converting a volume to indexed format.

  Many of these operations are intended for an OS or DOS environment. However, you can also use the Device Support Facility in a VM environment to diagnose problems that might be related to your DASD. This program has diagnostic capabilities that Format/Allocate, DDR, and other DASD utilities do not. For more information about the Device Support Facility, see the *Device Support Facility User's Guide and Reference.*

- Load the Format/Allocate program and use it to format the DASD volumes you are using for VMSRES, VMPK01, VMPK02 (9313 DASD only), and VMPK04 (only if you plan to load source). Refer to your Installation Reference Worksheet for volume addresses.

  The allocation of space on the VMSRES volume is determined by the Starter System version that you are using, which must match the device type of your VMSRES DASD. (When using the 9313/9332 Starter System, the allocation of the VMPK01 volume is also supplied by the Starter System.) The other volumes are allocated in Step 9.

First Level and Second Level Considerations

Formatting a large DASD at first level can take 30 minutes or more. At second level, due to increased system overhead, the formatting operation can take even longer. If possible, format your DASD at first level.

Note: The Format/Allocate program causes high channel utilization that could cause delays for other users on the same channel.

In addition, if you plan to restore an FBA Starter System (9313/9332 or 9335/3370) at second level (in Step 2), your first level FBA device must be either dedicated or defined as a full-pack minidisk.

At second level, if your system volumes are full-pack minidisks with the same labels as the first level volumes (such as VMSRES), you should make sure that your second level volumes are not attached during any IPL of the first level system.

To do this step at **first level**, go to "Procedure, First Level" on page 3-6.

To do this step at **second level**, go to "Procedure, Second Level" on page 3-9.

Procedure, First Level

- Ready the DASD volumes that you are going to use.
- Mount and ready the Starter System Tape.
- IPL (initial program load) the tape drive (according to the directions in the *Operator's Guide* for your machine) to load the Device Support Facility.

  Note: After you IPL, your system might enter the WAIT state (SYSTEM light off, WAIT light on, system inactive). To exit the WAIT state, press ATTN or ENTER. To determine the correct key(s) to press to obtain these functions on your device, see Table 1-3 on page 1-14.

```
ICK005E DEFINE INPUT DEVICE, REPLY
'DDDD,CUU' OR 'CONSOLE'
ENTER INPUT/COMMAND:
```

This message indicates that the Device Support Facility is loaded and ready. If you want to use this program, follow the instructions in the *Device Support Facility User's Guide and Reference*. If you do not want to use this program, go on to the next operation (■).

- IPL the tape drive to load the Format/Allocate program.

  Note: After you IPL, your system might enter the WAIT state (SYSTEM light off, WAIT light on, system inactive). The Device Support Facility might still be displayed on your screen. To exit the WAIT state and clear the Device Support Facility, press ATTN or ENTER.

```
VM/370 FORMAT/ALLOCATE PROGRAM - VM/SP
ENTER FORMAT OR ALLOCATE:
```

This message indicates that Format/Allocate is loaded and ready.

- Complete the following sequence for each volume that you need (as indicated on your Worksheet):

  Note: The Format/Allocate program causes high channel utilization that could cause delays for other users on the same channel.

format

```
FORMAT FUNCTION SELECTED
ENTER DEVICE ADDRESS (CUU):
```

rdev

rdev is the real address of the volume that you are formatting.

```
ENTER DEVICE TYPE:
```

devtype

devtype is the device type of the DASD. Enter **3350**, **3375**, **3380**, or **fb-512** (for the 3370, 9313, 9332, or 9335).

```
ENTER START  PAGE|CYLINDER  NUMBER OR "LABEL":
```

ENTER

The default start page/cylinder is **0**.

```
ENTER END  PAGE|CYLINDER  NUMBER:
```

ENTER

The default end page/cylinder depends on the device type and model. The program can tell the difference between models of the same device type, such as 3380 and 3380-E4. See the following table.

Device Model	End Page/Cylinder
3350	cyl. 554
3370	page 69749
3370-2	page 89093
3375	cyl. 958
3380	cyl. 884
3380-E4	cyl. 1769
3380-K4	cyl. 2654
9313	page 30779
9332	page 45003
9332-600/602	page 69349
9335	page 100588

```
ENTER DEVICE LABEL:
```

spsys

spsys is a temporary label that you assign to each volume that you format, so the volume does not become CP-owned. A volume can become CP-owned if the label is listed in the SYSOWN macro in the DMKSYS file (SPSYS is not listed). If the volume becomes CP-owned, you cannot gain access to it in later installation operations.

The system residence volume is relabeled with its permanent label in Step 2; the other volumes are relabeled with their permanent labels in Step 9.

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

This prompt appears only for 33*nn* CKD DASD (3350, 3375, and 3380).

ENTER (or **no**) or **yes**

The default is **no**. Explicit write verification is usually not necessary unless the DASD is newly installed or has undergone maintenance. If you decide to perform write verification, formatting time is increased. For more information about write verification, see *VM/SP System Facilities for Programming*.

```
FORMAT STARTED
```

Note: Formatting a large DASD can take 30 minutes or more. During this time, the SYSTEM light is on, but there is no screen activity. A message tells you when the formatting is done. If a problem occurs during the formatting process and you cannot proceed, go back to the beginning of this step.

WARNING

Unless a problem occurs, do not make **any** entry until you receive the FORMAT DONE message, or you might cause the program to issue the message before the formatting is complete.

```
FORMAT DONE
```

```
WRITE VERIFICATION WAS NOT PERFORMED
0000 NO. PAGE RECORDS WITH READ-CHECK ERRORS
```

These messages are not displayed if you do write verification. If the second message indicates any errors, stop the installation procedure and contact your Systems Engineer or hardware service personnel. You might have to go back and initialize the DASD volume with the Device Support Facility. See the *Device Support Facility User's Guide and Reference*.

```
ENTER FORMAT OR ALLOCATE:
```

- Repeat the previous sequence to format each volume that you need.
- When you complete the formatting, go to "Step 2. Restore the Starter System" on page 3-13.

Procedure, Second Level

WARNING

If your system volumes are full-pack minidisks with the same labels as the first level volumes (such as VMSRES), you should make sure that your second level volumes are not attached during any IPL of the first level system.

- Attach the required DASD to your virtual machine (the system operator might have to do this for you). If you have not already done so, record the virtual addresses in the "DASD Requirements" table on your Installation Reference Worksheet.
- Attach a tape drive to your virtual machine and mount the Starter System Tape (the system operator might have to do this for you).

  If possible, attach the tape drive at the default virtual address, 181. Otherwise, record the virtual address that you are using in the "Starter System Configuration" table on your Installation Reference Worksheet.
- IPL (initial program load) the tape drive to load the Device Support Facility:

ATTN

To determine the correct key(s) to press to obtain the ATTN and ENTER functions on your device, see Table 1-3 on page 1-14.

ipl ***vdev***

vdev is the virtual address of the tape drive.

Note: After you IPL, if your system appears to be inactive, you might be in the WAIT state. To exit the WAIT state, press ATTN, then press ENTER.

```
ICK005E DEFINE INPUT DEVICE, REPLY
'DDDD,CUU' OR 'CONSOLE'
ENTER INPUT/COMMAND:
```

This message indicates that the Device Support Facility is loaded and ready. If you want to use this program, follow the instructions in the *Device Support Facility User's Guide and Reference*. If you do not want to use this program, go on to the next operation (■).

- IPL the tape drive to load the Format/Allocate program:

ATTN

ipl ***vdev***

vdev is the virtual address of the tape drive.

Note: After you IPL, your system might enter the WAIT (inactive) state. The Device Support Facility might still be displayed on your screen. To exit the WAIT state and clear the Device Support Facility, press ATTN or ENTER.

```
VM/370 FORMAT/ALLOCATE PROGRAM - VM/SP
ENTER FORMAT OR ALLOCATE:
```

This message indicates that Format/Allocate is loaded and ready.

- Complete the following sequence for each volume that you need (as indicated on your Worksheet):

Note: The Format/Allocate program causes high channel utilization that could cause delays for other users on the same channel.

format

```
FORMAT FUNCTION SELECTED
ENTER DEVICE ADDRESS (CUU):
```

vdev

vdev is the virtual address of the volume that you are formatting.

```
ENTER DEVICE TYPE:
```

devtype

devtype is the device type of the DASD. Enter **3350**, **3375**, **3380**, or **fb-512** (for the 3370, 9313, 9332, or 9335).

```
ENTER START  PAGE|CYLINDER  NUMBER OR "LABEL":
```

ENTER

The default start page/cylinder is **0**.

```
ENTER END  PAGE|CYLINDER  NUMBER:
```

ENTER

The default end page/cylinder depends on the device type and model. The program can tell the difference between models of the same device type, such as 3380 and 3380-E4. See the following table.

Device Model	End Page/Cylinder
3350	cyl. 554
3370	page 69749
3370-2	page 89093
3375	cyl. 958
3380	cyl. 884
3380-E4	cyl. 1769
3380-K4	cyl. 2654
9313	page 30779
9332	page 45003
9332-600/602	page 69349
9335	page 100588

```
ENTER DEVICE LABEL:
```

spsys

spsys is a temporary label that you assign to each volume that you format, so the volume does not become CP-owned. A volume can become CP-owned if the label is listed in the SYSOWN macro in the DMKSYS file (SPSYS is not listed). If the volume becomes CP-owned, you cannot gain access to it in later installation operations.

The system residence volume is relabeled with its permanent label in Step 2; the other volumes are relabeled with their permanent labels in Step 9.

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

This prompt appears only for 33*nn* CKD DASD (3350, 3375, and 3380).

ENTER (or **no**) or **yes**

The default is **no**. Explicit write verification is usually not necessary unless the DASD is newly installed or has undergone maintenance. If you decide to perform write verification, formatting time is increased. For more information about write verification, see *VM/SP System Facilities for Programming.*

```
FORMAT STARTED
```

Note: Formatting a large DASD can take 30 minutes or more. During this time, there is no screen activity. A message tells you when the formatting is done. If a problem occurs during the formatting process and you cannot proceed, go back to the beginning of this step.

WARNING

Unless a problem occurs, do not make **any** entry until you receive the FORMAT DONE message, or you might cause the program to issue the message before the formatting is complete.

```
FORMAT DONE
```

```
WRITE VERIFICATION WAS NOT PERFORMED
0000 NO. PAGE RECORDS WITH READ-CHECK ERRORS
```

These messages are not displayed if you do write verification. If the second message indicates any errors, stop the installation procedure and contact your Systems Engineer or hardware service personnel. You might have to go back and initialize the DASD volume with the Device Support Facility. See the *Device Support Facility User's Guide and Reference.*

```
ENTER FORMAT OR ALLOCATE:
```

- Repeat the previous sequence to format each volume that you need.
- When you complete the formatting, go to “Step 2. Restore the Starter System” on page 3-13.

Step 2. Restore the Starter System

Overview

The second and third tape files on the Starter System Tape contain the stand-alone DASD Dump/Restore (DDR) program and the Starter System, respectively. In this step, you load DDR and use it to restore the Starter System from the Starter System Tape to the system residence volume that you formatted in Step 1.

Note: Restoring the Starter System to a 9313 or 9332 DASD requires two volumes (VMSRES and VMPK01).

First Level and Second Level Considerations

To do this step at **first level**, go to "Procedure, First Level" on page 3-14.

To do this step at **second level**, go to "Procedure, Second Level" on page 3-19.

Procedure, First Level

- IPL the tape drive to load the DASD Dump/Restore (DDR) program.

  Note: After you IPL, your system might enter the WAIT state (SYSTEM light off, WAIT light on, system inactive). The Format/Allocate program might still be displayed on your screen. To exit the WAIT state and clear the Format/Allocate program, press ATTN or ENTER.

```
VM/370 DASD DUMP/RESTORE PROGRAM - VM/SP
ENTER CARD READER ADDRESS OR CONTROL STATEMENTS
ENTER:
```

This message indicates that DDR is loaded and ready.

- Restore the Starter System by entering the following commands:

sysprint cons

This command directs system output to the console.

```
ENTER:
```

input *rdev* *devtype* (*option*

rdev is the real address of the tape drive.

devtype is the device type of the tape drive. Enter **2401**, **2415**, **2420**, **3420**, **3422**, **3430**, **3480**, **8809**, or **9347**.

option controls the position of the tape, and depends on the device type of the DASD:

- For the 9313 or 9332 DASD, enter **leave** to leave the tape positioned at the end of the file after restoring the first DASD volume.
- For all other DASD types, enter **rewind** to rewind the tape without unloading.

```
ENTER:
```

output *rdev* *devtype* scratch

rdev is the real address of your system residence volume. (On 9313 or 9332 DASD, this is the first of two volumes.)

devtype is the device type of the DASD. Enter **3350**, **3375**, **3380**, or **fb-512** (for the 3370, 9313, 9332, or 9335).

scratch suppresses label checking by the DDR program. You assigned a temporary SPSYS label to this volume in Step 1; DDR restores the permanent VMSRES label from the tape along with the rest of the volume.

```
ENTER:
```

restore all

```
DMKDDR725R DASD input was (is) larger than output device.
Do you want to continue? Respond yes or no.
```

This prompt appears only for the 9313 DASD.

yes

```
RESTORING VMSRES
DATA DUMPED mm/dd/yy AT hh.mm.ss GMT FROM VMSRES RESTORED TO SCRATCH
INPUT CYLINDER EXTENTS      OUTPUT CYLINDER EXTENTS
        START     STOP      START     STOP
        nnnn      nnnn      nnnn      nnnn
        nnnn      nnnn      nnnn      nnnn
        nnnn      nnnn      nnnn      nnnn
        nnnn      nnnn      nnnn      nnnn
END OF RESTORE
BYTES RESTORED nnnnnnnnnn
```

The start and stop cylinder/block extents vary, depending on the Starter System device type. The block extents might not be displayed if the console is in DISPLAY mode.

```
ENTER:
```

9313 and 9332 DASD Only

output _rdev_ fb-512 scratch

rdev is the real address of your VMPK01 volume.

```
ENTER:
```

restore all

```
DMKDDR725R DASD input was (is) larger than output device.
Do you want to continue? Respond yes or no.
```

This prompt appears only for the 9313 DASD.

yes

```
RESTORING VMPK01
DATA DUMPED mm/dd/yy AT hh.mm.ss GMT FROM VMPK01 RESTORED TO SCRATCH
INPUT CYLINDER EXTENTS      OUTPUT CYLINDER EXTENTS
        START     STOP      START     STOP
        nnnn      nnnn      nnnn      nnnn
        nnnn      nnnn      nnnn      nnnn
        nnnn      nnnn      nnnn      nnnn
        nnnn      nnnn      nnnn      nnnn
END OF RESTORE
BYTES RESTORED nnnnnnnnnn
```

The start and stop cylinder/block extents vary, depending on the Starter System device type. The block extents might not be displayed if the console is in DISPLAY mode.

```
ENTER:
```

End of 9313 and 9332 DASD Only

ENTER

This entry ends the DDR program.

```
END OF JOB
```

Restarting the Restore Operation

If you need to restart the restore operation:

1. Rewind the Starter System Tape.
2. IPL the tape drive **two times** to bypass the Device Support Facility and the Format/Allocate program.
3. Go back to the beginning of this step.

9313, 9332-600/602, 9335, and 3370-2 DASD Only

The 9313/9332 Starter System is supplied with an allocation map for the 9332-400/402 models. The 9335/3370 Starter System is supplied with an allocation map for the 3370 model. If you restored the Starter System to a 9313, 9332-600/602, 9335, or 3370-2, you must do a partial allocation to correct the allocation map:

- Rewind the Starter System Tape.
- IPL the tape drive **two times** to bypass the Device Support Facility and load the Format/Allocate program:

```
VM/370 FORMAT/ALLOCATE PROGRAM - VM/SP
ENTER FORMAT OR ALLOCATE:
```

allocate

```
ALLOCATE FUNCTION SELECTED
ENTER DEVICE ADDRESS (CUU):
```

rdev

rdev is the real address of the VMSRES volume.

```
ENTER DEVICE TYPE:
```

fb-512

```
ENTER DEVICE LABEL:
```

vmsres

```
ENTER ALLOCATION DATA FOR VOLUME VMSRES
TYPE PAGE PAGE
```

tdsk *startt* *endt*
perm *startp* *endp*
end

Enter the allocation data for your VMSRES device from the following table:

Device	*startt*	*endt*	*startp*	*endp*
9313	28138	30035	30036	30779
9332-600/602	28138	30035	30036	69349
3370-2	53769	57563	57564	89093
9335	53484	57278	57279	100588

```
ALLOCATION RESULTS
PERM          nnnnn   nnnnn
TEMP          nnnnn   nnnnn
DRCT          nnnnn   nnnnn
PERM          nnnnn   nnnnn
TDSK          nnnnn   nnnnn
PERM          nnnnn   nnnnn
DEVICE rdev VOLUME VMSRES    ALLOCATION ENDED
ENTER FORMAT OR ALLOCATE:
```

End of 9313, 9332-600/602, 9335, and 3370-2 DASD Only

9313 and 9332-600/602 DASD Only

allocate

```
ALLOCATE FUNCTION SELECTED
ENTER DEVICE ADDRESS (CUU):
```

rdev

rdev is the real address of the VMPK01 volume.

```
ENTER DEVICE TYPE:
```

fb-512

```
ENTER DEVICE LABEL:
```

vmpk01

```
ENTER ALLOCATION DATA FOR VOLUME VMPK01
TYPE PAGE PAGE
```

tdsk ***startt*** ***endt***
perm ***startp*** ***endp***
end

Enter the allocation data for your VMPK01 device from the following table:

Device	*startt*	*endt*	*startp*	*endp*
9313	20267	21984	21985	30779
9332-600/602	18917	20634	20635	69349

```
ALLOCATION RESULTS
PERM         nnnnn   nnnnn
TEMP         nnnnn   nnnnn
DRCT         nnnnn   nnnnn
PERM         nnnnn   nnnnn
TDSK         nnnnn   nnnnn
PERM         nnnnn   nnnnn
DEVICE rdev VOLUME VMPK01    ALLOCATION ENDED
ENTER FORMAT OR ALLOCATE:
```

End of 9313 and 9332-600/602 DASD Only

- ■ When you complete the restore, go to "Step 3. IPL and Define the Starter System" on page 3-24.

Procedure, Second Level

- IPL the tape drive to load the DASD Dump/Restore (DDR) program:

ATTN

ipl *vdev*

vdev is the virtual address of the tape drive.

Note: After you IPL, your system might enter the WAIT (inactive) state. The Format/Allocate program might still be displayed on your screen. To exit the WAIT state and clear the Format/Allocate program, press ATTN, then press ENTER.

```
VM/370 DASD DUMP/RESTORE PROGRAM - VM/SP
ENTER CARD READER ADDRESS OR CONTROL STATEMENTS
ENTER:
```

This message indicates that DDR is loaded and ready.

- Restore the Starter System by entering the following commands:

sysprint cons

This command directs system output to the virtual console.

```
ENTER:
```

input *vdev* *devtype* **(** *option*

vdev is the virtual address of the tape drive.

devtype is the device type of the tape drive. Enter **2401**, **2415**, **2420**, **3420**, **3422**, **3430**, **3480**, **8809**, or **9347**.

option controls the position of the tape, and depends on the device type of the DASD:

- For the 9313 or 9332 DASD, enter **leave** to leave the tape positioned at the end of the file after restoring the first DASD volume.
- For all other DASD types, enter **rewind** to rewind the tape without unloading.

```
ENTER:
```

output ***vdev*** ***devtype*** **scratch**

vdev is the virtual address of your system residence volume. (On 9313 or 9332 DASD, this is the first of two volumes.)

devtype is the device type of the DASD. Enter **3350, 3375, 3380,** or **fb-512** (for the 3370, 9313, 9332, or 9335).

scratch suppresses label checking by the DDR program. You assigned a temporary SPSYS label to this volume in Step 1. DDR restores the permanent VMSRES label from the tape along with the rest of the volume.

```
ENTER:
```

restore all

```
DMKDDR725R DASD input was (is) larger than output device.
Do you want to continue? Respond yes or no.
```

This prompt appears only for the 9313 DASD.

yes

```
RESTORING VMSRES
DATA DUMPED mm/dd/yy AT hh.mm.ss GMT FROM VMSRES RESTORED TO SCRATCH
INPUT CYLINDER EXTENTS      OUTPUT CYLINDER EXTENTS
       START     STOP       START     STOP
       nnnn      nnnn       nnnn      nnnn
       nnnn      nnnn       nnnn      nnnn
       nnnn      nnnn       nnnn      nnnn
       nnnn      nnnn       nnnn      nnnn
END OF RESTORE

BYTES RESTORED nnnnnnnnnn
```

The start and stop cylinder/block extents vary, depending on the Starter System device type. The block extents might not be displayed if the console is in DISPLAY mode.

```
ENTER:
```

9313 and 9332 DASD Only

output ***vdev*** **fb-512 scratch**

vdev is the virtual address of your VMPK01 volume.

```
ENTER:
```

restore all

```
DMKDDR725R DASD input was (is) larger than output device.
Do you want to continue? Respond yes or no.
```

This prompt appears only for the 9313 DASD.

yes

```
RESTORING VMPK01
DATA DUMPED mm/dd/yy AT hh.mm.ss GMT FROM VMPK01 RESTORED TO SCRATCH
INPUT CYLINDER EXTENTS      OUTPUT CYLINDER EXTENTS
        START     STOP      START     STOP
        nnnn      nnnn      nnnn      nnnn
        nnnn      nnnn      nnnn      nnnn
        nnnn      nnnn      nnnn      nnnn
        nnnn      nnnn      nnnn      nnnn
END OF RESTORE
BYTES RESTORED nnnnnnnnnn
```

The start and stop cylinder/block extents vary, depending on the Starter System device type. The block extents might not be displayed if the console is in DISPLAY mode.

```
ENTER:
```

End of 9313 and 9332 DASD Only

ENTER

This entry ends the DDR program.

```
END OF JOB
```

Restarting the Restore Operation

If you need to restart the restore operation:

1. Rewind the Starter System Tape.
2. IPL the tape drive **two times** to bypass the Device Support Facility and the Format/Allocate program.
3. Go back to the beginning of this step.

9313, 9332-600/602, 9335, and 3370-2 DASD Only

The 9313/9332 Starter System is supplied with an allocation map for the 9332-400/402 models. The 9335/3370 Starter System is supplied with an allocation map for the 3370 model. If you restored the Starter System to a 9313, 9332-600/602, 9335, or 3370-2, you must do a partial allocation to correct the allocation map:

- Rewind the Starter System Tape.
- IPL the tape drive **two times** to bypass the Device Support Facility and load the Format/Allocate program:

```
VM/370 FORMAT/ALLOCATE PROGRAM - VM/SP
ENTER FORMAT OR ALLOCATE:
```

allocate

```
ALLOCATE FUNCTION SELECTED
ENTER DEVICE ADDRESS (CUU):
```

vdev

vdev is the virtual address of the VMSRES volume.

```
ENTER DEVICE TYPE:
```

fb-512

```
ENTER DEVICE LABEL:
```

vmsres

```
ENTER ALLOCATION DATA FOR VOLUME VMSRES
TYPE PAGE PAGE
```

tdsk *startt* *endt*
perm *startp* *endp*
end

Enter the allocation data for your VMSRES device from the following table:

Device	***startt***	***endt***	***startp***	***endp***
9313	28138	30035	30036	30779
9332-600/602	28138	30035	30036	69349
3370-2	53769	57563	57564	89093
9335	53484	57278	57279	100588

```
ALLOCATION RESULTS
PERM         nnnnn   nnnnn
TEMP         nnnnn   nnnnn
DRCT         nnnnn   nnnnn
PERM         nnnnn   nnnnn
TDSK         nnnnn   nnnnn
PERM         nnnnn   nnnnn
DEVICE vdev VOLUME VMSRES    ALLOCATION ENDED
ENTER FORMAT OR ALLOCATE:
```

End of 9313, 9332-600/602, 9335, and 3370-2 DASD Only

9313 and 9332-600/602 DASD Only

allocate

```
ALLOCATE FUNCTION SELECTED
ENTER DEVICE ADDRESS (CUU):
```

rdev

rdev is the real address of the VMPK01 volume.

```
ENTER DEVICE TYPE:
```

fb-512

```
ENTER DEVICE LABEL:
```

vmpk01

```
ENTER ALLOCATION DATA FOR VOLUME VMPK01
TYPE PAGE PAGE
```

tdsk *startt* *endt*
perm *startp* *endp*
end

Enter the allocation data for your VMPK01 device from the following table:

Device	***startt***	***endt***	***startp***	***endp***
9313	20267	21984	21985	30779
9332-600/602	18917	20634	20635	69349

```
ALLOCATION RESULTS
PERM          nnnnn   nnnnn
TEMP          nnnnn   nnnnn
DRCT          nnnnn   nnnnn
PERM          nnnnn   nnnnn
TDSK          nnnnn   nnnnn
PERM          nnnnn   nnnnn
DEVICE vdev VOLUME VMPK01    ALLOCATION ENDED
ENTER FORMAT OR ALLOCATE:
```

End of 9313 and 9332-600/602 DASD Only

- ■ When you complete the restore, go to "Step 3. IPL and Define the Starter System" on page 3-24.

Step 3. IPL and Define the Starter System

Overview

In this step, you:

- IPL the Starter System that you restored to VMSRES in Step 2
- Answer a series of prompts to define your Starter System configuration
- Initialize the Starter System.

First and Second Level Considerations

Contention for service by the devices on shared control units could cause an IPL in a second level environment to take considerably longer than a similar operation at first level.

To do this step at **first level**, go to "Procedure, First Level" on page 3-25.

To do this step at **second level**, go to "Procedure, Second Level" on page 3-30.

Procedure, First Level

WARNING

> If you have control units that share more than 16 devices and are switchable to a second (non-IPL) processor, you should take the following precautions:
>
> 1. Put the channel-interface-enable switch on the non-IPL processor in the DISABLE position.
> 2. Before you IPL the new system, place any other loosely-coupled processor (via channels) in the STOP mode.

■ IPL CLEAR the VMSRES volume (according to the directions in the *Operator's Guide* for your machine).

Note: After you IPL, your system might enter the WAIT state (SYSTEM light off, WAIT light on, system inactive). The DDR program might still be displayed on your screen. To exit the WAIT state and clear the DDR program, press ATTN or ENTER.

```
VM/SP: STARTER SYSTEM
```

■ Define your Starter System configuration by answering the following series of prompts.

The Starter System provides default addresses and device types. To accept the Starter System default for a device prompt, just press the ENTER key.

You do not need a real printer, card reader, punch, second tape drive, or additional graphic device to generate the system. However, the system provides a default virtual equivalent for each of these devices. If you do not have these devices, press ENTER to accept the Starter System defaults.

If you do not want to accept a Starter System default, enter the real address (***rdev***) or device type (***devtype***). Refer to your Installation Reference Worksheet to help you complete this section.

Re-IPL Only

```
*** DO YOU WISH TO RE-DEFINE YOUR SYSTEM *** (YES,NO):
```

This prompt does not appear during the initial IPL of the Starter System. It appears only during a re-IPL.

yes or **no**

If you are re-IPLing the Starter System, and the device addresses are not specified correctly or have been changed, reply **yes** and answer the device prompts to redefine your system.

Otherwise, reply **no** and skip to the next operation (■) on page 3-28 to initialize the Starter System.

End of Re-IPL Only

```
ENTER PRINTER ADDRESS (CUU):
```

ENTER or *rdev*

00E is the default.

```
ENTER DEVICE TYPE (1403,1443,3203,3211,3262,3289E,4245,4248):
```

ENTER or *devtype*

1403 is the default.

```
ENTER PUNCH ADDRESS (CUU):
```

ENTER or *rdev*

00D is the default.

```
ENTER DEVICE TYPE (2540P,3525):
```

ENTER or *devtype*

2540P is the default.

```
ENTER READER ADDRESS (CUU):
```

ENTER or *rdev*

00C is the default.

```
ENTER DEVICE TYPE (2540R,2501,3505):
```

ENTER or *devtype*

2540R is the default.

```
ENTER ADDRESS WHERE FIRST TAPE IS MOUNTED (CUU):
```

ENTER or *rdev*

This is the address of your Starter System Tape. **181** is the default.

```
ENTER DEVICE TYPE (3420,2415,2420,2401,3422,3430,3480,8809,9347):
```

ENTER or *devtype*

3420 is the default.

```
ENTER ADDRESS OF A SECOND TAPE DRIVE (CUU):
```

ENTER or *rdev*

Do not specify the same address as the first tape drive. **182** is the default.

```
ENTER DEVICE TYPE (3420,2415,2420,2401,3422,3430,3480,8809,9347):
```

ENTER or *devtype*

3420 is the default.

```
ENTER DEVICE ADDRESS OF WORK PACK (CUU):
```

ENTER or *rdev*

This is the address of your VMPK01 volume (currently labeled SPSYS). **151** is the default.

```
ENTER DEVICE TYPE (3380,3330,3340,3350,3375,FB-512):
```

ENTER or *devtype*

3380 is the default.

```
ENTER ADDRESS OF A GRAPHIC DEVICE (CUU):
```

A graphic device is an additional terminal that can be used for system functions. This additional terminal is sometimes called the secondary console. Although not required to complete the installation, having a graphic device available in later steps, when the OPERATOR virtual machine is automatically logged on, enables you to leave OPERATOR logged on at the system console (so that system messages can be displayed there) while you log on to the graphic device as MAINT to continue the installation.

ENTER or *rdev*

020 is the default.

> **WARNING**
>
> Do not define a graphic device at an address with the same control unit (same first two hexadecimal digits) as the system console.

```
ENTER DEVICE TYPE (3277,3278,3066):
```

ENTER or *devtype*

3277 is the default. For a 3290 device, enter **3278**.

```
***SYSTEM DEFINITION COMPLETED***

rdev PRINTER
rdev PUNCH
rdev READER
rdev FIRST TAPE
rdev SECOND TAPE
rdev WORK PACK
rdev GRAPHIC DEVICE

ARE THE ABOVE ENTRIES CORRECT (YES,NO):
```

yes or **no**

If you answer **no**, the address and device type prompts reappear.

■ Initialize the Starter System, as follows:

```
VM/SP Release 6, Service Level nnnn; created on mm/dd/yy at hh:mm:ss

It is now hh:mm:ss EDT day mm/dd/yy
Change TOD clock (YES|NO):
```

yes

⋮

Set the time-of-day (TOD) clock. When prompted for the date and time:

- Enter the date in the format ***mm/dd/yy***
- Enter the time in the format ***hh:mm:ss*** or ***hh.mm.ss***

Check your hardware operation instructions to determine the correct method to enable the TOD clock.

```
DMKCPI971I System is uniprocessor generated
DMKUDR476I System directory loaded from volume VMSRES

DMKCPI974I No valid override file; using system defaults
```

```
DMKCPJ951I CP volid VMPK01 not mounted
```

You receive this message for all Starter System versions except the 9313/9332. VMPK01 is listed in the SYSOWN macro in the Starter System CP nucleus but is not mounted. However, the only volume that you need at this point is VMSRES, so you can ignore the message.

```
Start ((WARM|CKPT|FORCE|COLD) (DRAIN))|(SHUTDOWN):
```

cold

Because there is no data or accounting information to be recovered, you must request a cold start.

Note: You can ignore any DMKLNM*nnn*E messages that appear.

```
AUTO LOGON *** MAINT    USERS=001 BY SYSTEM
```

You are now logged on to the MAINT virtual machine.

```
DMKCPJ952I nnnnnK system storage
```

For some processor types, *nnnnn*K might be less than your actual system storage.

```
⋮
```

```
Formatting error recording area
```

This message appears only during the initial IPL.

```
DMKCPI966I Initialization complete
```

If you do not receive this message, you must go back to the beginning of this step and re-IPL.

```
VM/SP REL 6 STARTER
```

ENTER

This entry signals the system to set up MAINT's virtual machine by:

1. Accessing the CMS system minidisk (MAINT 190) and MAINT's primary (file mode A) minidisk or Shared File System (SFS) directory (in this case, the 191 minidisk)
2. Invoking MAINT's PROFILE EXEC, which issues the TERMINAL MODE VM commmand and accesses the MAINT 193 and 295 minidisks.

```
Ready;
```

■ Go to "Step 4. Load the Installation Tools and Profiles" on page 3-37.

Procedure, Second Level

- Obtain support for your second level terminals by entering the following commands:

ATTN

This entry puts you in first level CP.

set ecmode on

This command allows you to run a second level CP.

Local Terminals Only

- If you are using **local terminals** (whose controller is connected to the host computer through a direct channel), enter:

terminal conmode 3270

This command provides full-screen System Product Editor (XEDIT) support on your second level terminals. The display units must be local non-SNA devices. If you log off or disconnect and then log on again, you must reenter this command.

Note: If you receive the message

```
DMKCFT006E Invalid device type - devtype
```

you do not have a local terminal. Skip the next command.

terminal scrnsave on

This command instructs the system to save your second level screen when you go to first level CP.

End of Local Terminals Only

terminal linend %

This command defines your first level logical line end character as %, which enables you to use the standard logical line end character # at second level. You use these line end characters to indicate first level and second level CP commands.

For example:

#cp query dasd is a second level command.

%cp query dasd is a first level command.

| **define storage 2m**

Your first level system must have **at least** 2M storage to contain your second level VM/SP system.

Re-IPLing Your Second Level System

If you **log off** your **first level** virtual machine during this step or any of the remaining steps in the procedure, you must reissue the previous sequence of first level commands to reestablish the second level environment before you can re-IPL your second level system. You must also make sure that your I/O devices are attached to your second level system and varied on.

RESTART POINT

This is the restart point if you have a failure while building the Starter System at second level.

- IPL the VMSRES volume:

ipl _vdev_ clear

vdev is the first level virtual address of the VMSRES volume.

Note: After you IPL, your system might enter the WAIT (inactive) state, and the DDR program might still be displayed on your screen. To exit the WAIT state and clear the DDR program, press ATTN, then press ENTER.

```
VM/SP: STARTER SYSTEM
```

Note: If the missing interrupt handler (MIH) is turned on, the following message appears repeatedly:

```
546I Interruption cleared; device devtype, CSW csw, userid userid
```

This is not an indication of a problem; the system is operating correctly.

- Define your Starter System configuration by answering the following series of prompts.

  The Starter System provides default device addresses and device types. To accept the Starter System default for a device prompt, just press the ENTER key.

  You do not need a printer, card reader, punch, second tape drive, or additional graphic device to generate the system. However, your system provides a virtual equivalent for each of these devices. If you do not have these devices, press ENTER to accept the Starter System defaults.

If you do not want to accept a Starter System default, enter the address (*vdev*) or device type (*devtype*). If you supply an address for a device, use the first level virtual address. At second level, this address functions as a "real" address. Refer to your Installation Reference Worksheet to help you complete this section.

Whether you accept the Starter System default addresses or define your own, make sure that your second level "real" addresses match your first level virtual addresses.

Re-IPL Only

```
*** DO YOU WISH TO RE-DEFINE YOUR SYSTEM *** (YES,NO):
```

This prompt does not appear during the initial IPL of the Starter System. It appears only during a re-IPL.

yes or **no**

If you are re-IPLing the Starter System, and the device addresses are not specified correctly, or if the tapes or minidisks have been moved, reply **yes** and answer the device prompts to redefine your system.

Otherwise, reply **no** and skip to the next operation (■) on page 3-34 to initialize the Starter System.

End of Re-IPL Only

```
ENTER PRINTER ADDRESS (CUU):
```

ENTER or *vdev*

00E is the default.

```
ENTER DEVICE TYPE (1403,1443,3203,3211,3262,3289E,4245,4248):
```

ENTER or *devtype*

1403 is the default.

If you specify a printer device type that is different from the printer device type defined in your first level virtual machine, you must redefine your first level virtual printer to be able to print from your second level machine. Press ATTN to return to first level CP, then enter:

detach 00e
define *devtype* 00e

where *devtype* is the same printer device type that you just defined for your second level machine. Then press ATTN to return to second level.

```
ENTER PUNCH ADDRESS (CUU):
```

ENTER or *vdev*

00D is the default.

```
ENTER DEVICE TYPE (2540P,3525):
```

ENTER or *devtype*

2540P is the default.

```
ENTER READER ADDRESS (CUU):
```

ENTER or *vdev*

00C is the default.

```
ENTER DEVICE TYPE (2540R,2501,3505):
```

ENTER or *devtype*

2540R is the default.

```
ENTER ADDRESS WHERE FIRST TAPE IS MOUNTED (CUU):
```

ENTER or *vdev*

This is the address of your Starter System Tape. **181** is the default.

```
ENTER DEVICE TYPE (3420,2415,2420,2401,3422,3430,3480,8809,9347):
```

ENTER or *devtype*

3420 is the default.

```
ENTER ADDRESS OF A SECOND TAPE DRIVE (CUU):
```

ENTER or *vdev*

Do not specify the same address as the first tape drive. **182** is the default.

```
ENTER DEVICE TYPE (3420,2415,2420,2401,3422,3430,3480,8809,9347):
```

ENTER or *devtype*

3420 is the default.

```
ENTER DEVICE ADDRESS OF WORK PACK (CUU):
```

ENTER or *vdev*

This is the address of your VMPK01 volume (currently labeled SPSYS). **151** is the default.

```
ENTER DEVICE TYPE (3380,3330,3340,3350,3375,FB-512):
```

ENTER or *devtype*

3380 is the default.

```
ENTER ADDRESS OF A GRAPHIC DEVICE (CUU):
```

A graphic device is an additional terminal that can be used for system functions. This additional terminal is sometimes called the secondary console. Although not required to complete the installation, having a graphic device available in later steps, when the OPERATOR virtual machine is automatically logged on, enables you to leave OPERATOR logged on at the primary console (so that system messages can be displayed there) while you log on to the graphic device as MAINT to continue the installation.

ENTER or *vdev*

020 is the default.

> **WARNING**
>
> Do not define a graphic device at an address with the same control unit (same first two hexadecimal digits) as the primary console device.

```
ENTER DEVICE TYPE (3277,3278,3066):
```

ENTER or *devtype*

3277 is the default. For a 3290 device, enter **3278**.

```
***SYSTEM DEFINITION COMPLETED***

vdev PRINTER
vdev PUNCH
vdev READER
vdev FIRST TAPE
vdev SECOND TAPE
vdev WORK PACK
vdev GRAPHIC DEVICE

ARE THE ABOVE ENTRIES CORRECT (YES,NO):
```

yes or **no**

If you answer **no**, the address and device type prompts reappear.

- Initialize the Starter System, as follows:

```
VM/SP Release 6, Service Level nnnn; created on mm/dd/yy at hh:mm:ss

It is now hh:mm:ss EDT day mm/dd/yy
Change TOD clock (YES|NO):
```

no

At second level, you do not have access to the physical hardware needed to change the time-of-day (TOD) clock.

```
DMKCPI971I System is uniprocessor generated
DMKUDR476I System directory loaded from volume VMSRES

DMKCPI974I No valid override file; using system defaults
```

```
DMKCPJ951I CP volid VMPK01 not mounted
```

You receive this message for all Starter System versions except the 9313/9332. VMPK01 is listed in the SYSOWN macro in the Starter System CP nucleus but is not mounted. However, the only volume that you need at this point is VMSRES, so you can ignore the message.

```
Start ((WARM|CKPT|FORCE|COLD) (DRAIN))|(SHUTDOWN):
```

cold

Because there is no data or accounting information to be recovered, you must request a cold start.

Note: You can ignore any `DMKLNM`*nnn*`E` messages that appear.

```
AUTO LOGON *** MAINT    USERS=001 BY SYSTEM
```

You are now logged on to the MAINT virtual machine.

```
DMKCPJ952I nnnnnK system storage
```

For some processor types, *nnnnn*K might be less than your actual system storage.

⋮

```
Formatting error recording area
```

This message appears only during the initial IPL.

```
DMKCPI966I Initialization complete
```

If you do not receive this message, you must go back to the beginning of this step and re-IPL.

```
VM/SP REL 6 STARTER
```

ENTER

This entry signals the system to set up MAINT's virtual machine by:

1. Accessing the CMS system minidisk (MAINT 190) and MAINT's primary (file mode A) minidisk or Shared File System (SFS) directory (in this case, the 191 minidisk)
2. Invoking MAINT's PROFILE EXEC, which issues the TERMINAL MODE VM command and accesses the MAINT 193 and 295 minidisks.

```
Ready;
```

Stopping and Restarting the Installation

From this point on, when installing VM/SP at second level, you can stop the installation after the completion of any step by entering the following command to **disconnect** your **first level** virtual machine:

ATTN

%cp disconn

To restart, log on again to your first level virtual machine and enter:

terminal conmode 3270 (local terminals only)
begin

This puts you back where you were when you stopped.

■ Go to "Step 4. Load the Installation Tools and Profiles" on page 3-37.

Step 4. Load the Installation Tools and Profiles

Overview

The second tape file on the first volume of the VM/SP Product Tape contains installation tools and profiles (the first tape file contains header information). These installation tools and profiles include:

ITASK EXEC This is a high level tool that calls other tools and issues commands to do the installation and system generation tasks.

SPLOAD EXEC This tool directs the loading of tape files from the product tape and feature tapes, using the information contained in the SPLOAD PROFILE. It is called by ITASK, or you can invoke it directly.

SPLOAD PROFILE This file indicates the name and location of the tape files on the product tape and feature tapes, and specifies the minidisk or SFS directory where the SPLOAD EXEC loads each tape file.

PROFILE EXEC This is the default profile for the MAINT virtual machine. It issues the TERM MODE VM command (to allow MAINT to use CMS commands) and accesses the MAINT 193 and MAINT 295 minidisks.

In this step, you issue the VMFPLC2 command to load the INSTALL TOOLS tape file from the product tape to the MAINT 191 minidisk. For more information about ITASK EXEC, SPLOAD EXEC, and SPLOAD PROFILE, see the descriptions in Chapter 8, "Tools." For more information about the VMFPLC2 command, see the *VM/SP Service Guide*.

First and Second Level Considerations

To do this step at **first level**, go to "Procedure, First Level" on page 3-38.

To do this step at **second level**, go to "Procedure, Second Level" on page 3-40.

Procedure, First Level

Note: If your console is in Printer-Keyboard mode (a processor IML option), press the REQUEST key before entering each command. For more information about IML options, refer to the *Operator's Guide* provided with your processor.

- Mount and ready the first product tape volume.

  If your system has only one tape drive, remove the Starter System Tape and mount the product tape. If your system has more than one tape drive, you can mount the product tape on either of the drives that you defined in Step 3.

- Attach the tape drive to the MAINT virtual machine by entering the following commands:

vary on *rdev*

rdev is the real address of the tape drive where you mounted the product tape. This command marks the device available to CP.

```
rdev VARIED ONLINE
```

attach *rdev* maint 181

rdev is the real address of the tape drive. The SPLOAD EXEC requires that the tape drive be attached to MAINT at virtual address 181.

```
TAPE rdev ATTACH TO MAINT 181
```

- Load the INSTALL TOOLS tape file:

vmfplc2 fsf

This command moves the product tape forward one tape mark to bypass the header file.

vmfplc2 load

This command takes the next tape file (INSTALL TOOLS) from virtual address 181 and loads it to file mode A (in this case, the MAINT 191 minidisk).

> **Tailoring the SPLOAD PROFILE**
>
> In later steps, you invoke the ITASK EXEC or the SPLOAD EXEC to load the remaining tape files from the product tape. The minidisk and SFS directory destinations are specified in the SPLOAD PROFILE.
>
> **Warning:** Do not attempt to tailor the SPLOAD PROFILE unless you have a thorough working knowledge of the system and are aware of the implications of your changes. For example, changing this file might prevent you from using the ITASK EXEC to do some installation tasks, because ITASK expects to find certain information at specific minidisk or directory locations. There are no safeguards in the code to check the validity of your changes.

- To continue the installation procedure, go to "Step 5. Load the System Generation Tools, System Tools, CP Code, and System Definition Files" on page 3-42.

Procedure, Second Level

- Mount the first product tape volume on a tape drive attached to your **first level** virtual machine, as defined in Step 1 (the system operator may have to do this for you).
- Attach the tape drive to your **second level** (MAINT) virtual machine by entering the following commands:

vary on *vdev*

vdev is the virtual address of the tape drive attached to your first level virtual machine, where the product tape is mounted. At second level, this virtual address functions as the "real" address of the tape drive. This command marks the device available to second level CP.

```
vdev VARIED ONLINE
```

attach *vdev* maint 181

vdev is the same tape drive address that you used above. The SPLOAD EXEC requires that the tape drive be attached to MAINT at virtual address 181.

```
TAPE vdev ATTACH TO MAINT 181
```

- Load the INSTALL TOOLS tape file:

vmfplc2 fsf

This command moves the product tape forward one tape mark to bypass the header file.

vmfplc2 load

This command takes the next tape file (INSTALL TOOLS) from virtual address 181 and loads it to file mode A (in this case, the MAINT 191 minidisk).

Tailoring the SPLOAD PROFILE

In later steps, you invoke the ITASK EXEC or the SPLOAD EXEC to load the remaining tape files from the product tape. The minidisk and SFS directory destinations are specified in the SPLOAD PROFILE.

Warning: Do not attempt to tailor the SPLOAD PROFILE unless you have a thorough working knowledge of the system and are aware of the implications of your changes. For example, changing this file might prevent you from using the ITASK EXEC to do some installation tasks, because ITASK expects to find certain information at specific minidisk or directory locations. There are no safeguards in the code to check the validity of your changes.

- To continue the installation procedure, go to "Step 5. Load the System Generation Tools, System Tools, CP Code, and System Definition Files" on page 3-42.

Step 5. Load the System Generation Tools, System Tools, CP Code, and System Definition Files

Overview

Now that you have loaded the ITASK EXEC from the product tape, you can invoke ITASK with various operands to help you execute most of the remaining steps in this procedure. For a detailed description of ITASK operands, see "ITASK EXEC" on page 8-9.

In this step, you invoke the ITASK EXEC with the LOAD CP operands. ITASK:

- Calls the SPLOAD EXEC to load the following tape files from the product tape to the destinations specified in the SPLOAD PROFILE:
  - SYSGEN TOOLS to MAINT 193

    This tape file contains various system generation tools and files, including:

    5664167E $PPF
    VMFBLD EXEC
    UTILITY EXEC
    SNTMAP EXEC
    DISKMAP EXEC
    DIRECGEN EXEC
    PASSMOD EXEC
    INSTFPP EXEC
    $MSG4I$ EXEC
    $DASD$ CONSTS

  - SYSTEM TOOLS to MAINT 190

    This tape file contains other system tools, including:

    VMFASM EXEC
    VMFOVER EXEC
    VMFSETUP EXEC

  - CP DELTA to MAINT 294

    This tape file contains CP delta service code (update files and serviced text decks).

  - CP APPLY to MAINT 192

    This tape file contains CP apply service code (auxiliary control files).

  - CP OBJECT to MAINT 194

    This tape file contains the CP object code and the following printer universal character set and font offset buffer sample files:

    DMKUCS ASSEMBLE
    DMKUCB ASSEMBLE
    DMKUCC ASSEMBLE
    DMKPIA ASSEMBLE
    DMKPIB ASSEMBLE

  - SYSTEM SAMPLES to MAINT 295

    This tape file includes the following sample files:

    DMSNGP SAMPLE

DMKBOX ASSEMBLE
DMKFCB ASSEMBLE
RPWLIST DATA
VMSYS POOLDEF
VMSERVS DMSPARMS
VMSYSU POOLDEF
VMSERVU DMSPARMS
SERVER PROFILE
DIRECGEN EXTENTS

- DASDMODL SAMPLES to MAINT 295

  This tape file contains the following sample files:

  DIRECT *devtype[-mmm]*
  DMKSNT *devtype[-mmm]*

  where *devtype[-mmm]* identifies the DASD type and model of your VMSRES device

- DASDTYPE SAMPLES to MAINT 295

  This tape file contains the following sample files:

  DMKRIO *devtype*
  DMKSYS *devtype*

  where *devtype* identifies the DASD type of your VMSRES device

- Calls the COPYFILE command to create copies (with new names) of the following sample files:

Loaded Sample File	**New Sample File**
DIRECT *devtype[-mmm]*	VMUSERS DIRECT
DMKSNT *devtype[-mmm]*	DMKSNT ASSEMBLE
DMKRIO *devtype*	DMKRIO ASSEMBLE
DMKSYS *devtype*	DMKSYS ASSEMBLE
DMSNGP SAMPLE	DMSNGP ASSEMBLE

Note: If you need to apply preventive service, corrective service, or local service to CP, you can do so at the end of this step.

Procedure

WARNING

To do the load, the ITASK EXEC calls the COPYFILE command with the REPLACE option. This command overwrites any files already on the minidisk or in the SFS directory that have the same labels as the sample files being copied to it. If you are redoing this step, and you have already tailored some of the sample files, you might want to rename the files that you tailored before you invoke ITASK.

- Load the system generation tools, system tools, CP code, and system generation files:

itask load cp

```
DMSWSL409I Loading SYSGEN TOOLS to MAINT 193
DMSWSL409I Loading SYSTEM TOOLS to MAINT 190
```

Notes:

1. There are pauses between some of the following messages. Before loading the first tape file to MAINT 295, 294, 192, or 194, the SPLOAD EXEC must format the minidisk.
2. As each minidisk is formatted, you might receive two device error messages. You can ignore these messages.

```
DMSWSL409I Loading CP DELTA to MAINT 294
DMSWSL409I Loading CP APPLY to MAINT 192
DMSWSL409I Loading CP OBJECT to MAINT 194
DMSWSL409I Loading SYSTEM SAMPLES to MAINT 295
DMSWSL409I Loading DASDMODL SAMPLES to MAINT 295
DMSWSL409I Loading DASDTYPE SAMPLES to MAINT 295

DMSWTK965I You may wish to tailor the following files
           at this time:
VMUSERS DIRECT
DMKSNT  ASSEMBLE
DMKRIO  ASSEMBLE
DMKSYS  ASSEMBLE
DMKBOX  ASSEMBLE
DMKFCB  ASSEMBLE
DMSNGP  ASSEMBLE
VMSERVS DMSPARMS
VMSERVU DMSPARMS
VMSYS   POOLDEF
VMSYSU  POOLDEF
```

ipl 190 clear

WARNING - Second Level Only

Before loading the tape files in this step, ITASK EXEC issued a DIAGNOSE code X'24' for the address of your system residence volume to determine the DASD type and model. Because the product tape contains a DASDTYPE SAMPLES tape file and DASDMODL SAMPLES tape file for each supported DASD type and model, ITASK required this information to know which tape files to load.

If you are installing at **second level**, your system residence volume is a first level minidisk. If you have defined that minidisk as **less than a full pack minidisk**, the information that the DIAGNOSE code X'24' obtained and passed to ITASK refers to the real device and might not match the model of the virtual device that you defined. Because the DASDTYPE SAMPLES tape file is not model-specific, the correct file to match your DASD has been loaded. However, if the DASD type that you are using has more than one model (see the table on page 3-7), the DASDMODL SAMPLES tape file that ITASK loaded might not be correct. DASDMODL SAMPLES contains the DMKSNT and CP directory files.

To remedy this situation:

1. Load the contents of the correct tape file:

   sploador dasdmodl samples * *devtype[-mmm]*

   where *devtype[-mmm]* is the device type and model of your system residence device.

2. Copy the correct DMKSNT and directory files over the incorrect ones and rename them to the names required to generate your system:

   copyfile direct *devtype[-mmm]* b vmusers direct = (replace
   copyfile dmksnt *devtype[-mmm]* b dmksnt assemble = (replace

- The recommended procedure for applying service to CP is to complete the VM/SP installation first, then apply the service. However, if you need to apply preventive service, corrective service, or local service to CP now, use the procedures described in the *VM/SP Service Guide* to receive and/or apply the CP service. **Do not** receive or apply service for any other component. **Do not** use the VMFBLD EXEC to build CP. After you apply the CP service, continue with the rest of this installation procedure.

  Note: If you apply service to CP, you might need to rebuild serviced modules before you continue with the installation procedure.

- Go to “Step 6. Tailor the System Definition Files” on page 3-46.

Step 6. Tailor the System Definition Files

Overview

As shipped on the product tape, the system definition files contain default parameters and sample data. Based on the requirements that you established in pre-installation planning (Chapter 2), **you must examine and modify these files** to define your unique system configuration:

- The VM/SP product parameter file (5664167E $PPF), which you loaded to the MAINT 193 minidisk in Step 5, contains control options, minidisk and SFS directory assignments, and a part type/function list for each VM/SP component. These parameters are used by the nucleus building tool (VMFBLD EXEC) and other tools.
- The CP and CMS nucleus definition files, which you loaded to the MAINT 295 and 194 minidisks in Step 5, contain the following information:
  - The CP directory (VMUSERS DIRECT) identifies the virtual machines (user IDs) permitted to log on to your system.
  - The CP system control file (DMKSYS ASSEMBLE) describes the system residence device (VMSRES) and various system parameters.
  - The real I/O configuration file (DMKRIO ASSEMBLE) defines the configuration of your system input/output devices.
  - The system name table (DMKSNT ASSEMBLE) defines the name, size, and virtual storage location of named saved systems and physical saved segments.
  - The forms control buffer file (DMKFCB ASSEMBLE) contains real and virtual printer parameters.
  - The CMS nucleus generation profile (DMSNGP ASSEMBLE) contains responses to the system prompts that are generated when you build a CMS nucleus.
  - The DMKBOX ASSEMBLE file defines the appearance and content of system logos.
  - Other system definition files (DMKUCB, DMKUCC, DMKUCS, DMKPIA, DMKPIB) define the printer universal character set and the font offset buffer.
- The CMS file pool definition files, which you loaded to the MAINT 595 minidisk in Step 5, contain information used to build a system file pool (required) and a user file pool (optional):
  - VMSYS POOLDEF contains information about the minidisks in the system file pool.
  - VMSERVS DMSPARMS contains start-up parameters for the system file pool server machine.
  - VMSYSU POOLDEF contains information about the minidisks in the user file pool.
  - VMSERVU DMSPARMS contains start-up parameters for the user file pool server machine.

To tailor any of these files, use the VM/SP System Product Editor (XEDIT) that is provided on the Starter System. XEDIT offers full-screen editing on display terminals.

To help you do the tailoring, you can use the UTILITY EXEC to print copies of the following files:

- VMUSERS DIRECT
- DMKSYS ASSEMBLE
- DMKRIO ASSEMBLE
- DMKSNT ASSEMBLE
- DMSNGP ASSEMBLE.

Procedures

Note: Where necessary, you are directed to other books in the VM/SP library for additional information about the changes outlined in this step.

Restarting This Step

If you have to restart this step, make sure that you have read/write access to the minidisks where the sample files are located. Enter:

access 295 b

access 194 c

Printing the Sample Files

Before you begin the tailoring, you can invoke the UTILITY EXEC to print copies of the VMUSERS DIRECT, DMKSYS ASSEMBLE, DMKRIO ASSEMBLE, DMKSNT ASSEMBLE, and DMSNGP ASSEMBLE files:

utility prsample

The files are spooled to your system printer.

Notes:

1. If you modified the SPLOAD PROFILE and loaded any of these sample files to a non-default minidisk, access that minidisk before you invoke the UTILITY EXEC.
2. If the printer has been turned off:
   a. Turn the power on.
   b. Enter the **vary online** ***nnn*** command (where ***nnn*** is the address of your printer) to identify the device to CP.
   c. Define the printer parameters as indicated below.
3. If the print operation fails, or if the printer has just been turned on, you might have to define the printer parameters:
   a. Enter the DRAIN command to stop spooling operations on the device.
   b. Enter the necessary LOADBUF command(s) to define the printer.
   c. Enter the START command to restart the device.
   d. Invoke the UTILITY EXEC to print the files.

   For more information about using the DRAIN, LOADBUF, and START commands, see the *VM/SP CP System Command Reference.*

Using the System Product Editor

This section outlines only the basic System Product Editor (XEDIT) procedure. For additional information, see the *VM/SP CMS Primer* or the *VM/SP System Product Editor User's Guide.*

To examine or modify a file, invoke XEDIT with the file ID:

xedit ***fn ft fm***

Use the cursor control keys, the scroll keys, the insert key, and the delete key to make the necessary changes.

To save the file after you complete your changes, enter:

file

Product Parameter File Considerations

- Tailor the VM/SP product parameter file if you want to change the supplied system generation parameters:

xedit 5664167e $ppf

There is a separate area in the product parameter file for each VM/SP component. A component area begins with a component tag (such as :CP.).

Note: The *VM/SP Service Guide* describes the function of the tags and records in the product parameter file and also provides a complete listing of the base product parameter file shipped on the VM/SP product tape.

Examine the following parameters:

– **Minidisk and SFS Directory Assignments**

In each component area, there is a section that begins with a minidisk/directory assignments tag (:MDA.) that defines the minidisks and/or SFS directories assigned to the functional groups (BUILD, DELTA, APPLY, and so on) for that component. Make sure that these assignments reflect your actual system configuration, and that essential items are not commented out (with * in the first column).

– **Building an AP or MP System**

The base product parameter file contains parameters for building a uniprocessor (UP) system, but includes parameters for building an attached processor (AP) or multiprocessor (MP) system in CP override areas called CPAP and CPMP, respectively. In Step 7, when you issue the VMFBLD command to build the CP nucleus, to build an AP or MP system, enter the name of the corresponding override area (CPAP or CPMP) instead of CP.

– **National Language (American English or Uppercase English)**

VM/SP is shipped with mixed-case American English as the default national language. The language ID (*langid*) for American English is AMENG. Files for uppercase English are also provided on the product tape. The language ID for uppercase English is UCENG. When you build your VM/SP system, American English is installed by default as the system national language, which is the language used for most system functions. If you want to build an uppercase English system, locate the national language tag (:NLS.) in the CP, CMS, and GCS component areas of the product parameter file and change the language ID from AMENG to **UCENG** (the parameter must be in uppercase).

Note: Some system functions are always in American English regardless of the language installed as the system national language. For example, the language of the installation and system generation tools is American English and cannot be changed.

- **CP Options**

  There are three options that you can select when building the CP nucleus:

  - Small CP
  - CP FRET Trap
  - Virtual = Real.

  These options are selected by the CP control file and/or build list (loadlist) specified in the product parameter file. In the IBM-supplied product parameter file, these options are **not** selected. For information about these options and how to select them, see Appendix D, "Special Options for CP" on page D-1.

- Save your changes:

```
file
```

CP Directory Considerations

- Tailor the sample CP directory file to define your virtual machine (user) configuration:

```
xedit vmusers direct
```

Consider the following changes:

- **Defining Non-Restricted User Logon Passwords**

  Each virtual machine in the system is protected from unauthorized access by a logon password. This logon password is specified in the USER statement in the CP directory entry that defines the virtual machine.

  As a system security measure, the sample CP directory defines the logon prevention password NOLOG for all virtual machines except MAINT and OPERATOR. You cannot log on to any virtual machine that has a NOLOG password. In addition, the passwords defined for MAINT and OPERATOR are restricted passwords. The RPWLIST DATA file that you loaded from the product tape to the MAINT 295 disk in Step 5 contains a list of all restricted passwords, including the passwords supplied for MAINT and OPERATOR.

  The Starter System CP nucleus contains a feature called Auto-Deactivation of Restricted Passwords (ADRP). The new CP nucleus that you build in Step 7 also contains this feature. Under ADRP, entering the DIRECT command to process the directory file causes the system to search the directory for logon passwords that match the restricted list in the RPWLIST DATA file. All passwords that match are changed to NOLOG in the directory before the directory is placed on line.

  Use XEDIT to edit the sample directory and replace the supplied MAINT and OPERATOR logon passwords with unique non-restricted passwords.

Define unique non-restricted passwords for all other virtual machines to remove the NOLOG condition. For a list of the restricted passwords, see Appendix B, "Restricted Logon Passwords" on page B-1.

- **Using Mixed DASD**

  The sample CP directory defines all system DASD as the same type as the VMSRES device. If you are using a different device type for any system volume (such as VMPK01 or VMPK04), some adjustment is necessary in the size and location of the minidisks defined on that volume by the MDISK statements in the directory. For reference, you might want to look at the sample CP directory for the other DASD type. To load the file from the product tape, enter:

  spload dasdmodl samples direct *devtype[-mmm]*

  where ***devtype[-mmm]*** is the device type (and model designation, if any) of the other DASD. After you load the file, you can browse it (with XEDIT) and/or print it.

- **Allocating TEMP Space**

  TEMP space is documented by MDISK statements under the $TEMP$ user ID. The amount of TEMP space allocated in the sample CP directory is sufficient to generate a production VM/SP system. However, the amount of TEMP space required to support your installation depends on how many users you have and how you use your TEMP space. For example, the sample system definition files do not provide space for paging or for dumps. TEMP space is used for paging and for dumps unless you specifically define areas for those functions. See "DMKSYS Considerations" on page 3-53.

- **Modifying the System File Pool Server Machine**

  You might want to change the supplied specification for the system file pool server virtual machine (the user ID is VMSERVS). For example, you might want to increase the minidisk allocation for the system file pool.

  Note: Changing the definition of this machine might also require changes to the VMSYS POOLDEF file. You should not attempt to change any of these files unless you have a thorough knowledge of how SFS works.

  For more information about the configuration of a file pool server, see *VM/SP CMS Shared File System Administration.*

- **Modifying the User File Pool Server Machine**

  You might want to change the supplied specification for the user file pool server virtual machine (the user ID is VMSERVU). For example, you might want to increase the minidisk allocation for the user file pool.

  Note: Changing the definition of this machine might also require changes to the VMSYSU POOLDEF file. You should not attempt to change any of these files unless you have a thorough knowledge of how SFS works.

  For more information about the configuration of a file pool server, see *VM/SP CMS Shared File System Administration.*

- **Modifying the GCS Recovery Machine**

  If you are installing the GCS component, you might want to change the specification of the sample GCS recovery machine (the user ID is GCS). For more information about GCS, see the *VM/SP Planning Guide and Reference.*

- **Modifying the TSAF Machine**

  If you are installing the TSAF component, you might want to change the specification of the sample TSAF virtual machine (the user ID is TSAFVM). Also remember that all user IDs in all systems in a TSAF collection must be unique. For more information about TSAF, see *VM/SP Connectivity Planning, Administration, and Operation.*

- **Modifying the AVS Machine**

  If you are installing the AVS component, you might want to change the specification of the sample AVS virtual machine (the user ID is AVSVM). For more information about AVS, see *VM/SP Connectivity Planning, Administration, and Operation.*

- **Adding User IDs**

  If you plan to install optional components (such as GCS) or program products (such as RSCS) that require system user IDs not provided in the sample directory, you might want to create directory entries for them now using the format described in the *VM/SP Planning Guide and Reference.*

  However, if you plan to install an optional feature program product package, you should examine the installation instructions for the package. The installation procedure for the package might include an operation for defining the necessary user IDs.

- **Changing Minidisk Read/Write Passwords**

  Sample read/write passwords are provided for the minidisks defined for each virtual machine in the directory. If you plan to install optional feature program products on the VM/SP base, the feature program product installation tool (INSTFPP EXEC) assumes that certain minidisks have specific passwords. Do not delete or alter the passwords of minidisks related to these products until after they are installed.

- **Restructuring User Classes**

  The user classes specified in the directory are IBM-standard. To restructure these classes, you must make the appropriate changes to the directory and provide OVRD formatted space for the OVERRIDE file. OVRD space must be allocated on the same volume as the directory.

  If you plan to restructure the user classes, you should do it **after** your base system is installed. For additional information, see *VM/SP Administration.*

- **Defining Additional Minidisks**

  You can define additional permanent minidisks by adding MDISK statements under the appropriate user IDs. To establish shared minidisks, add the necessary LINK statements. For information about the format of MDISK and LINK statements, see the *VM/SP Planning Guide and Reference.*

■ Save your changes:

```
file
```

■ If you add minidisks to the directory or change minidisk allocations, use the DISKMAP EXEC to check for gaps and overlaps between minidisk assignments:

```
diskmap vmusers direct
```

DISKMAP produces a file called VMUSERS DISKMAP that is written to file mode A. Use XEDIT to examine the file. For more information about DISKMAP EXEC, see the *VM/SP Planning Guide and Reference.*

- Enter the DIRECT command to place the tailored directory on line:

> WARNING
>
> Before you enter the DIRECT command, make sure that you have changed the supplied MAINT and OPERATOR logon passwords and that your directory now contains only **non-restricted** passwords. All restricted logon passwords in the directory will be changed to NOLOG before the directory is placed on line.

direct vmusers direct

DMKSYS Considerations

- Tailor the sample DMKSYS file to define your system residence configuration:

xedit dmksys assemble

Consider the following changes:

- **Allocating Space for Paging and for Dumps**

  The Starter System and the sample files do not provide space for paging or for dumps. TEMP space is used for paging and for dumps until you allocate space for these functions. For information about calculating space requirements and allocating space in the SYSOWN macro, see the *VM/SP Planning Guide and Reference*

- **Restructuring User Classes**

  If you plan to restructure the supplied user classes, you might need to change the user class definitions in the SYSFCN macro. See the *VM/SP Planning Guide and Reference* for more information about this macro.

- **Defining the Real VMSRES Address**

  In Step 8 you can create a backup tape containing an IPLable copy of your CP nucleus. However, to make this backup tape usable, you must modify the SYSRES=123 statement in the SYSRES macro to specify the **real** address of the VMSRES volume as the *altaddr*. See the *VM/SP Planning Guide and Reference* for the proper format for this entry.

- **Building an AP or MP System**

  If you are generating an AP or MP system, you must change the SYSCOR macro. See the *VM/SP Planning Guide and Reference* for more information about this macro.

- Save your changes:

file

The tailored DMKSYS ASSEMBLE file is assembled in Step 7.

DMKRIO Considerations

- Tailor the sample DMKRIO file to define the configuration of your system input/output devices:

xedit dmkrio assemble

Consider the following changes:

- **Defining the Console Address**

  In the CONSOLE section of the file, the RIOGEN statement defines the real addresses of the primary console (CONS=) and alternate consoles (ALTCONS=).

  - At **first level**, make sure that the real address of the system console matches one of these addresses.
  - At **second level**, make sure that the address of the terminal you are using as the primary console matches one of these addresses. Also, make a note of other addresses that are listed. In Step 8 you might have to redefine your first level virtual console to one of these addresses before you can IPL your production system.

  In addition, make sure that the primary console address matches the address of MAINT's virtual console, as defined by the CONSOLE statement in the MAINT entry in the CP directory.

- **Defining Unsupported Devices**

  You can define an unsupported device in DMKRIO only if the device is dedicated to a virtual machine (see the *VM/SP Planning Guide and Reference*).

  Note: An unsupported device is flagged with an MNOTE during the assembly, which can be ignored.

- Save your changes:

file

The tailored DMKRIO ASSEMBLE file is assembled in Step 7.

DMKSNT Considerations

- Tailor the sample DMKSNT file to define the configuration of named saved systems and physical saved segments in virtual storage:

xedit dmksnt assemble

If you make any changes to this file, you must group like macros together in the DMKSNT CSECT. That is, group all NAMESYS macros together, group all NAMENCP macros together, group all NAME3800 macros together, and group all NAMELANG macros together. If you group them incorrectly (for example, by placing a NAMENCP macro between two NAMESYS macros), the system generates an MNOTE indicating that the macros are out of sequence.

WARNING

Make sure that your changes do not cause overlay of other products on your DASD. Also make sure that no physical saved segment or saved system overlays another physical saved segment or saved system in virtual storage unless they are **never** used at the same time. Table 6-1 on page 6-3 shows the layout of physical saved segments and saved systems defined by the sample DMKSNT.

Consider the following changes:

- **Using Mixed DASD**

  The sample DMKSNT file assumes that all system DASD volumes are the same device type as the VMSRES volume. If you are using a different device type for any volume, some adjustment in the location of named saved systems and/or physical saved segments on the volume might be necessary. For reference, you might want to look at the sample DMKSNT file for the other DASD type. To load the file from the product tape, enter:

  spload dasdmodl samples dmksnt *devtype[-mmm]*

  where ***devtype[-mmm]*** is the device type (and model designation, if any) of the other DASD. After you load the file, you can browse it (using XEDIT) and/or print it.

- **Modifying the GCS Entry**

  The sample DMKSNT file includes an entry that defines a GCS saved system. If you are installing the GCS component, in your pre-installation planning (Chapter 2) you recorded on your Installation Reference Worksheet any changes to be made to this sample GCS entry (such as the load address, or the name of the saved system). Make those changes now to the file.

  If you plan to install more than one GCS nucleus, you must create entries for the additional systems.

- **Adding New Entries**

  If you plan to install additional national languages or optional feature program products that require physical saved segments, you should add the DMKSNT entries now. Otherwise, you will have to reassemble DMKSNT and rebuild the CP nucleus.

The tailored DMKSNT ASSEMBLE file is assembled in Step 7.

■ Save your changes:

file

■ To verify your changes, use the SNTMAP EXEC, an SNT mapping tool that resides on the MAINT 193 minidisk:

sntmap dmksnt

SNTMAP produces two output files on file mode A:

- DASD SNTMAP lists the saved system and physical saved segment names and DASD allocations in order of occurrence on each DASD volume.

- MEMORY SNTMAP lists the saved system and physical saved segment names and DASD allocations according to location in virtual storage.

For more information about the SNTMAP EXEC, see the *VM/SP Planning Guide and Reference.*

DMSNGP Considerations

■ Tailor the sample DMSNGP file to define your CMS configuration:

xedit dmsngp assemble

Consider the following changes:

Note: On each line, make sure that you have at least one blank inserted between the macro statement and the comment section.

- **CMS Nucleus Address**

  You must update the CYLADDR statement to indicate the starting cylinder/block on MAINT 190 where the CMS nucleus is written, which depends on your DASD type:

Device	Cylinder/Block
3350	70 (cyl)
3375	108 (cyl)
3380	70 (cyl)
FBA	63744 (blk)

- **National Language (American English or Uppercase English)**

  The default is American English. If you want to build an uppercase English system, you must change the LANGID statement from AMENG to UCENG.

- **Other Changes**

  You might want to change other statements, such as the version identification (VERSION) and the installation heading (INSTID). You must place single quotes around any character string that you enter as the new VERSION or INSTID.

For more information about DMSNGP, see the *VM/SP Planning Guide and Reference.*

■ Save your changes:

file

The tailored DMSNGP ASSEMBLE file is assembled when the CMS nucleus is built in Step 11. **Do not** attempt to assemble it with the other system definition files in Step 7.

File Pool Considerations

The generation of a file pool requires data from two file pool definition files:

- The POOLDEF file contains information about the location of the minidisks in the file pool and their function.
- The DMSPARMS file contains start-up parameters for the file pool server machine.

Two sets of files are supplied on the product tape:

- VMSYS POOLDEF and VMSERVS DMSPARMS define a system file pool that is used for loading GCS, TSAF, and AVS code.
- VMSYSU POOLDEF and VMSERVU DMSPARMS define a user file pool that you can build for user files.

Although these files can be tailored, you should not make any changes unless you have a thorough working knowledge of SFS. For more information, see *VM/SP CMS Shared File System Administration.*

Other Considerations

- If you want to change the supplied forms control buffer macros, tailor the sample DMKFCB ASSEMBLE file. For more information, see *VM/SP Administration.*
- If you want to change the design or contents of system logos, such as the system logon screen logo or the printer separator page logo, tailor the sample DMKBOX ASSEMBLE file.
- If this is a migration, and you have defined new CP commands, see *VM/SP Administration* for information about adding the new commands to CP.
- If you want to modify the supplied universal character set or font offset buffer for the printer(s) in your system, tailor the corresponding sample file(s):

Printer Type	Module (Sample File)
1403	DMKUCS
3211	DMKUCB
3203	DMKUCC
3289	DMKPIA
3262	DMKPIB

See the module prològues for specific information.

These files were loaded to the MAINT 194 minidisk; therefore, to examine or tailor any of these files, you must first access 194:

```
access 194 c
```

- When you complete the tailoring required for your installation, go to "Step 7. Build the New CP Nucleus" on page 3-58.

Step 7. Build the New CP Nucleus

Overview

To build your new CP nucleus, you invoke the ITASK EXEC with the operands that indicate the type of nucleus you want to build:

BUILD CP	to build a uniprocessor (UP) nucleus.
BUILD CPAP	to build an attached processor (AP) nucleus.
BUILD CPMP	to build a multiprocessor (MP) nucleus.

ITASK:

- Accesses the MAINT 295 minidisk, where most of the sample system definition files are located.
- Calls the VMFASM EXEC to assemble the following sample files:
  - DMKBOX ASSEMBLE
  - DMKFCB ASSEMBLE
  - DMKSNT ASSEMBLE
  - DMKSYS ASSEMBLE
  - DMKRIO ASSEMBLE.
- Calls the VMFBLD EXEC to build the CP nucleus. VMFBLD calls the VMFOVER EXEC to obtain the control file name, loadlist name, and other CP parameters from the product parameter file.

  Note: If you enter either CPAP or CPMP on the command, VMFOVER looks for an override area with that name in the product parameter file. An override area contains alternative and/or additional parameters. In this case, the alternative parameters are the file names of the control file and loadlist used to build an AP or MP nucleus. For more information about override areas and the VMFOVER EXEC, see the *VM/SP Service Guide.*

After you build the nucleus, you can save and print the CP load map, which contains the storage addresses of the nucleus control sections and entry points.

Procedure

- Build the CP nucleus:

itask build *compname*

compname is the name of the component area in the product parameter file that contains the CP parameters you want to use. Enter **cp** to build a UP system, **cpap** to build an AP system, or **cpmp** to build an MP system.

```
DMSUPD181E No update files were found
DMSWAM1907I Assembling DMKBOX

ASSEMBLER (XF) DONE
NO STATEMENTS FLAGGED IN THIS ASSEMBLY
DMSWAM1909I DMKBOX TEXT A created
```

⋮

Note: If any statements are flagged during the assembly procedure, the CP nucleus is **not** built. See "Assembly Errors" on page 3-61.

```
DMSWBN1851I Processing loadlist with VMFBDNUC exec
```

loadlist is the name of the CP loadlist specified in the CP, CPAP, or CPMP area of the product parameter file.

```
NUCLEUS LOADED ON VMSRES --- STARTING CYL/BLK=nnnn  , LAST CYL/BLK USED=nnnn
```

```
CP entered; disabled wait PSW '00020000 00000012'
```

Notes:

1. On many processors the PSW message does not display automatically. If the PSW message does not display within about one minute after the `NUCLEUS LOADED` message, enter:

   #cp display psw

2. If you receive an incorrect PSW, you must go back to "Step 2. Restore the Starter System" on page 3-13 and restore the Starter System again. You might have made errors when you tailored the sample files in Step 6. When you reach Step 6 again, check your changes carefully.

3. If you do not receive the `NUCLEUS LOADED` message, or if you receive a non-zero return code, see "Load Errors" on page 3-62.

4. The `NUCLEUS LOADED` message gives the starting and ending cylinders/blocks for the nucleus. If you added minidisks or changed minidisk allocations when you tailored the CP directory in Step 6, you should verify that this

nucleus has not expanded into another allocated area such as TEMP, WARM START, and so on, or unpredictable results could occur.

An easy way to check this is to look at the VMUSERS DISKMAP file that you created after you tailored the directory. The `LAST CYL/BLK USED=`*nnnn* portion of the response displayed above must not exceed the $CPNUC$ ending cylinder/block defined in the directory. Before you can look at the VMUSERS DISKMAP file, you must exit the WAIT state. Enter:

#cp ipl 190 clear

Saving and Printing the CP Load Map

The CP load map has been spooled to MAINT's virtual printer. To save the load map on the CP object code minidisk (MAINT 194), enter:

#cp spool prt nohold
#cp close prt
#cp ipl 190 clear

When `VM READ` appears in the corner of your screen, enter:

ENTER
access 194 c
query rdr

```
MAINT spoolid PRT nnnnnnnn 001 NONE STANDARD OFF
```

receive *spoolid* cpnuc map c

After you save the load map, you can print a copy of it by entering:

print cpnuc map c

■ To continue the installation procedure, go to "Step 8. Shut Down the Starter System CP and IPL the New CP" on page 3-67.

Assembly Errors

- If any DMK*xxx* ASSEMBLE file fails to assemble, you might have made an error when tailoring the file in Step 6.

Note: Any unsupported device defined in DMKRIO is flagged with an MNOTE during the assembly. If you intentionally defined the unsupported device when you tailored the file, the flagged statement does not require correction. Correct all other flagged statement(s) using steps 1-4 listed below until the assembly produces no flagged statements except the allowed unsupported device MNOTE(s). Then do step 5 to build the nucleus.

To correct the error:

1. Examine each flagged statement.
2. Correct the statement(s) using the System Product Editor (XEDIT).
3. Reassemble the file:

itask assemble dmk*xxx* *compname*	**dmk*xxx*** is the file name of the file to be assembled. ***compname*** is the name of the component area in the product parameter file that contains the CP parameters you want to use. Enter **cp** to build a UP system, **cpap** to build an AP system, or **cpmp** to build an MP system.

4. Repeat this sequence until the file assembles correctly.
5. When all files assemble correctly, enter the following command:

itask build *compname* noassem	***compname*** is the name of the component area in the product parameter file that contains the CP parameters you want to use. Enter **cp** to build a UP system, **cpap** to build an AP system, or **cpmp** to build an MP system.

Load Errors

- If the new nucleus fails to load:

  1. Inspect the CP load map and the virtual PSW for indications of a load error. For a list of the loader wait state codes, see *VM/SP System Messages and Codes.*
  2. Correct the error.
  3. Build the nucleus. Enter:

itask build *compname* noassem

compname is the name of the component area in the product parameter file that contains the CP parameters you want to use. Enter **cp** to build a UP system, **cpap** to build an AP system, or **cpmp** to build an MP system.

- If you shut down following a nucleus loading error and cannot re-IPL:

  1. Mount the Starter System Tape, if it is not already mounted.
  2. IPL the Starter System Tape **two times** to bypass the Device Support Facility and load the Format/Allocate program.
  3. Format the WARM START area:

```
VM/370 FORMAT/ALLOCATE PROGRAM - VM/SP
ENTER FORMAT OR ALLOCATE:
```

format

```
FORMAT FUNCTION SELECTED
ENTER DEVICE ADDRESS (CUU):
```

nnn

nnn is the address (real at first level, virtual at second level) of your VMSRES volume.

```
ENTER DEVICE TYPE:
```

devtype

devtype is the device type of the DASD. Enter **3350**, **3375**, **3380**, or **fb-512** (for the 3370, 9313, 9332, or 9335).

```
ENTER START  PAGE|CYLINDER  NUMBER OR "LABEL":
```

WARNING

After you receive this prompt, do not press ENTER before you type the start page/cylinder. If you do, you will begin formatting at page/cylinder zero and wipe out your previous work.

nnnnn

Enter the start page/cylinder according to your DASD type. See Table 3-2.

```
ENTER END  PAGE|CYLINDER  NUMBER:
```

WARNING

After you receive this prompt, do not press ENTER before you type the end page/cylinder. If you do, you will format up to the last page/cylinder and wipe out your previous work.

nnnnn

Enter the end page/cylinder according to your DASD type. See Table 3-2.

Table 3-2. Values for Formatting the Starter System WARM START and CHECKPOINT Areas after a Load Error

Device Type	WARM START Start Cyl/Page	WARM START End Cyl/Page	CHECKPOINT Start Cyl/Page	CHECKPOINT End Cyl/Page
3350	17	18	242	242
3375	25	26	380	380
3380	17	18	346	346
9313/9332	1722	1849	11692	11755
9335/3370	1722	1849	32190	32253

```
ENTER DEVICE LABEL:
```

vmsres

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

This prompt appears only for 33*nn* CKD DASD (3350, 3375, and 3380).

ENTER or **no**

The default is **no**.

```
FORMAT STARTED

FORMAT DONE
```

```
WRITE VERIFICATION WAS NOT PERFORMED
```

This message does not appear for FBA DASD.

```
0000 NO. PAGE RECORDS WITH READ-CHECK ERRORS

ENTER FORMAT OR ALLOCATE:
```

4. Repeat the previous sequence to format the CHECKPOINT area.

5. IPL the Starter System Tape to end the Format/Allocate program and load the DASD Dump/Restore (DDR) program:

```
VM/370 DASD DUMP/RESTORE PROGRAM - VM/SP
ENTER CARD READER ADDRESS OR CONTROL STATEMENTS
ENTER:
```

6. Define your input and output devices:

sysprint cons

This command directs system output to the console (virtual console at second level).

```
ENTER:
```

input *nnn* *devtype*

nnn is the address (real at first level, virtual at second level) of your tape drive.

devtype is the device type of your tape drive.

```
ENTER:
```

output *nnn* *devtype* scratch

nnn is the address (real at first level, virtual at second level) of your VMSRES volume.

devtype is the device type of your VMSRES DASD.

```
ENTER:
```

7. Restore the Starter System:

restore *allstart* *allend*

This command restores the Starter System allocation. The values of ***allstart*** and ***allend*** are determine by your VMSRES DASD type. See Table 3-3 on page 3-65.

⋮

ENTER:

restore *nucstart* *nucend*

This command restores the Starter System nucleus. The values of ***nucstart*** and ***nucend*** are determined by your VMSRES DASD type. See Table 3-3.

⋮

ENTER:

restore *dirstart* *dirend*

This command restores the Starter System CP directory. The values of ***dirstart*** and ***dirend*** are determined by your VMSRES DASD type. See Table 3-3.

⋮

ENTER:

Table 3-3. Values for Restoring the Starter System Allocation, Nucleus, and Directory after a Load Error

Device Type	***allstart*** (cyl/blk)	***allend*** (cyl/blk)	***nucstart*** (cyl/blk)	***nucend*** (cyl/blk)	***dirstart*** (cyl/blk)	***dirend*** (cyl/blk)
3350	000	000	001	005	306	308
3375	000	000	001	008	526	529
3380	000	000	001	005	531	532
9313/9332	00000	00015	00016	04415	138112	139935
9335/3370	00000	00015	00016	04415	318768	320591

8. (Second level only) Obtain support for your second level terminals (see page 3-30.)
9. IPL your VMSRES volume.
10. Rewrite the CP directory:

direct vmusers direct

11. Examine the system definition files for errors. The fact that a file assembled properly does not necessarily mean that the file contains complete or correct information. The assembly procedure looks only for syntax errors.

If you **do** find errors in the system definition files, correct them, then go back to the beginning of this step on page 3-59 and reenter the following command to reassemble all of the files and rebuild the nucleus:

itask build *compname*

If you **do not** find any errors in the system files, enter the following command to rebuild the nucleus without reassembling the files:

itask build *compname* noassem

Step 8. Shut Down the Starter System CP and IPL the New CP

Overview

At this point, you shut down the Starter System CP that you have been using and IPL the new CP nucleus that you built in Step 7. The new CP recognizes the input/output devices that you configured in the DMKRIO file.

After the IPL, you are automatically logged on to the virtual machine specified by the SYSOPER keyword in the SYSOPR macro in the DMKSYS file. The default virtual machine is OPERATOR.

While you are logged on as OPERATOR, format the OPERATOR 191 minidisk. You cannot use a minidisk for CMS files unless it is initialized and formatted into fixed-size blocks.

After you format the OPERATOR 191 minidisk, log on again as MAINT. Then you can use UTILITY EXEC to save an IPLable copy of your CP nucleus on tape for recovery purposes. You might also want to define an additional copy of your CP nucleus.

First Level and Second Level Considerations

To do this step at **first level**, go to "Procedure, First Level" on page 3-68.

To do this step at **second level**, go to "Procedure, Second Level" on page 3-73.

Procedure, First Level

- ■ Shut down the Starter System:

#cp shutdown

```
SYSTEM WARM START DATA SAVED
SYSTEM SHUTDOWN COMPLETE
```

- ■ IPL the real address of your VMSRES volume (according to the directions in the *Operator's Guide* for your machine).

```
VM/SP Release 6, Service Level nnnn; created on mm/dd/yy at hh:mm:ss

It is now hh:mm:ss EDT day mm/dd/yy
Change TOD clock (YES|NO) :
```

yes or **no**

⋮

If you reply **yes**, you will receive date and time prompts:

- Enter the date in the format ***mm/dd/yy***.
- Enter the time in the format ***hh:mm:ss*** or ***hh.mm.ss***.

Check your hardware operation instructions to determine the correct method to enable the TOD clock.

```
DMKCPI971I System is uniprocessor generated
DMKUDR476I System directory loaded from volume VMSRES

DMKCPI974I No valid override file; using system defaults
```

```
DMKCPJ951I CP volid volid not mounted
```

⋮

volid refers to a CP-owned DASD volume listed in the SYSOWN macro in the DMKSYS file. The SYSOWN macro might list some volumes that you are not using in your system. Even if this volume is one that you do intend to use, the message does not indicate a problem. Remember that you assigned temporary SPSYS labels to all of your system volumes when you formatted them in Step 1. VMSRES was relabeled in Step 2; you allocate and relabel your remaining system volumes in Step 9.

```
Start ((WARM|CKPT|FORCE|COLD)(DRAIN))|(SHUTDOWN):
```

cold

Because there is no data or accounting information to be recovered, you must request a cold start.

⋮

```
AUTO LOGON *** OPERATOR  USERS=001 BY SYSTEM
```

You are now logged on to the OPERATOR virtual machine.

```
DMKCPJ952I nnnnnK system storage
```

Depending on your processor type, *nnnnn*K might be less than your actual system storage.

⋮

```
DMKCPI966I Initialization complete
```

define storage 16m

ipl 190 clear

MAINT 190 is the CMS system minidisk. If you receive a device error message, you can ignore it.

```
VM/SP REL 6 STARTER
```

Although you have IPLed the new CP, you are still using the Starter System CMS.

> **WARNING**
>
> Do not press ENTER until you type the following command.

access (nodisk

This command prevents CMS from automatically accessing the OPERATOR 191 minidisk. If CMS tried to access 191, you would receive an error message, because 191 has not been initialized.

terminal mode vm

This command allows you (as OPERATOR) to issue CMS commands.

- Format the OPERATOR 191 minidisk:

format 191 a

By default, this command formats the minidisk using the block size that is optimal for performance and data storage for the DASD type. If you want to format the minidisk with a specific block size, use the BLKSIZE option. For more information about block size defaults and options, see the description of the FORMAT command in the *VM/SP CMS Command Reference*.

```
DMSFOR603R FORMAT will erase all files on disk A(191).
Do you wish to continue? Enter 1 (YES) or 0 (NO).
```

1

```
DMSFOR605R Enter disk label:
```

label

label is a 1-to-6-character label, such as OPR191, that you assign to this minidisk. Select any label that you want.

```
Formatting disk A
```

```
nnnnn cylinders|FB-512 blocks  formatted on disk A(191)
```

nnnnn is the number of cylinders or blocks assigned to 191 in the CP directory.

- Log on to the MAINT virtual machine by doing **one** of the following:
  - Enter the **disconn** command to disconnect OPERATOR, then log on to the console as MAINT.

    or

  - Enter the **enable all** command to enable all of the devices specified in DMKRIO, then log on as MAINT at one of the enabled terminals (leaving OPERATOR logged on at the console).

Saving an IPLable CP Nucleus on Tape

After you log on as MAINT, you can save a copy of your CP nucleus on tape for recovery purposes. Invoke the UTILITY EXEC with the NUCTAPE operand:

utility nuctape

Notes:

1. To save the nucleus, you must have a scratch tape mounted on a tape drive in read/write mode attached to MAINT at virtual address 181.
2. The saved nucleus is not usable unless the **real** address of VMSRES is properly defined in DMKSYS (see page 3-53).

Defining More Than One CP Nucleus

You can improve system availability if you define and save more than one copy of your CP nucleus. Then, if the primary nucleus is damaged or unavailable, the system operator can select an alternate nucleus to IPL. For more information, see the *VM/SP Planning Guide and Reference.*

IPLing a CP Nucleus Saved on Tape

To retrieve a CP nucleus that you have saved on tape:

- If your system is down, IPL the **real** address of the tape drive (as you did when loading the stand-alone programs in Step 1 of this procedure).
- If your system is running, mount the tape at MAINT 181 and enter:

  ipl 181 clear

The new CP nucleus is written over the old CP at the address and location defined in the CP directory and DMKSYS.

Go back to page 3-68 and complete the processing shown in this step, beginning with the prompt:

```
It is now hh:mm:ss EDT day mm/dd/yy
Change TOD clock (YES|NO) :
```

Stopping and Restarting the Installation

Now that you have your new CP up and running, you can shut down your system at the completion of any of the remaining steps in the procedure. To restart, IPL your VMSRES volume and request a WARM START.

■ To continue the installation procedure, go to "Step 9. Allocate the Remaining System Volumes" on page 3-78.

Procedure, Second Level

- Shut down the Starter System:

#cp shutdown

```
SYSTEM WARM START DATA SAVED
SYSTEM SHUTDOWN COMPLETE
CP ENTERED; DISABLED WAIT PSW '000A0000 00000008'
```

- Redefine your console:

query cons

⋮

This response gives you the address of your first level virtual console. If the address matches one of the console addresses that you noted from your second level DMKRIO (in Step 6), you can skip to the next operation (■). If the address does not match, redefine the console by entering the following command.

define _vdev1_ _vdev2_

vdev1 is the address in the system response to the previous command.

vdev2 is a console address defined in your second level DMKRIO.

- IPL your VMSRES volume:

ipl _vdev_

vdev is the first level virtual address of your VMSRES volume. This address functions as a "real" address at second level.

```
VM/SP Release 6, Service Level nnnn; created on mm/dd/yy at hh:mm:ss

It is now hh:mm:ss EDT day mm/dd/yy
Change TOD clock (YES|NO) :
```

no

At second level, you do not have access to the physical hardware needed to change the TOD (Time of Day) clock.

```
DMKCPI971I System is uniprocessor generated
DMKUDR476I System directory loaded from volume VMSRES

DMKCPI974I No valid override file; using system defaults
```

```
DMKCPJ951I CP volid volid not mounted
```

volid refers to a CP-owned DASD volume listed in the SYSOWN macro in the DMKSYS file. The SYSOWN macro might list some volumes that you are not using in your system. Even if this volume is one that you do intend to use, the message does not indicate a problem. Remember that you assigned temporary SPSYS labels to all of your system volumes when you formatted them in Step 1. VMSRES was relabeled in Step 2; you relabel and allocate your remaining system volumes in Step 9.

```
Start ((WARM|CKPT|FORCE|COLD)(DRAIN))|(SHUTDOWN):
```

cold

Because there is no data or accounting information to be recovered, you must request a cold start.

```
AUTO LOGON *** OPERATOR  USERS=001 BY SYSTEM
```

You are now logged on to the OPERATOR virtual machine.

```
DMKCPJ952I nnnnnK system storage
```

Depending on your processor type, *nnnnn*K might be less than your actual system storage.

⋮

```
DMKCPI966I Initialization complete
```

define storage 16m

ipl 190 clear

MAINT 190 is the CMS system minidisk. If you receive a device error message, you can ignore it.

```
VM/SP REL 6 STARTER
```

Although you have IPLed the new CP, you are still using the Starter System CMS.

WARNING

Do not press ENTER until you type the following command.

access (nodisk

This command prevents CMS from automatically accessing the OPERATOR 191 minidisk. If CMS tried to access 191, you would receive an error message, because 191 is not initialized.

terminal mode vm

This command allows you (as OPERATOR) to issue CMS commands.

- Format the OPERATOR 191 minidisk:

format 191 a

By default, this command formats the minidisk using the block size that is optimal for performance and data storage for the DASD type. If you want to format the minidisk with a specific block size, use the BLKSIZE option. For more information about block size defaults and options, see the description of the FORMAT command in the *VM/SP CMS Command Reference*.

```
DMSFOR603R FORMAT will erase all files on disk A(191).
Do you wish to continue? Enter 1 (YES) or 0 (NO).
```

1

```
DMSFOR605R Enter disk label:
```

label

label is a 1-to-6-character label, such as OPR191, that you assign to this minidisk. Select any label that you want.

```
Formatting disk A
```

```
nnnnn cylinders|FB-512 blocks  formatted on disk A(191)
```

nnnnn is the number of cylinders or blocks assigned to 191 in the CP directory.

- Log on to the MAINT virtual machine by doing **one** of the following:
  - Enter the **#cp disconn** command to disconnect OPERATOR, then log on to the same device as MAINT.

    or

  - Log on as MAINT at a second terminal (leaving OPERATOR logged on at the primary terminal). To do this, you must:

    1. Enter the following commands **from the primary terminal** to define and enable the second terminal. In these commands, ***vdev*** is the **first level**

virtual address of the graphic device (terminal) that you want to use. This address functions as a "real" address at second level.

> ATTN
> **define graf *vdev***
> ATTN
> **vary on *vdev***
> **enable *vdev***

2. Enter the following command **from the second terminal** to identify the terminal to your system. In this command, ***userid*** is your **first level** user ID. This completes the link between the graphic device and your first level virtual machine.

   dial *userid*

3. Log on to the second terminal as MAINT.

Saving an IPLable CP Nucleus on Tape

After you log on as MAINT, you can save a copy of your CP nucleus on tape for recovery purposes. Invoke the UTILITY EXEC with the NUCTAPE operand:

utility nuctape

Notes:

1. To save the nucleus, you must have a scratch tape mounted on a tape drive in read/write mode attached to MAINT at virtual address 181.
2. The saved nucleus is not usable unless the **real** address of VMSRES is properly defined in DMKSYS (see page 3-53).

Defining More Than One CP Nucleus

You can improve system availability if you define and save more than one copy of your CP nucleus. Then, if the primary nucleus is damaged or unavailable, the system operator can select an alternate nucleus to IPL. For more information, see the *VM/SP Planning Guide and Reference*.

Stopping and Restarting the Installation

Now that you have your new CP up and running, you can shut down your second level system at the completion of any of the remaining steps in the procedure. To restart, IPL your VMSRES volume and request a WARM START.

IPLing a CP Nucleus Saved on Tape

To retrieve a CP nucleus that you have saved on tape:

- If your system is down, IPL the **real** address of the tape drive (as you did when loading the stand-alone programs in Step 1 of this procedure).
- If your system is running, mount the tape at MAINT 181 and enter:

  ipl 181 clear

The new CP nucleus is written over the old CP at the address and location defined in the CP directory and DMKSYS.

Go back to page 3-73 and complete the processing shown in this step, beginning with the prompt:

```
It is now hh:mm:ss EDT day mm/dd/yy
Change TOD clock (YES|NO) :
```

■ To continue the installation procedure, go to "Step 9. Allocate the Remaining System Volumes" on page 3-78.

Step 9. Allocate the Remaining System Volumes

Overview

Before you load the remaining files from the product tape, you need to allocate the space on your system volumes (except VMSRES, which was allocated in Step 2 when the Starter System was restored). To do this, you invoke the ITASK EXEC with the ALLOCATE operand. ITASK:

- Looks at the $ALLOC$ user ID in the CP directory to obtain the permanent system volume labels
- Prompts you for the addresses of the volumes that you want to allocate
- Calls the IPL FMT program (a version of the stand-alone Format/Allocate program), located on the MAINT 190 minidisk.

  IPL FMT allocates the volumes using information contained in the CP directory, and uses the permanent volume labels from $ALLOC$ to replace the temporary SPSYS labels that you assigned in Step 1.

First Level and Second Level Considerations

To do this step at **first level**, go to "Procedure, First Level" on page 3-79.

To do this step at **second level**, go to "Procedure, Second Level" on page 3-81.

Procedure, First Level

Note: Do this step while logged on to the MAINT virtual machine.

- Mark your DASD volumes available to CP (they were marked unavailable by the shutdown in Step 8). Enter the following command for each volume that you formatted in Step 1 (except VMSRES):

vary online ***rdev***

rdev is the real address of the volume.

- Allocate the DASD volumes:

itask allocate

```
DMSWTK981R What is the real address of your volid volume?
Type: real address or SKIP
```

You are prompted for each DASD volume listed in the $ALLOC$ user ID in the CP directory. You might not be using some of these volumes in your system. Refer to your Installation Reference Worksheet for the labels and addresses of your system volumes.

rdev or **skip**

Enter the real address, ***rdev***, if you want to allocate this volume. Enter **skip** if you do not want to allocate this volume.

Note: If you are using the 9313/9332 Starter System, enter **skip** when prompted for the address of the VMPK01 volume. Because the 9313/9332 Starter System was restored to two volumes (VMSRES and VMPK01), VMPK01 was allocated during the restore.

```
DASD rdev ATTACH TO MAINT nnn
```

This message appears only if you entered an address at the previous prompt.

⋮

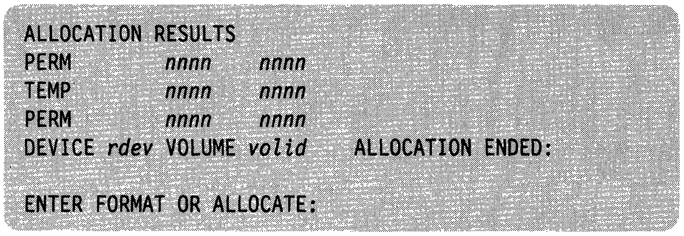
```
PUN FILE nnnn TO MAINT  COPY nnn NOHOLD
```

This message appears only once, after you have completed the previous sequence for each volume listed in $ALLOC$.

```
VM/370 FORMAT/ALLOCATE PROGRAM - VM/SP
ENTER FORMAT OR ALLOCATE:

FORMAT
```

⋮

Note: The system displays format and allocate sequences for each volume that you are allocating, even though the formatting was completed in Step 1. The temporary SPSYS label that you assigned in Step 1 is replaced with a permanent volume label obtained from the $ALLOC$ user ID.



```
ALLOCATION RESULTS
PERM        nnnn    nnnn
TEMP        nnnn    nnnn
PERM        nnnn    nnnn
DEVICE rdev VOLUME volid    ALLOCATION ENDED:

ENTER FORMAT OR ALLOCATE:
```

This is the format of the allocation results for the VMPK01 volume on a 3380 device. For other volumes and other devices, the format could vary.

■ Go to "Step 10. Shut Down and Re-IPL" on page 3-83.

Procedure, Second Level

Note: Do this step while logged on to the MAINT virtual machine.

■ Mark your DASD volumes available to CP (they were marked unavailable by the shutdown in Step 8). Enter the following command for each volume that you formatted in Step 1 (except VMSRES):

vary online _vdev_

vdev is the first level virtual address of the volume, which functions as a "real" address at second level.

■ Allocate the DASD volumes:

itask allocate

```
DMSWTK981R What is the real address of your volid volume?
Type: real address or SKIP
```

You are prompted for each DASD volume listed in the $ALLOC$ user ID in the CP directory. You might not be using some of these volumes in your system. Refer to your Installation Reference Worksheet for the labels and addresses of your system volumes.

vdev or **skip**

Remember that the "real" address at second level is the first level virtual address of the volume. Enter the "real" address, _vdev_, if you want to allocate this volume. Enter **skip** if you do not want to allocate this volume.

Note: If you are using the 9313/9332 Starter System, enter **skip** when prompted for the address of the VMPK01 volume. Because the 9313/9332 Starter System was restored to two volumes (VMSRES and VMPK01), VMPK01 was allocated during the restore.

```
DASD vdev ATTACH TO MAINT nnn
```

This message appears only if you entered an address at the previous prompt.

⋮

```
PUN FILE nnnn TO MAINT  COPY nnn NOHOLD
```

This message appears only once, after you have completed the previous sequence for each volume listed in $ALLOC$.

```
VM/370 FORMAT/ALLOCATE PROGRAM - VM/SP
ENTER FORMAT OR ALLOCATE:

FORMAT
```

⋮

Note: The system displays format and allocate sequences for each volume that you are allocating, even though the formatting was completed in Step 1. The temporary SPSYS label that you assigned in Step 1 is replaced with a permanent volume label obtained from the $ALLOC$ user ID.

```
ALLOCATION RESULTS
PERM       nnnn    nnnn
TEMP       nnnn    nnnn
PERM       nnnn    nnnn
DEVICE vdev VOLUME volid    ALLOCATION ENDED

ENTER FORMAT OR ALLOCATE:
```

This is the format of the allocation results for the VMPK01 volume on a 3380 device. For other volumes and other devices, the format could vary.

■ Go to "Step 10. Shut Down and Re-IPL" on page 3-83.

Step 10. Shut Down and Re-IPL

Overview

At this point, you do a shutdown and re-IPL to designate as "CP-owned" the system volumes that you allocated in Step 9.

The re-IPL automatically logs you on to the virtual machine specified by the SYSOPER keyword in the SYSOPR macro in the DMKSYS file. The default virtual machine is OPERATOR.

After the re-IPL, you must log on again as MAINT to continue the installation procedure.

Procedure

- Shut down the system:

#cp shutdown reipl

This command performs an automatic warm start.

```
VM/SP system restart due to shutdown REIPL
```

⋮

```
AUTO LOGON *** OPERATOR  USERS=001 BY SYSTEM
```

You are now logged on to the OPERATOR virtual machine. However, if you previously disconnected OPERATOR (at the end of Step 8), OPERATOR is immediately disconnected by the system.

```
DMKCPJ952I nnnnnK system storage
```

Depending on your processor type, *nnnnn*K might be less than your actual system storage.

⋮

```
DMKCPI966I Initialization complete
```

- Log on to the MAINT virtual machine:
  - If OPERATOR is disconnected, log on as MAINT at the primary console or at the previously defined secondary console.
  - If OPERATOR is **not** disconnected, do **one** of the following:
    - At the primary console, enter the **#cp disconn** command to disconnect the OPERATOR virtual machine, then log on to the same device as MAINT.

      or

    - Leave OPERATOR logged on at the primary console and log on as MAINT at the previously defined secondary console.
- Go to "Step 11. Load CMS and Procedures Language/VM; Build and Save CMS" on page 3-85.

Step 11. Load CMS and Procedures Language/VM; Build and Save CMS

Overview

To load CMS and Procedures Language/VM, and then build and save CMS, you invoke the ITASK EXEC with the LOAD CMS operands. ITASK:

- Calls the SPLOAD EXEC to load the following tape files from the product tape to the destinations specified in the SPLOAD PROFILE:
  - CMS BASE to the MAINT 193 minidisk

    This tape file contains text files and other items used primarily by system programmers. It also includes the SFS sample files.
  - REXX DELTA to the MAINT 293 minidisk

    This tape file contains Procedures Language/VM delta service code (update files and serviced text decks).
  - REXX APPLY to the MAINT 392 minidisk

    This tape file contains Procedures Language/VM apply service code (auxiliary control files and update shells).
  - REXX OBJECT to the MAINT 193 minidisk

    This tape file contains Procedures Language/VM text files.
  - CMS DELTA to the MAINT 293 minidisk

    This tape file contains CMS delta service code (update files and serviced text decks).
  - CMS APPLY to the MAINT 392 minidisk

    This tape file contains CMS apply service code (auxiliary control files and update shells).
  - CMS SYSTEM to the MAINT 190 minidisk

    This tape file contains execs and modules (commands) available to the general CMS user. It also includes the VMLIB callable services library.
- Calls the ASMGEND EXEC to build the system assembler and create the auxiliary directory.
- Calls the VMFASM EXEC to assemble the CMS nucleus generation profile (DMSNGP ASSEMBLE).
- Calls the VMFBLD EXEC to build and save the CMS nucleus using the information contained in DMSNGP and in the VM/SP product parameter file (5664167E $PPF).

After you build and save the nucleus, you can save and print the CMS load map, which contains the storage addresses of the nucleus control sections and entry points.

If you want to apply preventive service, corrective service, or local service to CMS, you can do so at the end of this step.

First Level and Second Level Considerations

To do this step at **first level**, go to “Procedure, First Level” on page 3-87.

To do this step at **second level**, go to “Procedure, Second Level” on page 3-94.

Procedure, First Level

Note: Do this step while logged on to the MAINT virtual machine.

- Re-attach the tape drive where the product tape is mounted (it was detached by the shutdown in Step 8):

attach *rdev* maint 181

rdev is the real address of the tape drive. The SPLOAD EXEC requires that the tape drive be attached to MAINT at virtual address 181.

```
TAPE rdev ATTACH TO MAINT 181
```

- Load the CMS and Procedures Language/VM code:

itask load cms

Notes:

1. There are pauses between some of the following messages. Before loading the first tape file to MAINT 293 or 392, the SPLOAD EXEC must format the minidisk.
2. As each minidisk is formatted, you might receive two device error messages. You can ignore these messages.

```
DMSWSL409I Loading CMS BASE to MAINT 193
DMSWSL409I Loading REXX DELTA to MAINT 293
DMSWSL409I Loading REXX APPLY to MAINT 392
DMSWSL409I Loading REXX OBJECT to MAINT 193
DMSWSL409I Loading CMS DELTA to MAINT 293
DMSWSL409I Loading CMS APPLY to MAINT 392
DMSWSL409I Loading CMS SYSTEM to MAINT 190

ASSEMBLE XF GEND PROC

ENTER TARGET DISK MODE FOR ASSEMBLE MODULES
DEFAULTS TO S-DISK IF NONE ENTERED.
```

ENTER

```
ASSEMBLE XF GEND COMPLETE
```

Note: If you receive the message `ASSEMBLE XF GEND FAILED`, the system assembler failed to build, possibly because the CMS SYSTEM files did not load properly. This could happen if you modified the SPLOAD PROFILE and changed the target minidisk for the CMS SYSTEM tape file from MAINT 190 to some other minidisk. To correct this situation, access the minidisk where you loaded the CMS SYSTEM files, enter the **asmgend** command to build the system assembler, then enter the **itask build cms** command to build the CMS nucleus.

```
DMSUPD181E No update files were found
DMSOPN002E File DMKSP MACLIB * not found
DMSWAM1907I Assembling DMSNGP
```

Note: If an error occurs dunng the assembly, examine the flagged statement(s), correct the statement(s) using the System Product Editor (XEDIT), and enter the **itask build cms** command to build the nucleus.

```
ASSEMBLER (XF) DONE
NO STATEMENTS FLAGGED IN THIS ASSEMBLY
DMSWAM1909I DMSNGP TEXT A created
```

Nucleus Generation Prompts and Responses

Note: Depending on the status of your DMSNGP file, you might receive none, some, or all of the prompts shown in this section (which ends on page 3-92). A particular prompt appears **only** if the corresponding statement in DMSNGP is missing or contains no value (and the DEFNUC macro contains no default value), or if the DMSNGP statement contains a question mark (?). If you do not receive a prompt, the system is using either the value specified in DMSNGP or the default defined in DEFNUC.

```
DMSINQ606R System disk address =
```

ENTER

190 is the default.

```
DMSINQ615R Y-disk address =
```

ENTER

19E is the default.

```
DMSINQ640R HELP disk address =
```

ENTER

19D is the default.

```
DMSINQ764R Language id =
```

ENTER or **uceng**

This response identifies the language ID (*langid*) of your system national language. The default is mixed-case American English (**AMENG**). If you want to build an uppercase English system, enter **uceng**.

```
DMSINQ293R Is this a DBCS language?  Enter 1 (YES) or 0 (NO).
```

ENTER

The default is **0** (NO); American English and uppercase English are not DBCS (Double-Byte Character Set) languages.

```
DMSINQ295R Language level id =
```

ENTER

If you load multiple versions of a national language into saved segments, the language level ID is a single alphanumeric character (A-Z, 0-9) that must match the level ID operand of the LANGGEN command used to build the language saved segment to be used with this CMS nucleus. It must also match the third character of the language saved segment name defined in the system name table (DMKSNT). S is the default.

For more information about the LANGGEN command and the language level ID, see *VM System Facilities for Programming*.

```
DMSINQ1104R File system read/write cache buffer size =
```

ENTER

This is the amount of read-ahead and write-behind buffering to be used for file input and file output operations. You can enter an amount from 1 to 28 (K). **12K** is the default.

```
DMSINQ296R Should the installation segment be used?  Enter 1 (YES) or 0 (NO).
```

The CMS installation saved segment is an optional saved segment into which you can place frequently-used EXEC files and System Product Editor (XEDIT) macros. When a user IPLs CMS, the CMS installation saved segment is connected to the user's virtual machine. Multiple users can share the same executing copy of an EXEC file or XEDIT macro stored in this saved segment.

You can install the CMS installation saved segment either as a physical saved segment or as a logical saved segment. If you install it as a logical saved segment, you can also include other objects in the saved segment, such as modules, text files, callable services libraries (CSLs), language repositories, and user-defined objects.

You install the CMS installation saved segment **after** you install your base system, but you must indicate **now** if you want to use it. For more information about installing this saved segment, see Chapter 6, "Installing Saved Segments."

ENTER (or **1**) or **0**

The default is **1** (YES). Enter **0** if you do not want to use this saved segment.

```
DMSINQ310R Installation segment name =
```

This prompt appears only if you accepted the default (or entered **1**) at the previous prompt.

ENTER or ***segname***

Enter a one-to-eight alphanumeric character name for the CMS installation saved segment, or press ENTER to accept the default name, **CMSINST**.

```
DMSINI729R Do you want to save the system?  Enter 1 (YES) or 0 (NO).
```

ENTER

The default is **1** (YES).

```
DMSINI730R Saved system name =
```

ENTER or ***sysname***

The default saved system name is **CMS**. This name must match the name defined in DMKSNT.

```
DMSINI607R Rewrite the nucleus?  Enter 1 (YES) or 0 (NO).
```

1

Enter **1** to write the CMS nucleus on the minidisk that you specify in your response to the next prompt.

```
DMSINI608R IPL device address =
```

ENTER

The default is the address of the CMS system minidisk (MAINT 190, defined by the answer to prompt DMSINQ606R).

```
DMSINI609R Nucleus (CYL or BLK) address =
```

nnnnn

nnnnn is the location on the CMS system minidisk (MAINT 190) where the new CMS nucleus is written. Enter the correct cylinder/block address for your DASD type:

Device	CYL/BLK
3350	70
3375	108
3380	70
FBA	63744

```
DMSINI610R Also IPL CYL/BLK 0? Enter 1 (YES) or 0 (NO).
```

1

Enter **1** to write the initial IPL text on cylinder/block 0 of the minidisk where the CMS nucleus is written.

The initial IPL text is a bootstrap program that reads the CMS nucleus from the cylinder/block where it is written (as defined in your response to prompt DMSINI609R). The initial IPL text is always written on the same cylinder/block as the nucleus. If the initial IPL text is not also written on cylinder/block 0, you must specify the cylinder/block address of the nucleus when you issue IPL commands for this system. For more information, see the description of the IPL command in the *VM/SP CP General User Command Reference*.

```
DMSINI611R Enter version identification:
```

The version identification is displayed each time that you IPL the CMS system you are now generating.

ENTER or ***version***

You can enter up to 32 descriptive characters to identify this version and level of CMS, or you can press ENTER to accept the default version ID:

`VM/SP Release 6` *mm/dd/yy hh:mm:ss*

```
DMSINI612R Enter installation heading:
```

The installation heading appears at the beginning of each output file created using this CMS nucleus.

ENTER or ***heading***

You can enter up to 64 descriptive characters to serve as an installation heading, or you can press ENTER to accept the default heading:

```
VM/SP CONVERSATIONAL MONITOR SYSTEM
```

End of Nucleus Generation Prompts and Responses

⋮

```
SYSTEM SAVED
DMSWSP327I The installation saved segment could not be loaded
```

The CMS installation saved segment (CMSINST) cannot be loaded into your virtual machine now because this saved segment is not installed until after you complete the base VM/SP installation.

This message does not appear if you indicated (in the DMSNGP file or in response to the DMSINQ296R prompt) that you do not want to use this saved segment.

```
VM/SP Release 6 mm/dd/yy hh:mm:ss
```

This is the default version ID. If you defined your own version ID, it appears here and each time you IPL either the CMS system minidisk (MAINT 190) or the CMS named saved system (CMS).

ENTER

Your CMS system is saved. From now on, you can enter **ipl cms** instead of IPLing the CMS system minidisk (MAINT 190).

⋮

```
Ready;
```

Saving and Printing the CMS Load Map

The CMS load map has been spooled to MAINT's virtual printer. To save the load map on the CMS base minidisk (MAINT 193), first reroute the file to MAINT's virtual reader. Enter:

spool prt nohold
close prt

When the file is sent to the reader, you receive the following message:

```
PRT FILE spoolid TO MAINT COPY 0001 NOHOLD
```

Obtain write access to the CMS base minidisk, then receive the file from the reader, using the *spoolid* from the previous message. Enter:

access 193 r
receive _spoolid_ cmsnuc map r

After you save the load map, you can print a copy of it by entering:

print cmsnuc map r

- The recommended procedure for applying service to CMS is to complete the VM/SP installation first, then apply the service. However, if you need to apply preventive service, corrective service, or local service to CMS now, use the procedures described in the *VM/SP Service Guide* to receive and/or apply the service and rebuild the CMS nucleus and the nonexecutable CMS modules. **Do not** receive or apply service for any other component. After you rebuild CMS, continue with the rest of this installation procedure.

  Note: If you apply service to CMS, you might need to rebuild serviced modules before you continue with the installation procedure.

- To continue the installation, go to "Step 12. Load IPCS and Complete the CUSTOMER PROFILE" on page 3-101.

Procedure, Second Level

Note: Do this step while logged on to the MAINT virtual machine.

- Re-attach the tape drive where the product tape is mounted (it was detached by the shutdown in Step 8):

attach *vdev* maint 181

vdev is the first level virtual address of the tape drive. At second level, this address functions as a "real" address. The SPLOAD EXEC requires that the tape drive be attached to MAINT at virtual address 181.

```
TAPE vdev ATTACH TO MAINT 181
```

- Load the CMS and Procedures Language/VM code:

itask load cms

Notes:

1. There are pauses between some of the following messages. Before loading the first tape file to MAINT 293 or 392, the SPLOAD EXEC must format the minidisk.
2. As each minidisk is formatted, you might receive two device error messages. You can ignore these messages.

```
DMSWSL409I Loading CMS BASE to MAINT 193
DMSWSL409I Loading REXX DELTA to MAINT 293
DMSWSL409I Loading REXX APPLY to MAINT 392
DMSWSL409I Loading REXX OBJECT to MAINT 193
DMSWSL409I Loading CMS DELTA to MAINT 293
DMSWSL409I Loading CMS APPLY to MAINT 392
DMSWSL409I Loading CMS SYSTEM to MAINT 190

ASSEMBLE XF GEND PROC

ENTER TARGET DISK MODE FOR ASSEMBLE MODULES
DEFAULTS TO S-DISK IF NONE ENTERED.
```

ENTER

```
ASSEMBLE XF GEND COMPLETE
```

Note: If you receive the message ASSEMBLE XF GEND FAILED, the system assembler failed to build, possibly because the CMS SYSTEM files did not load properly. This could happen if you modified the SPLOAD PROFILE and changed the destination for the CMS SYSTEM tape file from MAINT 190 to some other minidisk. To correct this situation, access the minidisk where you loaded the CMS SYSTEM files, enter the **asmgend** command to build the system assembler, then enter the **itask build cms** command to build the CMS nucleus.

```
DMSUPD181E No update files were found
DMSOPN002E File DMKSP MACLIB * not found
DMSWAM1907I Assembling DMSNGP
```

Note: If an error occurs during the assembly, examine the flagged statement(s) and correct them using the System Product Editor (XEDIT), then enter the **itask build cms** command to build the nucleus.

```
ASSEMBLER (XF) DONE
NO STATEMENTS FLAGGED IN THIS ASSEMBLY
DMSWAM1909I DMSNGP TEXT A created
```

Nucleus Generation Prompts and Responses

Note: Depending on the status of your DMSNGP file, you might receive none, some, or all of the prompts shown in this section (which ends on page 3-99). A particular prompt appears **only** if the corresponding statement in DMSNGP is missing or contains no value (and the DEFNUC macro contains no default value), or if the DMSNGP statement contains a question mark (?). If you do not receive a prompt, the system is using either the value specified in DMSNGP or the default defined in DEFNUC.

```
DMSINQ606R System disk address =
```

ENTER

190 is the default.

```
DMSINQ615R Y-disk address =
```

ENTER

19E is the default.

```
DMSINQ640R HELP disk address =
```

ENTER

19D is the default.

```
DMSINQ764R Language id =
```

ENTER or **uceng**

This response identifies the language ID (*langid*) of your system national language. The default is mixed-case American English (**AMENG**). If you want to build an uppercase English system, enter **uceng**.

```
DMSINQ293R Is this a DBCS language?  Enter 1 (YES) or 0 (NO).
```

ENTER

The default is **0** (NO); American English and uppercase English are not DBCS (Double-Byte Character Set) languages.

```
DMSINQ295R Language level id =
```

ENTER

If you load multiple versions of a national language into saved segments, the language level ID is a single alphanumeric character (A-Z, 0-9) that must match the level ID operand of the LANGGEN command used to build the language saved segment to be used with this CMS nucleus. It must also match the third character of the language saved segment name defined in the system name table (DMKSNT). S is the default.

For more information about the LANGGEN command and the language level ID, see *VM System Facilities for Programming*.

```
DMSINQ1104R File system read/write cache buffer size =
```

ENTER

This is the amount of read-ahead and write-behind buffering to be used for file input and file output operations. You can enter an amount from 1 to 28 (K). **12K** is the default.

```
DMSINQ296R Should the installation segment be used?  Enter 1 (YES) or 0 (NO).
```

The CMS installation saved segment is an optional saved segment into which you can place frequently-used EXEC files and System Product Editor (XEDIT) macros. When a user IPLs CMS, the CMS installation saved segment is connected to the user's virtual machine. Multiple users can share the same executing copy of an EXEC file or XEDIT macro stored in this saved segment.

You can install the CMS installation saved segment either as a physical saved segment or as a logical saved segment. If you install it as a logical saved segment, you can also include other objects in the saved segment, such as modules, text files, callable services libraries (CSLs), language repositories, and user-defined objects.

You install the CMS installation saved segment **after** you install your base system, but you must indicate **now** if you want to use it. For more information about installing this saved segment, see Chapter 6, "Installing Saved Segments."

ENTER (or **1**) or **0**

The default is **1** (YES). Enter **0** if you do not want to use this saved segment.

```
DMSINQ310R Installation segment name =
```

This prompt appears only if you accepted the default (or entered **1**) at the previous prompt.

ENTER or ***segname***

Enter a one-to-eight alphanumeric character name for the CMS installation saved segment, or press ENTER to accept the default name, **CMSINST**.

```
DMSINI729R Do you want to save the system?  Enter 1 (YES) or 0 (NO).
```

ENTER

The default is **1** (YES).

```
DMSINI730R Saved system name =
```

ENTER or ***sysname***

The default saved system name is **CMS**. This name must match the name defined in DMKSNT.

```
DMSINI607R Rewrite the nucleus?  Enter 1 (YES) or 0 (NO).
```

1

Enter **1** to write the CMS nucleus on the minidisk that you specify in your response to the next prompt.

```
DMSINI608R IPL device address =
```

ENTER

The default is the address of the CMS system minidisk (MAINT 190, defined by the answer to prompt DMSINQ606R).

```
DMSINI609R Nucleus (CYL or BLK) address =
```

nnnnn

nnnnn is the location on the CMS system minidisk (MAINT 190) where the new CMS nucleus is written. Enter the correct cylinder/block address for your DASD type:

Device	CYL/BLK
3350	70
3375	108
3380	70
FBA	63744

```
DMSINI610R Also IPL CYL/BLK 0?  Enter 1 (YES) or 0 (NO).
```

1

Enter **1** to write the initial IPL text on cylinder/block 0 of the minidisk where the CMS nucleus is written.

The initial IPL text is a bootstrap program that reads the CMS nucleus from the cylinder/block where it is written (as defined in your response to prompt DMSINI609R). The initial IPL text is always written on the same cylinder/block as the nucleus. If the initial IPL text is not also written on cylinder/block 0, you must specify the cylinder/block address of the nucleus when you issue IPL commands for this system. For more information, see to the description of the IPL command in the *VM/SP CP General User Command Reference*.

```
DMSINI611R Enter version identification:
```

The version identification is displayed each time you IPL the CMS system you are now generating.

ENTER or *version*

You can enter up to 32 descriptive characters to identify this version and level of CMS, or you can press ENTER to accept the default version ID:

```
VM/SP Release 6 mm/dd/yy hh:mm:ss
```

```
DMSINI612R Enter installation heading:
```

The installation heading appears at the beginning of each output file created using this CMS nucleus.

ENTER or *heading*

You can enter up to 64 descriptive characters to serve as an installation heading, or you can press ENTER to accept the default heading:

```
VM/SP CONVERSATIONAL MONITOR SYSTEM
```

End of Nucleus Generation Prompts and Responses

⋮

```
SYSTEM SAVED

DMSWSP327I The installation saved segment could not be loaded
```

The CMS installation saved segment (CMSINST) cannot be loaded into your virtual machine now because this saved segment is not installed until after you complete the base VM/SP installation.

This message does not appear if you indicated (in the DMSNGP file or in response to the DMSINQ296R prompt) that you do not want to use this saved segment.

```
VM/SP Release 6 mm/dd/yy hh:mm:ss
```

This is the default version ID. If you entered your own version ID, it appears here and each time you IPL either the CMS system minidisk (MAINT 190) or the CMS named saved system (CMS).

ENTER

Your CMS system is saved. From now on, you can enter **ipl cms** instead of IPLing the CMS system minidisk (MAINT 190).

⋮

```
Ready;
```

Saving and Printing the CMS Load Map

The CMS load map has been spooled to MAINT's virtual printer. To save the load map on the CMS base minidisk (MAINT 193), first reroute the file to MAINT's virtual reader. Enter:

spool prt nohold
close prt

When the file is sent to the reader, you receive the following message:

```
PRT FILE spoolid TO MAINT COPY 0001 NOHOLD
```

Obtain write access to the CMS base minidisk, then receive the file from the reader, using the *spoolid* from the previous message. Enter:

access 193 r
receive *spoolid* cmsnuc map r

After you save the load map, you can print a copy of it by entering:

print cmsnuc map r

- The recommended procedure for applying service to CMS is to complete the VM/SP installation first, then apply the service. However, if you need to apply preventive service, corrective service, or local service to CMS now, use the procedures described in the *VM/SP Service Guide* to receive and/or apply the service and rebuild the CMS nucleus and the nonexecutable CMS modules. **Do not** receive or apply service for any other components. After you rebuild CMS, continue with the rest of this installation procedure.

  Note: If you apply service to CMS, you might need to rebuild serviced modules before you continue with the installation procedure.

- To continue the installation, go to "Step 12. Load IPCS and Complete the CUSTOMER PROFILE" on page 3-101.

Step 12. Load IPCS and Complete the CUSTOMER PROFILE

Overview

To load the Interactive Problem Control System (IPCS), you invoke the ITASK EXEC with the LOAD IPCS operands. ITASK calls the SPLOAD EXEC to load the following tape files from the product tape to the destinations specified in the SPLOAD PROFILE:

- IPCS DELTA to the MAINT 293 minidisk

  This tape file contains IPCS delta service code (update files and serviced text decks).

- IPCS APPLY to the MAINT 392 minidisk

  This tape file contains IPCS apply service code (auxiliary control files and update shells).

- IPCS OBJECT to the MAINT 193 minidisk

  This tape file contains the IPCS object code.

One of the IPCS object files is named CUSTOMER PROFILE. This file contains all of the standard fields that you need to submit authorized program analysis reports (APARs). As loaded from the product tape, these fields contain sample data to show the type of information required. You should edit the file and replace the sample entries with data that describes your installation.

If you need additional information to complete the CUSTOMER PROFILE, contact your IBM Branch Office.

For information about using IPCS, see the *VM Interactive Problem Control System Guide and Reference.*

Procedure

- Load the IPCS code:

itask load ipcs

```
DMSWSL409I Loading IPCS DELTA to MAINT 293
DMSWSL409I Loading IPCS APPLY to MAINT 392
DMSWSL409I Loading IPCS OBJECT to MAINT 193

Ready;
```

- Edit the CUSTOMER PROFILE:

xedit customer profile

```
CUSTOMER NAME = AMERICAN CUSTOMER COMPANY   CUST. NO. = 12345-67
CUST. ADDRESS = 123 ANYPLACE AVENUE
                ANYWHERE, NY  12345
CUST. CONTACT = JOHN A. DOE
CUST. PHONE   = 123-456-7890
--------------------------------------------------------------------------------
I.B.M. REP    = SERVICE REP. NAME           EMPLOYEE NO. = 000000
I.B.M. ADDRESS= IBM CORPORATION
                STREET ADDRESS
                CITY, STATE   ZIP CODE
--------------------------------------------------------------------------------
REGION = 01  B/O = 123  WTC COUNTRY NO. =        NAME =
ITPS CODE = BROA  B/O TEL. = 123-456-7890   TIE LINE = 8-123-7890
--------------------------------------------------------------------------------
```

Replace the entries in the sample file (shown above) with data that describes your installation.

- Save the edited file:

file

- Go to “Step 13. Format the Remaining Minidisks in the Base CP Directory” on page 3-103.

Step 13. Format the Remaining Minidisks in the Base CP Directory

Overview

The following minidisks in the base CP directory do not have code loaded to them or are not associated with any particular component:

- AUTOLOG1 191
- AVSVM 191
- CMSBATCH 195
- CMSUSER 191
- EREP 191
- MAINT 201, 319, 595
- OPERATNS 191, 193
- VMSERVS 191
- VMSERVU 191

You cannot use these minidisks for CMS files until you initialize them and format them into fixed-size blocks. You can invoke the ITASK EXEC to assist you. ITASK calls the CMS FORMAT command to format the minidisks.

Procedure

Note: To complete this step, you must have already modified the CP directory (in Step 6) to provide non-restricted passwords for the virtual machines that "own" these minidisks. You cannot format a minidisk for a virtual machine that has a NOLOG password in the directory.

■ Format the base CP minidisks:

```
itask baseids
  :
```

As each minidisk is formatted, you might receive a device error message. You can ignore this message.

```
Ready;
```

■ Go to "Step 14. Build the System File Pool" on page 3-104.

Step 14. Build the System File Pool

Overview

In later steps, GCS, TSAF, and AVS code is loaded to directories defined in the system file pool. The name of the system file pool is VMSYS. The user ID for the system file pool server is VMSERVS.

To build the system file pool, you invoke the ITASK EXEC with the FILESERV VMSYS operands. ITASK:

- Accesses the MAINT 295 minidisk and obtains the logon password for the VMSERVS virtual machine and the write password for the AUTOLOG1 191 minidisk from the VMUSERS DIRECT file.
- Links to the VMSERVS 191 minidisk and copies the following files from MAINT 295:

  VMSYS POOLDEF
  VMSERVS DMSPARMS
  SERVER PROFILE

- Creates a tool called RUNSERVE EXEC on VMSERVS 191, then releases and detaches VMSERVS 191.
- Issues the AUTOLOG VMSERVS command to log the VMSERVS virtual machine on in a disconnected state.

  Note: MAINT is designated as the secondary user on the CONSOLE statement in the VMSERVS entry in the CP directory. The Single Console Image Facility allows the secondary user (MAINT) to become the controlling virtual machine when the primary user (VMSERVS) is disconnected. MAINT receives all console output from VMSERVS, and MAINT can issue commands on VMSERVS by using the CP SEND command. For more information about the Single Console Image Facility, see *VM/SP System Facilities for Programming.*

- (On VMSERVS) Accesses the MAINT 293 and MAINT 193 minidisks.
- (On VMSERVS) Invokes the RUNSERVE EXEC, which issues the FILESERV GENERATE VMSYS POOLDEF command to define and initialize the VMSYS file pool.
- (On VMSERVS) Renames SERVER PROFILE to PROFILE EXEC. This file contains the following commands:

  ```
  CP SET EMSG ON
  CP SET RUN ON
  SET AUTOREAD OFF
  ACCESS 293 B
  ACCESS 193 C
  EXEC FILESERV START
  ```

- (On VMSERVS) Issues the FILESERV START command to start the VMSERVS server machine using the start-up parameters contained in the VMSERVS DMSPARMS file. In this file, MAINT is designated as the first administrator of the VMSYS file pool.
- Links to the AUTOLOG1 191 minidisk and creates a PROFILE EXEC file that includes an AUTOLOG VMSERVS command, then releases and detaches AUTOLOG1 191.

- Issues the SET FILEPOOL VMSYS command to designate VMSYS as the default file pool for any file pool commands that MAINT issues.
- Issues the ENROLL USER command to enroll MAINT in the VMSYS file pool.
- Copies the SFS sample files from the MAINT 193 minidisk into the VMSYS:MAINT.SAMPLES directory.
- Issues the ENROLL PUBLIC command to let all users on this processor connect to the file pool.
- Issues GRANT AUTHORITY commands for PUBLIC to give read authority on the SAMPLES directory and on the individual sample files to all users on this processor.

Procedure

■ Build the system file pool:

itask fileserv vmsys

Note: The Single Console Image Facility routes console output from the disconnected VMSERVS server machine to MAINT with the prefix `VMSERVS:`.

⋮

```
VMSERVS: Ready for operator communications
Ready;
```

You might receive these messages in the reverse order.

Note: For information about enrolling users in the system file pool and other administrator tasks, see *VM/SP CMS Shared File System Administration.* However, you can wait until you complete the base VM/SP installation.

■ Go to “Step 15. Build the User File Pool (Recommended)” on page 3-106.

Step 15. Build the User File Pool (Recommended)

Overview

If you do not want to build the user file pool, go to "Step 16. Load and Save HELP (Optional)" on page 3-108.

The name of the sample user file pool is VMSYSU. The user ID for the user file pool server is VMSERVU.

To build the user file pool, you invoke the ITASK EXEC with the FILESERV VMSYSU operands. ITASK:

- Accesses the MAINT 295 minidisk and obtains the logon password for the VMSERVU virtual machine and the write password for the AUTOLOG1 191 minidisk from the VMUSERS DIRECT file.
- Links to the VMSERVU 191 minidisk and copies the following files from MAINT 295:

  ```
  VMSYSU POOLDEF
  VMSERVU DMSPARMS
  SERVER PROFILE
  ```

- Creates a tool called RUNSERVE EXEC on VMSERVU 191, then releases and detaches VMSERVU 191.
- Issues the AUTOLOG VMSERVU command to log the VMSERVU virtual machine on in a disconnected state.

  Note: MAINT is designated as the secondary user on the CONSOLE statement in the VMSERVU entry in the CP directory. The Single Console Image Facility allows the secondary user (MAINT) to become the controlling virtual machine when the primary user (VMSERVU) is disconnected. MAINT receives all console output from VMSERVU, and MAINT can issue commands on VMSERVU by using the CP SEND command. For more information about the Single Console Image Facility, see *VM/SP System Facilities for Programming*.

- (On VMSERVU) Accesses the MAINT 293 and MAINT 193 minidisks.
- (On VMSERVU) Invokes the RUNSERVE EXEC, which issues the FILESERV GENERATE VMSYSU POOLDEF command to define and initialize the VMSYSU file pool.
- (On VMSERVU) Renames SERVER PROFILE to PROFILE EXEC. This file contains the following commands:

  ```
  CP SET EMSG ON
  CP SET RUN ON
  SET AUTOREAD OFF
  ACCESS 293 B
  ACCESS 193 C
  EXEC FILESERV START
  ```

- (On VMSERVU) Issues the FILESERV START command to start the VMSERVU server machine using the start-up parameters contained in the VMSERVU DMSPARMS file. In this file, MAINT is designated as the first administrator of the VMSYSU file pool.

- Links to the AUTOLOG1 191 minidisk and creates a PROFILE EXEC file that includes an AUTOLOG VMSERVU command, then releases and detaches AUTOLOG1 191.

Procedure

■ Build the user file pool:

```
itask fileserv vmsysu
```

Note: The Single Console Image Facility routes console output from the disconnected VMSERVU server machine to MAINT with the prefix `VMSERVU:`.

```
:
VMSERVU: Ready for operator communications
Ready;
```

You might receive these messages in the reverse order.

Note: For information about enrolling users in the user file pool and other administrator tasks, see *VM/SP CMS Shared File System Administration.* However, you can wait until you complete the base VM/SP installation.

■ Go to "Step 16. Load and Save HELP (Optional)" on page 3-108.

Step 16. Load and Save HELP (Optional)

Overview

If you do not want to load the supplied system HELP files, go to "Step 17. Load, Build, and Save GCS (Optional)" on page 3-111.

To load and save HELP, you invoke the ITASK EXEC with the LOAD HELP operands. ITASK:

- Calls the SPLOAD EXEC to load the HELP FILES tape file from the product tape to the MAINT 19D minidisk.
- Asks you if you want to have the HELP files converted to uppercase (from mixed-case, as loaded). You need to convert the files if you are building an uppercase English system.
- Issues the SAVEFD command to initialize and save the HELP minidisk directory (also known as the file status table, or FST) in a physical saved segment named HELP. For more information about the function of the SAVEFD command, see the *VM/SP Application Development Guide for CMS.*

Note: After base installation, if you want to build a system that is the opposite (American English or uppercase English) of the one you are building now, you must reload the HELP files from the product tape. For more information, see Chapter 7, "Installing a New System National Language" on page 7-1.

Procedure

■ Define your virtual storage less than the address at which the HELP saved segment is to be loaded:

define storage 12m

The sample DMKSNT defines 13248K as the virtual storage location of the HELP saved segment.

```
STORAGE=12288K
CP entered;  disabled wait PSW '00020000 00000000'
```

ipl cms

```
VM/SP Release 6 mm/dd/yy hh:mm:ss
```

Or your own version ID.

ENTER

⋮

■ Load and save HELP:

itask load help

Notes:

1. If you are loading a minidisk that has not been formatted yet, the SPLOAD EXEC formats the minidisk before proceeding with the load.
2. When SPLOAD formats a minidisk, you might receive two device error messages. You can ignore these messages.
3. Because there are many HELP files, the load operation could take 15 minutes or more.

```
DMSWSL409I Loading HELP FILES to MAINT 19D

DMSWTK966R Do you wish to have the HELP files converted
           to uppercase?  Type (No) or Yes
```

ENTER (or **no**) or **yes**

The default is **no**. Enter **yes** if you are building an uppercase English system.

```
SYSTEM SAVED
```

Step 16

```
Ready;
```

The ready message indicates that the load is complete.

- Define your virtual storage back to the maximum (16M):

define storage 16m

```
STORAGE=16384K
CP entered;  disabled wait PSW '00020000 00000000'
```

ipl cms

```
VM/SP Release 6 mm/dd/yy hh:mm:ss
```

Or your own version ID.

ENTER

- Go to "Step 17. Load, Build, and Save GCS (Optional)" on page 3-111.

Step 17. Load, Build, and Save GCS (Optional)

Overview

You must install the GCS component if you plan to install SNA products or RSCS Version 2. **If you do not want to install GCS, go to "Step 18. Load TSAF (Optional)" on page 3-124.**

To load GCS, you invoke the ITASK EXEC with the LOAD GCS operands. ITASK calls the SPLOAD EXEC to load the following tape files from the product tape to the destinations specified in the SPLOAD PROFILE:

- GCS DELTA to the VMSYS:MAINT.GCS.PUTSERV directory

  This tape file contains GCS delta service code (update files and serviced text decks).

- GCS APPLY to the VMSYS:MAINT.GCS.APPLY directory

  This tape file contains GCS apply service code (auxiliary control files and update shells).

- GCS INTERFACE to the MAINT 193 minidisk

  This tape file contains files that enable GCS to interface with IPCS.

- GCS OBJECT to the VMSYS:MAINT.GCS.OBJECT directory

  This tape file contains the GCS object code, including the GROUP EXEC and GCSLOAD EXEC files.

After the code is loaded, you invoke the GROUP EXEC and complete a series of panels to build a GCS configuration file.

Note: If you do not have a full-screen display device, you cannot use the GROUP EXEC because you cannot display the panels. In that case, you must build the configuration file manually using the build macros described in the *VM/SP Group Control System Command and Macro Reference.*

When building the configuration file, refer to your Installation Reference Worksheet.

After you build the configuration file, you invoke the ITASK EXEC with the BUILD GCS operands. ITASK:

- Modifies a copy of the GCS loadlist (GCSLOAD EXEC) and changes the entry that contains the file name of the GCS configuration file (the default is GCS) to match the file name of the configuration file that you just built (the default is also GCS), then files the modified loadlist on MAINT 295
- Renames the file type of the configuration file from GROUP to ASSEMBLE
- Calls the VMFASM EXEC to assemble the configuration file
- Calls the VMFBLD EXEC to build and save the GCS nucleus.

After you build and save the nucleus, you can save and print the GCS load map, which contains the storage addresses of the nucleus control sections and entry points.

If you want to reconfigure GCS after you install it, you must modify the configuration file and rebuild the GCS nucleus. If you want to install another GCS nucleus, you must build a new configuration file. Each configuration file must have a unique name and have an entry in the system name table (DMKSNT ASSEMBLE). See "Reconfiguring GCS" on page 3-122.

Procedure

- Load the GCS code:

itask load gcs

Note: If the destination is an SFS directory that has not been created yet, the SPLOAD EXEC creates the directory before continuing with the load.

```
DMSWSL409I Loading GCS DELTA to VMSYS:MAINT.GCS.PUTSERV
DMSWSL409I Loading GCS APPLY to VMSYS:MAINT.GCS.APPLY
DMSWSL409I Loading GCS INTERFACE to MAINT 193
DMSWSL409I Loading GCS OBJECT to VMSYS:MAINT.GCS.OBJECT

Ready;
```

- Set up the GROUP EXEC messages in storage.

Notes:

1. If you are **rebuilding** a GCS nucleus, and your current configuration file does **not** require changes, you can skip this part of the step and go directly to the nucleus build operation on page 3-120.
2. If you do not have a full-screen display device, you cannot use the GROUP EXEC to build the configuration file. Therefore, **do not** set up the GROUP EXEC messages in storage. You must build the configuration file manually, using the build macros described in the *VM/SP Group Control System Command and Macro Reference*. Then continue this step with the nucleus build operation on page 3-120.

copyfile csimes text b csiume = a

This command creates a temporary GCS message file (containing the messages for the GROUP EXEC) on file mode A that has the file name (CSIUME) required by the SET LANGUAGE command.

set language ameng (add csi user

This command sets American English (AMENG) as the language of the temporary GCS message file and places the file in user storage.

- Invoke the GROUP EXEC to display the configuration panels.

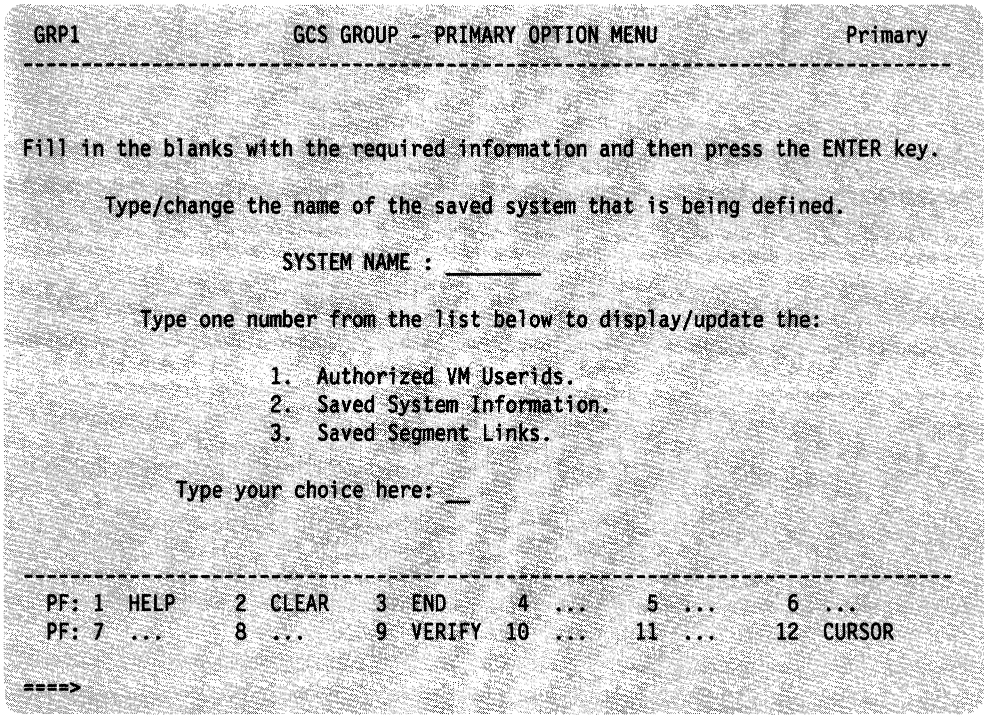
Use your Installation Reference Worksheet to help you complete the panels. For more information, see "Planning for the Group Control System (GCS)" on page 2-4.

group *systemname*

This command assigns ***systemname*** as the file name of the GCS configuration file that you are building and invokes the Primary Option Menu. ***systemname*** is optional at this time; if specified, it must match the SYSNAME entry in the DMKSNT file (the sample SYSNAME entry is **GCS**). If you specify the ***systemname*** here, the Primary Option Menu appears with the SYSTEM NAME field filled in. If you do not specify the ***systemname*** here, you must complete the SYSTEM NAME field on the Primary Option Menu.

Table 3-4. Function Keys Used with the GROUP EXEC Panels

KEY	FUNCTION
PF1 HELP	Shows information about a certain panel.
PF2 CLEAR	Clears the input areas where you enter information.
PF3 END	Leaves the present panel and returns you to a previous one. • If you press PF3 on the Primary Option Menu, you return to CMS. • If you press PF3 on any other screen, you return to the Primary Option Menu. • If you press PF3 after you enter information, but before you save it, you get the message: `577E File has been changed; type QQUIT to quit anyway`
PF5 REFRESH	Fills in the panel's input areas with the values you last saved there.
PF6 SAVE	Saves information that you entered on the panel (for the configuration file *systemname* GROUP).
PF7 PREVIOUS	Returns to the previous panel, if there is one.
PF8 NEXT	Moves ahead to the next panel, if there is one.
PF9 VERIFY	Checks to see if information that you entered on the panel also exists somewhere else (in a directory entry, for instance). If you press PF9, the GROUP EXEC runs a check and issues a message telling you how "valid" your information is.
PF12 CURSOR	Moves the cursor to the panel's command line.
PF4, PF10, PF11	Not Used
ENTER	Saves information that you entered and processes any valid CP or CMS command typed on the command line. Two specific commands you can enter are: 1. QQUIT, entered from the Primary Option Menu, returns you to CMS. If entered on any other panel, it returns you to the Primary Option Menu. 2. CANCEL, entered on any panel, returns you to CMS.



```
 GRP1                 GCS GROUP - PRIMARY OPTION MENU              Primary
------------------------------------------------------------------------------


Fill in the blanks with the required information and then press the ENTER key.

       Type/change the name of the saved system that is being defined.

                      SYSTEM NAME : ________

          Type one number from the list below to display/update the:

                     1.  Authorized VM Userids.
                     2.  Saved System Information.
                     3.  Saved Segment Links.

             Type your choice here: __


------------------------------------------------------------------------------
  PF: 1  HELP     2  CLEAR    3  END      4  ...     5  ...      6  ...
  PF: 7  ...      8  ...      9  VERIFY  10  ...    11  ...     12  CURSOR

====>
```

- Complete the Primary Option Menu:

  1. **Fill in or change the SYSTEM NAME.**
     If you invoked the GROUP EXEC with the ***systemname*** operand, this panel appears with the SYSTEM NAME already filled in.

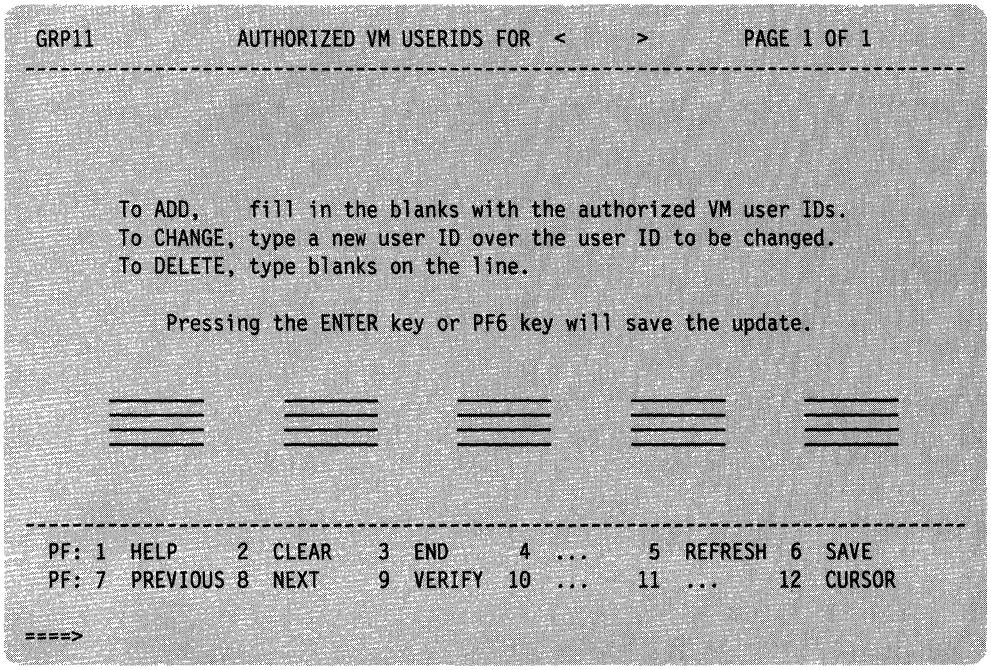
  2. **Verify your input before selecting another screen.**
     Press PF9 to verify that this SYSTEM NAME has a valid entry in your system name table (DMKSNT). You receive two messages:

     a. Tells whether the SYSTEM NAME exists
     b. Asks you to enter a selection number.

     Note: If a configuration file for this SYSTEM NAME already exists (for example, MYGCS GROUP), the GROUP EXEC makes a backup copy of it, with a different file type, before you start making changes:

MYGCS OLDGROUP	(original file, new file type)
MYGCS GROUP	(same file, ready for changes)

  3. **Select the next panel (1, 2, or 3) and press ENTER.**
     The first time you leave the Primary Option Menu, select panel 1; the next time, panel 2; the third time, panel 3.



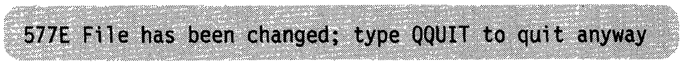
```
GRP11          AUTHORIZED VM USERIDS FOR  <        >         PAGE 1 OF 1
--------------------------------------------------------------------------------



        To ADD,    fill in the blanks with the authorized VM user IDs.
        To CHANGE, type a new user ID over the user ID to be changed.
        To DELETE, type blanks on the line.

            Pressing the ENTER key or PF6 key will save the update.


        ________      ________      ________      ________      ________
        ________      ________      ________      ________      ________
        ________      ________      ________      ________      ________
        ________      ________      ________      ________      ________


--------------------------------------------------------------------------------
 PF: 1  HELP     2  CLEAR    3  END     4  ...     5  REFRESH  6  SAVE
 PF: 7  PREVIOUS 8  NEXT     9  VERIFY  10 ...     11 ...      12 CURSOR

====>
```

■ Complete the Authorized User IDs panel:

1. **Type in the user ID(s).**
   Press ENTER or PF6 after typing each user ID to save your information and advance to the next space at the same time. Every time you press either of these keys, the GROUP EXEC tells you how many user IDs it has processed. Use PF7 and PF8 if you have more than one screenful of user IDs.

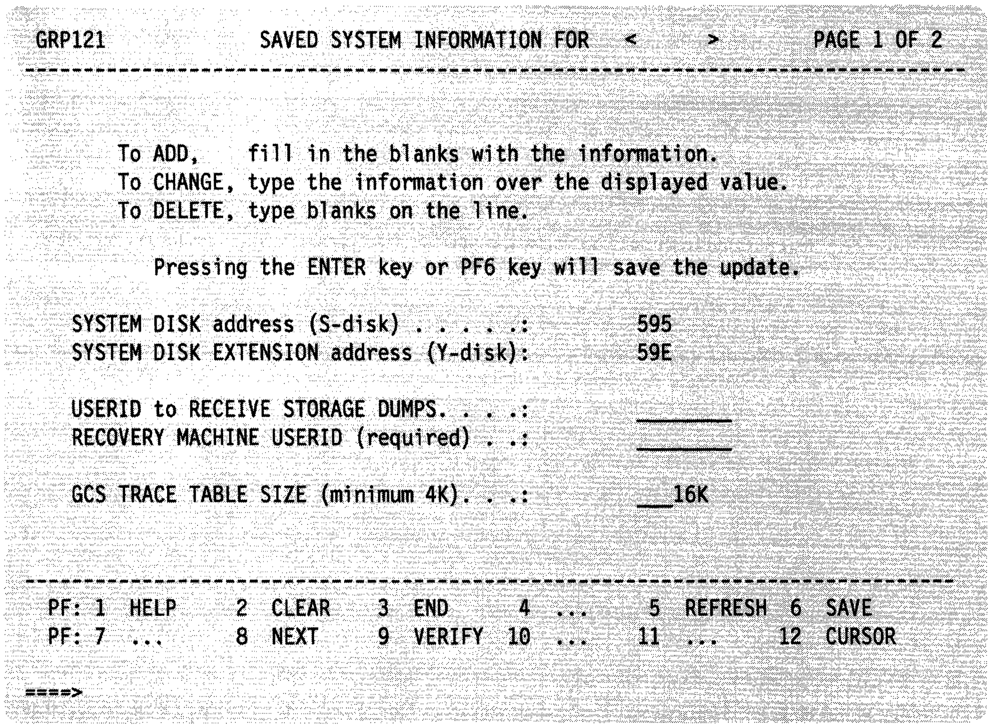
2. **Verify your input.**
   Press PF9. The GROUP EXEC tells you how many user IDs on the page being viewed are valid or invalid and places arrows beside any user IDs not found in the CP directory.

3. **Return to the Primary Option Menu.**
   Press PF3. If you forgot to press ENTER or PF6, you will remain on this screen and see the message:

   ```
   577E File has been changed; type QQUIT to quit anyway
   ```

   To continue, simply press PF6. You will receive a message telling you how many authorized user IDs have been processed. Press PF3 again to return to the Primary Option Menu.



```
 GRP121            SAVED SYSTEM INFORMATION FOR  <        >        PAGE 1 OF 2
--------------------------------------------------------------------------------


        To ADD,    fill in the blanks with the information.
        To CHANGE, type the information over the displayed value.
        To DELETE, type blanks on the line.

           Pressing the ENTER key or PF6 key will save the update.

    SYSTEM DISK address (S-disk) . . . . .:         595
    SYSTEM DISK EXTENSION address (Y-disk):         59E

    USERID to RECEIVE STORAGE DUMPS. . . .:         ________
    RECOVERY MACHINE USERID (required) . .:         ________

    GCS TRACE TABLE SIZE (minimum 4K). . .:         ___16K


--------------------------------------------------------------------------------
  PF: 1  HELP     2  CLEAR    3  END      4  ...     5  REFRESH  6  SAVE
  PF: 7  ...      8  NEXT     9  VERIFY  10  ...    11  ...     12  CURSOR

====>
```

- Complete the first Saved System Information panel:

  1. **Specify your disk addresses.**
     The default 595 and 59E virtual addresses are already saved and verified. If you must choose different addresses, type over the default values. Make sure the new information is correct; the GROUP EXEC does not prevent you from saving invalid information.

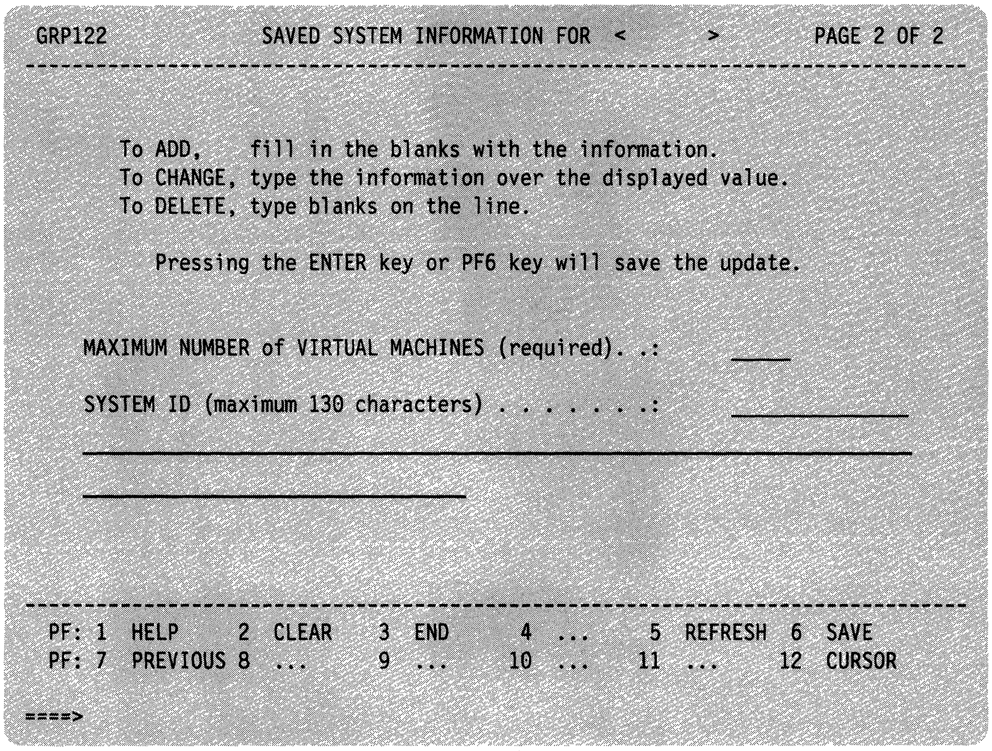
  2. **Name an authorized dump receiver.**
     Name an authorized recovery machine.
     GCS requires you to enter a recovery machine user ID. The GROUP EXEC tells you whether that user ID is valid, but it does not prevent you from saving an invalid entry.

  3. **Specify a trace table size.**
     By default, the GROUP EXEC saves a value of 16K. If you decide to save a different amount, simply type the new value over the default value.

  4. **Verify your input.**
     Press PF9 to verify that the dump receiver and recovery machine user IDs have valid entries in the CP directory. If your dump receiver also appears on the other panel of Authorized VM user IDs, the GROUP EXEC verifies it as a "Valid Authorized" user ID. Make sure any new SYSTEM DISK or TRACE TABLE information is correct; the GROUP EXEC does not prevent you from saving invalid information.

  5. **Save your input.**
     If you are satisfied with your choices, press PF6 or ENTER to save the information.

  6. **Go on to the second panel.**
     Press PF8 to continue on the second Saved System Information panel.



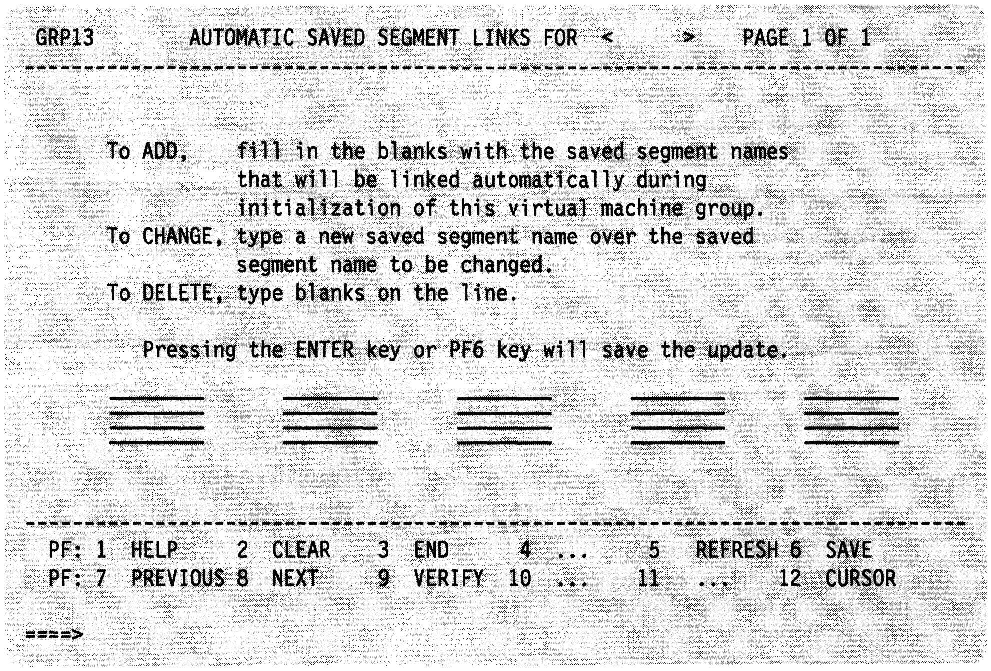
```
GRP122              SAVED SYSTEM INFORMATION FOR <        >        PAGE 2 OF 2
--------------------------------------------------------------------------------


        To ADD,    fill in the blanks with the information.
        To CHANGE, type the information over the displayed value.
        To DELETE, type blanks on the line.

           Pressing the ENTER key or PF6 key will save the update.


     MAXIMUM NUMBER of VIRTUAL MACHINES (required). .:       ____

     SYSTEM ID (maximum 130 characters) . . . . . . .:       _______________

     ______________________________________________________________________

     ________________________________



--------------------------------------------------------------------------------
 PF: 1  HELP     2  CLEAR    3  END      4  ...     5  REFRESH  6  SAVE
 PF: 7  PREVIOUS 8  ...      9  ...     10  ...    11  ...     12  CURSOR

====>
```

- Complete the second Saved System Information panel:

  1. **Specify a MAXIMUM NUMBER.**
     Simply type a value of **1** or more, and press ENTER . The GROUP EXEC responds with a message *only* if you enter an invalid value.

  2. **Type in your SYSTEM ID text.**
     This is optional. Move the cursor to the SYSTEM ID space (if it is not already there) and enter your text. The GROUP EXEC does not let you enter any more than 130 characters.

  3. **Save your input and return to the Primary Option Menu.**
     Press ENTER to save your information and then press PF3 .



```
 GRP13        AUTOMATIC SAVED SEGMENT LINKS FOR  <       >    PAGE 1 OF 1
-------------------------------------------------------------------------------


         To ADD,     fill in the blanks with the saved segment names
                     that will be linked automatically during
                     initialization of this virtual machine group.
         To CHANGE, type a new saved segment name over the saved
                     segment name to be changed.
         To DELETE, type blanks on the line.

            Pressing the ENTER key or PF6 key will save the update.

         ________       ________       ________       ________       ________
         ________       ________       ________       ________       ________
         ________       ________       ________       ________       ________
         ________       ________       ________       ________       ________



-------------------------------------------------------------------------------
 PF: 1  HELP     2  CLEAR    3  END      4  ...      5  REFRESH 6  SAVE
 PF: 7  PREVIOUS 8  NEXT     9  VERIFY  10  ...     11  ...     12  CURSOR

====>
```

■ Complete the Automatic Saved Segment Links panel:

Note: Do not include CMSBAM or CMSVSAM here. For more information about these VSAM segments, see Chapter 6.

1. **Type in the segment name(s).**
   Press ENTER or PF6 after typing each segment name to save your information and advance to the next space at the same time. Every time you press either of these keys, the EXEC tells you how many segment names it has processed. Use PF7 and PF8 if you more than one screen of names.

2. **Verify your input.**
   Press PF9. The GROUP EXEC tells you how many segment names on the page being viewed are valid or invalid and places arrows beside any names it does not find in your system name table (DMKSNT).

3. **Return to the Primary Option Menu.**
   If you have not saved your input yet, press PF6. Press PF3 to return to the Primary Option Menu.

- After you provide the necessary input on all panels and return to the Primary Option Menu for the last time, press PF3 to exit from the GROUP EXEC.
- Remove the temporary GCS message file from user storage and erase it from file mode A:

set language ameng (delete csi user

erase csiume text a

- Build the GCS nucleus:

itask build gcs ***systemname***

systemname is the file name of the GCS configuration file that you just built. If you do not specify a name, the default is **GCS.**

```
DMSUPD181E No update files were found
DMSOPN002E File DMKSP MACLIB * not found
DMSWAM1907I Assembling systemname
```

⋮

If You Receive an Assembly Error

1. Reestablish the nucleus build access order:

   vmfsetup 5664167e gcs (access bld

2. Check your configuration file (which ITASK has renamed to *systemname* ASSEMBLE) for incorrect or incomplete entries. For information about the required fields, see "Planning for the Group Control System (GCS)" on page 2-4.
3. Correct the configuration file or go through the GROUP EXEC panels again to rebuild the file.

   Note: If you rebuild the configuration file, you must use the same *systemname*. After you rebuild the configuration file, you must rename or erase the old *systemname* ASSEMBLE file and then rename the file type of the new configuration file from GROUP to ASSEMBLE.

4. Assemble the configuration file:

   vmfasm ***systemname*** **5664167e gcs (ppf**

5. Build the nucleus:

   vmfbld 5664167e gcs gcsload

⋮

```
MSG FROM MAINT  :  CSIINI134I systemname has nnnnnn bytes of available
common free storage
SYSTEM SAVED
```

```
PRT FILE spoolid TO MAINT COPY 001 NOHOLD
```

This message indicates that the GCS load map file has been sent to MAINT's virtual reader. If you plan to save or print the GCS load map, record the *spoolid*.

```
SYSTEM RESET
DMKDSP450W CP entered;  disabled wait PSW '00020000 00000000'
```

- IPL CMS:

ipl cms

```
VM/SP Release 6 mm/dd/yy hh:mm:ss
```

Or your own version ID.

ENTER

```
Ready;
```

Saving and Printing the GCS Load Map

The GCS load map has been sent to MAINT's virtual reader. To save the load map in the GCS object directory (VMSYS:MAINT.GCS.OBJECT), enter:

access vmsys:maint.gcs.object e
receive *spoolid* gcsnuc map e

After you save the load map, you can print a copy of it by entering:

print gcsnuc map e

- When you complete your GCS installation, go to "Step 18. Load TSAF (Optional)" on page 3-124.

Reconfiguring GCS

After you install GCS, you might want to alter the GCS configuration:

- To modify GCS (to add new authorized users, for example), you must modify the existing GCS configuration file and rebuild the GCS nucleus.
- To move or enlarge GCS, see Appendix H, "Example of Alternate GCS Nucleus Placement" on page H-1.
- To install multiple GCS nuclei, you must build a new configuration file for each additional nucleus. Additional systems must be defined in DMKSNT and in the CP directory.

To modify an existing system or add a new system, you can follow a similar sequence of steps. If you are modifying your system, do only the steps marked with an asterisk (*). If you are installing a new system, do only the steps marked with a plus (+).

1. *+Reestablish the nucleus build access order:

   ```
   vmfsetup 5664167e gcs (access bld
   ```

2. *+(Full screen display devices only) Set up the GROUP EXEC messages in storage:

   ```
   copyfile csimes text b csiume = a
   set language ameng (add csi user
   ```

3. *Rename the file type of the existing configuration file:

   ```
   rename systemname assemble a = group =
   ```

4. *+Build or modify the configuration file:

   - If you are using a full screen display device, enter:

     ```
     group systemname
     ```

     and complete the panels to define the system parameters (see pages 3-114 through 3-119). If this a configuration file for an additional nucleus, _systemname_ must be unique.

   - If you are **not** using a full screen display device, you must build the configuration file manually, using the build macros described in the *VM/SP Group Control System Command and Macro Reference.*

5. *+(Full screen display devices only) Remove the temporary GCS message file from user storage and erase it from file mode A:

   ```
   set language ameng (delete csi user
   erase csiume text a
   ```

6. +Edit the GCS loadlist:

   ```
   xedit gcsload exec
   ```

   and change the configuration file entry (&1 &2 &3 *fn*, where *fn* is the file name of the configuration file for the previously-generated GCS nucleus) to specify the file name of the configuration file for this nucleus.

7. *+Rename the file type of the configuration file:

   ```
   rename systemname group a = assemble =
   ```

8. * + Assemble the configuration file:

   vmfasm ***systemname*** **5664167e gcs (ppf**

9. * + Build and save the GCS nucleus:

   vmfbld 5664167e gcs gcsload

10. * + Save or print the GCS load map, as indicated on page 3-121. Remember to give each load map file a different name.

Step 18. Load TSAF (Optional)

Overview

If you do not want to load the TSAF component, go to "Step 19. Load AVS (Optional)" on page 3-125.

To load the Transparent Services Access Facility (TSAF), you invoke the ITASK EXEC with the LOAD TSAF operands. ITASK calls the SPLOAD EXEC to load the following tape files from the product tape to the destinations specified in the SPLOAD PROFILE:

- TSAF DELTA to the VMSYS:MAINT.TSAF.PUTSERV directory

  This tape file contains TSAF delta service code (update files and serviced text decks).

- TSAF APPLY to the VMSYS:MAINT.TSAF.APPLY directory

  This tape file contains TSAF apply service code (auxiliary control files and update shells).

- TSAF INTERFACE to the MAINT 193 minidisk

  This tape file contains files that enable TSAF to interface with IPCS.

- TSAF OBJECT to the VMSYS:MAINT.TSAF.OBJECT directory

  This tape file contains the ready-to-run TSAF module. No build is required.

For more information about TSAF, see *VM/SP Connectivity Planning, Administration, and Operation.*

Procedure

■ Load the TSAF code:

itask load tsaf

Note: If the destination is an SFS directory that has not been created yet, the SPLOAD EXEC creates the directory before continuing with the load.

```
DMSWSL409I Loading TSAF DELTA to VMSYS:MAINT.TSAF.PUTSERV
DMSWSL409I Loading TSAF APPLY to VMSYS:MAINT.TSAF.APPLY
DMSWSL409I Loading TSAF INTERFACE to MAINT 193
DMSWSL409I Loading TSAF OBJECT to VMSYS:MAINT.TSAF.OBJECT

Ready;
```

■ Go to "Step 19. Load AVS (Optional)" on page 3-125.

Step 19. Load AVS (Optional)

Overview

If you do not want to load the AVS component, go to "Step 20. Load Source Code (Optional)" on page 3-126.

To load APPC/VM VTAM Support (AVS), you invoke the ITASK EXEC with the LOAD AVS operands. ITASK:

- Calls the SPLOAD EXEC to load the following tape files from the product tape to the destinations specified in the SPLOAD PROFILE:
  - AVS DELTA to the VMSYS:MAINT.AVS.PUTSERV directory

    This tape file contains AVS delta service code (update files and serviced text decks).
  - AVS APPLY to the VMSYS:MAINT.AVS.APPLY directory

    This tape file contains AVS apply service code (auxiliary control files and update shells).
  - AVS INTERFACE to the MAINT 193 minidisk

    These files enable AVS to interface with IPCS.
  - AVS OBJECT to the VMSYS:MAINT.AVS.OBJECT directory

    This tape file contains the AVS object code.
- Issues the COPYFILE (REPLACE command to copy the following files from the VMSYS:MAINT.AVS.OBJECT directory to the AVSVM 191 minidisk:
  - AGW LOADLIB
  - PROFILE GCS.

For more information about AVS, see *VM/SP Connectivity Planning, Administration, and Operation.*

Procedure

■ Load the AVS code:

itask load avs

Note: If the destination is an SFS directory that has not been created yet, the SPLOAD EXEC creates the directory before continuing with the load.

```
DMSWSL409I Loading AVS DELTA to VMSYS:MAINT.AVS.PUTSERV
DMSWSL409I Loading AVS APPLY to VMSYS:MAINT.AVS.APPLY
DMSWSL409I Loading AVS INTERFACE to MAINT 193
DMSWSL409I Loading AVS OBJECT to VMSYS:MAINT.AVS.OBJECT

Ready;
```

■ Go to "Step 20. Load Source Code (Optional)" on page 3-126.

Step 20. Load Source Code (Optional)

Overview

Source code for the CP and IPCS components and source code for some CMS modules is provided on the product tape. Source code for the GCS and TSAF components and additional CMS modules is provided on a separately-orderable feature tape. **If you do not want to load source, go to "Step 21. Update the SYSTEM NETID File (Optional)" on page 3-128.**

To load the source code, you invoke the ITASK EXEC with specific operands to identify the source tape files that you want to load. ITASK calls the SPLOAD EXEC to load the specified source tape files to the destinations defined in the SPLOAD PROFILE:

Tape File	Destination
CP SOURCE	MAINT 394
CMS SOURCE (product tape)	MAINT 393
IPCS SOURCE	MAINT 393
GCS SOURCE	VMSYS:MAINT.GCS.SOURCE
TSAF SOURCE	VMSYS:MAINT.TSAF.SOURCE
CMSFT SOURCE (feature tape)	MAINT 393

All source files are loaded down in **packed** format. After the files are loaded, you can unpack individual files by using the COPYFILE command with the UNPACK option. For more information about this option, see the *VM/SP CMS Command Reference*. The MAINT 394 and 393 minidisks are defined on the VMPK04 volume with space for all of the packed files plus some (but not all) unpacked modules. If you plan to unpack all of your files, you should redefine these minidisks 80% larger.

Procedure

- Load the source tape files from the product tape:

itask load *group*src

group refers to the GROUP portion of the tape file identifier (GROUP ELEMENT) used in the SPLOAD PROFILE: **cp**, **cms**, or **ipcs**.

Notes:

1. If you are loading a minidisk that has not been formatted yet, the SPLOAD EXEC formats the minidisk before continuing with the load.
2. When SPLOAD formats a minidisk, you might receive two device error messages. You can ignore these messages.

```
DMSWSL409I Loading group SOURCE to MAINT vdev
```

- Enter the previous command for each source tape file that you want to load.

Source Feature Tape Installation

- Remove the product tape and mount the source feature tape.
- Load the source tape files from the feature tape:

itask load *group*src

group refers to the GROUP portion of the tape file identifier (GROUP ELEMENT) used in the SPLOAD PROFILE: **gcs**, **tsaf**, or **cmsft**.

Note: If the destination is an SFS directory that has not been created yet, the SPLOAD EXEC creates the directory before continuing with the load.

```
DMSWSL409I Loading group SOURCE to dirname|userid vdev
```

- Enter the previous command for each source tape file that you want to load.

End of Source Feature Tape Installation

- When you complete the load, go to "Step 21. Update the SYSTEM NETID File (Optional)" on page 3-128.

Step 21. Update the SYSTEM NETID File (Optional)

Overview

If you created an RSCS (Remote Spooling Communications Subsystem) networking virtual machine in the CP directory (in Step 6), you should now update the SYSTEM NETID file located on the MAINT 190 minidisk. **If you do not need to update SYSTEM NETID, go to "Step 22. Install EREP (Optional)" on page 3-131.**

The records in the SYSTEM NETID file have the format:

```
cpuid nodeid netid
```

where:

cpuid
is the CPU serial number

nodeid
is the local node ID of the RSCS virtual machine, as defined in the LOCAL statement of the RSCS configuration file on the RSCS 191 (MAINT 59F) minidisk

netid
is the user ID of the RSCS virtual machine, as defined in the CP directory.

When you issue commands to communicate across the network, the SYSTEM NETID file is referenced as follows:

1. To transmit notes, files, and messages, the NOTE, SENDFILE, and TELL commands issue the IDENTIFY command.
2. The IDENTIFY command issues the QUERY CPUID command to retrieve the CPU serial number, then searches the SYSTEM NETID file for a matching serial number.
3. The IDENTIFY command issues the QUERY USERID command to retrieve the node identification, then compares it to the node in the SYSTEM NETID record.

   If there is a conflict in nodes between the SYSTEM NETID file and the response from QUERY USERID, the node in SYSTEM NETID takes precedence.

Procedure

Note: If you are installing an MP system, create a record in the SYSTEM NETID file for each processor that you want to be able to IPL.

If you are installing an AP system, create a SYSTEM NETID record only for the master processor.

- Establish write access to the CMS system minidisk:

access 190 b

- Obtain your CPU serial number:

query cpuid

```
CPUID = FFssssssdddd0000
```

ssssss is your CPU serial number; *dddd* is your processor device type.

- Edit the SYSTEM NETID file:

xedit system netid b2

```
*CPUID NODEID NETID
013773 GDLPDP RSCS
```

This is a sample SYSTEM NETID file. Enter your CPU serial number (*ssssss*, obtained above) as the CPUID entry in this file. Replace the sample NODEID and NETID entries with the correct information for your installation.

- When you complete the update, save the file and release the CMS system minidisk from write access:

file

release b

- Save the modified CMS nucleus:

| **ipl 190 clear parm savesys cms**

```
SYSTEM SAVED
```

```
VM/SP Release 6 mm/dd/yy hh:mm:ss
```

Or your own version ID.

■ Go to "Step 22. Install EREP (Optional)" on page 3-131.

Step 22. Install EREP (Optional)

Overview

If you did not order the Environmental Record Editing and Printing program (EREP), go to "Step 23. Do a Full System Backup (Recommended)" on page 3-133.

If you ordered EREP, this licensed program is supplied on a separate distribution tape. To load EREP, you invoke the INSTFPP EXEC installation tool to load the EREP files to the EREP minidisk (MAINT 201 is the default).

Procedure

- Mount the EREP tape.
- Load the EREP code:

instfpp or **instfpp (all**

The default is **all**.

ENTER

```
REWIND COMPLETE
DMKVDD040E Device 001 does not exist
NOW FORMATTING TEMPORARY WORKDISK "001" WITH FILEMODE "C."
FORMATTING OF TEMPORARY WORKDISK "001/C" IS COMPLETE.

nnnn-nnn VM/EREP VER. n REL. n MOD. LEVEL n SL nnn
DO YOU WISH TO INSTALL THIS PRODUCT NOW (YES OR NO)?
```

yes

```
PRT FILE nnnn TO MAINT  COPY nnn NOHOLD

xxxxxxxx MEMO HAS BEEN PRINTED

THE DEFAULT MINIDISK FOR CPEREP INTERFACE FILES ON EREP n.n.n IS:
*______ _______ ______* ( 201 ) *______ _______ ______*

    ENTER   A NULL LINE TO ACCEPT DEFAULT
                         OR
                 NEW MINIDISK CUU
```

ENTER or ***vdev***

Press ENTER to accept the default (**201**) or enter the new minidisk address (***vdev***).

```
LOADING CPEREP INTERFACE FILES FROM DISTRIBUTION TAPE.....
CPEREP MODULE SUCCESSFULLY CREATED
LOADING EREP TXTLIB FILES FROM DISTRIBUTION TAPE.....

xxxxxxxx - VM/EREP n.n.n INSTALLATION COMPLETED SUCCESSFULLY

NOW UPDATING THE PROD LEVEL P FILE...


LEAVING INSTFPP EXEC...
REWIND COMPLETE
```

■ Go to “Step 23. Do a Full System Backup (Recommended)” on page 3-133.

Step 23. Do a Full System Backup (Recommended)

Overview

To protect the system that you have just installed, you should do a full system backup and dump the contents of your DASD onto tape.

Use the UTILITY EXEC to create a stand-alone DASD Dump/Restore (DDR) program on tape, then use that DDR program to backup your system.

Note: This method is preferable to using the DDR on the CMS system minidisk because it ensures that a DDR program is available to restore your system even if the system goes down during the backup. This method is also preferable to using the Starter System DDR, if available, because the Starter System DDR might not be at the same level as your production system.

To obtain maximum protection if you plan to apply preventive or corrective service, you might want to do one backup before applying service and do another backup after applying service.

Procedure

Note: Do this step while logged on to the MAINT virtual machine.

- Mount and ready a utility tape.
- Dump the DDR program onto the utility tape:

utility utiltape ddr

```
DMSWUT975I Moving IPL DDR to tape

DMSWUT976I The IPL DDR program is on tape file number 1

Ready;
```

- Rewind the tape, then IPL the tape drive to load the DDR program:

ipl _vdev_

vdev is the virtual address of the tape drive.

```
VM/370 DASD DUMP/RESTORE PROGRAM - VM/SP
ENTER CARD READER ADDRESS OR CONTROL STATEMENTS
ENTER:
```

- Mount and ready a backup tape.
- Dump the DASD volumes to tape. Follow this sequence for each volume that you want to dump:

sysprint cons

This command directs system output to the console.

```
ENTER:
```

input _vdev_ _devtype_ _label_

vdev is the virtual address of the DASD volume.

devtype is the device type of the DASD.

label is the label of the DASD volume.

```
ENTER:
```

output *vdev* *devtype* (mode *bpi* compact

vdev is the virtual address of the tape drive.

devtype is the device type of the tape drive.

bpi is the tape density (**1600**, **6250**, or **38k**).

compact is an operand that instructs DDR to compress the data on the output tape. For more information about this function, see *VM/SP System Facilities for Programming.*

```
ENTER:
```

dump cpvol

This command dumps onto the tape cylinder 0 (block 0-15 in an FBA device) and all space allocated as PERM, DRCT, and OVRD.

```
DUMPING   label
```

⋮

```
END OF DUMP
```

ENTER

```
END OF JOB
```

■ Go to "Supplementary Procedures" on page 3-136.

Supplementary Procedures

You have now completed the base VM/SP installation. However, you might also want to complete the following supplementary procedures:

- **Verify your installed system.**

  To make sure that CP and CMS are working properly, you can use the Installation Verification Procedure (IVP). See Chapter 5, "Verifying Your Installed VM/SP System" on page 5-1.

- **Install saved segments.**

  For procedures to install the CMSDOS, CMSBAM, CMSVSAM, CMSAMS, CMSINST, CMSVMLIB, and CMSFILES saved segments, see Chapter 6, "Installing Saved Segments" on page 6-1.

- **Install a new system national language.**

  If you received a feature tape for a language that you want to install as the new system national language (replacing American English), see Chapter 7, "Installing a New System National Language" on page 7-1.

- **Install additional national languages.**

  If you received a feature tape for a language that you want to install on your system as an option for selected users, see *VM System Facilities for Programming*.

- **Apply Service.**

  If a PUT (program update tape) was included with your shipment, you might want to apply some or all of the service updates that it contains. For information about VM/SP service, see the *VM/SP Service Guide*.

Chapter 4. Installing VM/SP Using an Existing VM/SP System

The **recommended** procedure to install VM/SP on an existing (prior release) VM/SP (or VM/SP HPO) system is the Starter System procedure described in Chapter 3. If you are migrating from a VM/SP release prior to VM/SP Release 4, you **must** use the Starter System procedure.

The procedure in this chapter is provided for VM/SP Release 4 and VM/SP Release 5 installations that do not have enough DASD available to use the Starter System procedure, or whose system configuration is substantially different from the configuration defined in the IBM-supplied system definition files. However, to use this procedure, installation of specific between-release support is required (see "Migration Considerations" on page 4-2).

Table 4-1 shows the sequence of steps in this procedure.

Table 4-1. Outline of Procedure for Installing VM/SP Using an Existing VM/SP System

Step	**Description**
1	Modify the current CP directory
2	Format the new minidisks
3	Load the VM/SP product tape
4	Build a new CMS nucleus
5	Do optional tasks
6	Tailor the system definition files
7	Build a new CP nucleus
8	Load the system HELP files (optional)
9	Update the SYSTEM NETID file (optional)
10	Load IPCS and complete the CUSTOMER PROFILE
11	Save CMS and HELP
12	Build the system file pool
13	Build the user file pool (recommended)
14	Load, build, and save GCS (optional)
15	Load TSAF (optional)
16	Load AVS (optional)
17	Load the source feature tape (optional)
18	Install EREP (optional)
(sup)	Do a full system backup
(sup)	Verify your installed system
(sup)	Install saved segments
(sup)	Install a new system national language
(sup)	Install additional national languages
(sup)	Receive and apply service

Assumptions

This procedure assumes that:

- You have some VM system programming experience. Therefore, the steps in this procedure are not as detailed as those in the Starter System procedure.
- You are using the MAINT virtual machine to do the installation. All minidisks and SFS directories used for installation and/or system generation are defined for MAINT.
- You understand the page layout and text formats used in this book (see "The Format Used in the Procedures in This Book" on page 1-11).

Migration Considerations

Consider the following, where appropriate:

- If you are migrating from a VM/SP release prior to VM/SP Release 4, you **must** use the Starter System procedure (Chapter 3). If you are migrating from VM/SP Release 4 or VM/SP Release 5, the Starter System procedure is **recommended.**
- **To use the procedure described in this chapter, you must first load and apply the service in APAR VM32788 (CP) and APAR VM32863 (CMS).**
- VM/SP Release 6 includes substantial changes from previous releases. Chapters 1 and 2 of this book highlight the changes that relate to installation and system generation.
- Before building a component, make sure that any minidisk (other than 190) containing previous-release files for that component is either not accessed or accessed lower (closer to Z) than any minidisk or SFS directory used in that step of the procedure.
- VM/SP requires at least 1M real storage.

For more information about VM/SP Release 6 additions, enhancements, and migration considerations, see the *VM/SP Release 6 Guide.*

Planning Considerations

Before you begin this installation procedure, you should carefully plan your installation. Chapter 2, "Planning Your VM/SP Installation," provides important information about:

- Identifying your DASD requirements
- Planning for the system file pool and user file pool
- Planning for the Group Control System (GCS)
- Planning for the Transparent Services Access Facility (TSAF)
- Planning for APPC/VM VTAM Support (AVS)
- Tailoring the system generation files
- Migrating spool files
- Installing national languages on your system.

Preparation

Before you begin Step 1, you should:

- Use the DDR program to create a complete backup tape of your current system. If necessary, you can recover any portion of your current system from the backup tape.

  Note: You can use the COMPACT option on DDR to save time and tape. For more information, see *VM/SP System Facilities for Programming*.

- Have a printout of your current CP directory readily available.

- Consider migrating any spool files that you want to save:

  - You can use the SPTAPE command to save the spool files on tape. For more information about using SPTAPE, see the *VM/SP CP System Command Reference*.

  - If you are migrating from VM/SP Release 4 or later, you can migrate the spool files using the procedure described in Appendix E, "Migrating Spool Files Using SHUTDOWN/WARM IPL" on page E-1.

Notes:

1. This procedure is designed for uniprocessor (UP) installations but can also be used for attached processor (AP) and multiprocessor (MP) installations. Additional requirements for AP and MP installations are noted in the text.

2. The stand-alone programs used in this procedure might not work on processors with the Loop Adapter feature enabled.

3. Certain prompts and responses in this procedure use the term "FB-512" when referring to a fixed-block architecture (FBA) DASD.

Step 1. Modify the Current CP Directory

Overview

The primary tasks in this step are:

- Define a new CMS system minidisk
- Define other required minidisks
- Adjust minidisk sizes (where necessary)
- Provide links to the HELP minidisk (if required)
- Place the updated directory on line.

Procedure

■ Edit your current CP directory and make the following changes:

1. Compare your current minidisk sizes with the requirements for this release and make any necessary adjustments. For more information about the default or recommended minidisk sizes, see Appendix A, "Minidisks and SFS Directories Reserved for MAINT" on page A-1.

2. Define other minidisks as required for the functions, components, and products that you plan to install, such as:

   - Service
   - CMS Shared File System (SFS) file pools
   - National language HELP files
   - Group Control System (GCS)
   - APPC/VM VTAM Support (AVS)
   - Environmental Record Editing and Printing program (EREP).

   For more information, see Appendix A, "Minidisks and SFS Directories Reserved for MAINT" on page A-1.

3. Define MAINT's virtual storage as 16M.

4. Define a minidisk to replace the current CMS system minidisk (MAINT 190). The examples in this procedure use MAINT 490.

   Notes:

   a. Defining the 490 minidisk on the same physical volume as the 190 minidisk simplifies the installation procedure.

   b. Make sure that you define 490 the same size as the 190 minidisk listed in Table A-2 on page A-3.

5. (Current VM/SP Release 1 users only) Make sure that each user has a link to the default system HELP files minidisk, MAINT 19D:

   - Place **LINK MAINT 19D 19D RR** in each user's directory.

     or

   - Have each user add **LINK MAINT 19D 19D RR** to their PROFILE EXEC.

■ File the updated directory and use the DISKMAP EXEC to check for overlaps between minidisk assignments:

diskmap *fn ft*

fn ft is the file ID of your directory. DISKMAP produces a file called *fn* DISKMAP that is written to file mode A. For more information about the DISKMAP EXEC, see the *VM/SP Planning Guide and Reference*.

- Place the directory on line:

direct *fn ft*

fn ft is the file ID of your directory. For more information about the DIRECT command, see the *VM/SP Planning Guide and Reference*.

- To make these changes effective, log off, then log back on as MAINT.

Step 2. Format the New Minidisks

Overview

The primary tasks in this step are:

- Determine the block size of the MAINT 190 minidisk
- Format the new MAINT 490 minidisk
- Format MAINT 490 with the RECOMP option
- Format all other new or resized minidisks
- Move any HELP files from MAINT 19D that you do not want overlaid.

Procedure

■ Determine the block size of the MAINT 190 minidisk:

query disk

■ Format the new MAINT 490 minidisk using the same block size as MAINT 190:

format 490 *fm* (blksize *nnnn*

fm is any available file mode.

nnnn is the block size of the 190 minidisk.

■ Format MAINT 490 with the RECOMP option:

format 490 *fm* *loc* (recomp blksize *nnnn*

fm is the same file mode that you used in the previous command.

loc is the first CKD cylinder or FBA block of the location on 490 where the new CMS nucleus will reside, according to the device type and minidisk size listed in Table 4-2 on page 4-7. This operand makes the cylinders/blocks from 0 to ***loc*** available for use.

nnnn is the same block size that you used in the previous command.

Table 4-2. CMS Nucleus Size and Location

Device Type	**190 Allocation**	**Nucleus Size**	**Location *loc***
3330	173 cyl	18 cyl	155 (cyl)
3340	429 cyl	45 cyl	384 (cyl)
3350	80 cyl	10 cyl	70 (cyl)
3375	120 cyl	12 cyl	108 (cyl)
3380	78 cyl	8 cyl	70 (cyl)
FBA	72000 blk	8256 blk	63744 (blk)

Note: If you try to use the current CMS system minidisk (190) and there is insufficient free space, the RECOMP operand causes an error message to be issued. For this reason, the use of an alternate CMS system minidisk (490) is recommended.

- Format all other new minidisks that you defined in Step 1 plus all minidisks that changed in size (remember to copy your data to a temporary minidisk first):

format *vdev fm*

vdev is the virtual address of the minidisk.

fm is any available file mode.

- If you have HELP files on MAINT 19D that you do not want overlaid, move them to a temporary minidisk.

Step 3. Load the Product Tape

Overview

The primary tasks in this step are:

- Mount the product tape on device 181
- Load the installation tools and profiles to MAINT 191
- Load the system generation tools and files to MAINT 193
- Load the system tools to MAINT 490
- Load the sample files to MAINT 295
- Load the CP delta service code to MAINT 294
- Load the CP apply service code to MAINT 192
- Load the CP object code to MAINT 194
- Load the CMS base code to MAINT 193
- Load the Procedures Language/VM delta service code to MAINT 293
- Load the Procedures Language/VM apply service code to MAINT 392
- Load the Procedures Language/VM object code to MAINT 193
- Load the CMS delta service code to MAINT 293
- Load the CMS apply service code to MAINT 392
- Load the CMS system code to MAINT 490

Procedure

Note: See page 1-6 for the layout of the tape files on the VM/SP Product Tape.

■ Mount the first product tape volume on tape device 181.

■ Load the installation tools and profiles to MAINT 191 (make sure that 191 is accessed as file mode A):

```
vmfplc2 fsf
```

This command moves the product tape forward one tape mark to bypass the header file.

```
vmfplc2 load * * a
```

This command loads the next tape file, INSTALL TOOLS, which includes:

ITASK EXEC
SPLOAD EXEC
SPLOAD PROFILE
PROFILE EXEC (for MAINT).

■ Load the system generation tools and files to MAINT 193:

```
access 193 t
```

You must access 193 with a file mode letter below S (closer to Z). This prevents a revised CMS command from being used before the CMS nucleus is prepared to issue new or revised messages.

vmfplc2 load * * t

This command loads the next tape file, SYSGEN TOOLS, which includes:

5664167E $PPF
VMFBLD EXEC
UTILITY EXEC
SNTMAP EXEC
DISKMAP EXEC
DIRECGEN EXEC
PASSMOD EXEC
INSTFPP EXEC
$MSG4I$ EXEC
$DASD$ CONSTS.

- ■ Load the system tools to MAINT 490:

access 490 u

You must access 490 with a file mode letter below S (closer to Z). This prevents a revised CMS command from being used before the CMS nucleus is prepared to issue new or revised messages.

vmfplc2 load * * u

This command loads the next tape file, SYSTEM TOOLS, which includes:

VMFASM EXEC
VMFOVER EXEC
VMFSETUP EXEC.

- ■ Load the sample files to MAINT 295:

access 295 b

vmfplc2 load * * b

This command loads the next tape file, SYSTEM SAMPLES, which includes:

DMSNGP SAMPLE
DMKBOX ASSEMBLE
DMKFCB ASSEMBLE
RPWLIST DATA
VMSYS POOLDEF
VMSERVS DMSPARMS
VMSYSU POOLDEF
VMSERVU DMSPARMS
SERVER PROFILE
DIRECGEN EXTENTS.

vmfplc2 load * *devtype[-mmm]* b

devtype[-mmm] is the device type and model (if any) of your system residence DASD:

3350
3370
3370-2
3375
3380
3380-E4
3380-K4
9313
9332
9332-600
9335.

This command loads the following files from the DASDMODL SAMPLES tape file:

DIRECT *devtype[-mmm]*
DMKSNT *devtype[-mmm]*.

vmfplc2 load * *devtype* b

devtype is the device type of your system residence DASD. This command loads the following files from the DASDTYPE SAMPLES tape file:

DMKRIO *devtype*
DMKSYS *devtype*.

- Load the CP delta service code to MAINT 294:

access 294 d

vmfplc2 load * * d

This command loads the next tape file, CP DELTA, which contains CP update files and serviced text decks.

- Load the CP apply service code to MAINT 192:

access 192 e

vmfplc2 load * * e

This command loads the next tape file, CP APPLY, which contains CP auxiliary control files.

- Load the CP object code to MAINT 194:

access 194 f

vmfplc2 load * * f

This commmand loads the next tape file, CP OBJECT, which includes the CP object code and the following printer universal character set and font offset buffer sample files:

DMKPIA ASSEMBLE
DMKPIB ASSEMBLE
DMKUCB ASSEMBLE
DMKUCC ASSEMBLE
DMKUCS ASSEMBLE.

■ Load the CMS base code to MAINT 193:

vmfplc2 load * * t

This command loads the next tape file, CMS BASE, which contains text files and other items used primarily by system programmers. This file also contains the SFS sample files. For more information about these SFS sample files, see the *VM/SP CMS User's Guide*.

■ Load the Procedures Language/VM delta service code to MAINT 293:

access 293 v

You must access 293 with a file mode letter below S (closer to Z). This prevents a revised CMS command from being used before the CMS nucleus is prepared to issue new or revised messages.

vmfplc2 load * * v

This command loads the next tape file, REXX DELTA, which contains Procedures Language/VM update files and serviced text decks.

■ Load the Procedures Language/VM apply service code to MAINT 392:

access 392 w

You must access 392 with a file mode letter below S (closer to Z). This prevents a revised CMS command from being used before the CMS nucleus is prepared to issue new or revised messages.

vmfplc2 load * * w

This command loads the next tape file, REXX APPLY, which contains Procedures Language/VM auxiliary control files and update shells.

- Load the Procedures Language/VM object code to MAINT 193:

```
vmfplc2 load * * t
```

This command loads the next tape file, REXX OBJECT, which contains Procedures Language/VM text files.

- Load the CMS delta service code to MAINT 293:

```
vmfplc2 load * * v
```

This command loads the next tape file, CMS DELTA, which contains CMS update files and serviced text decks.

- Load the CMS apply service code to MAINT 392:

```
vmfplc2 load * * w
```

This command loads the next tape file, CMS APPLY, which contains CMS auxiliary control files and update shells.

- Load the CMS system code to MAINT 490:

```
vmfplc2 load * * u
```

This command loads the next tape file, CMS SYSTEM, which contains execs and modules available to the general CMS user. This tape file also contains the IBM-supplied callable services library (VMLIB).

Tailoring the SPLOAD PROFILE

Other tape files (containing HELP, IPCS, GCS, TSAF, AVS, and source code) are loaded later in this procedure using the ITASK EXEC and SPLOAD EXEC installation tools, which refer to the SPLOAD PROFILE for load parameters. For information about the SPLOAD PROFILE syntax, see "SPLOAD EXEC" on page 8-19.

Warning: Do not attempt to tailor this file unless you have a thorough working knowledge of the system and are aware of the implications of your changes. For example, if you change the SPLOAD PROFILE, you might not be able to use the ITASK EXEC installation tool, because ITASK expects to find certain information at specific locations. There are no safeguards in the code to check the validity of your changes.

Step 4. Build a New CMS Nucleus

Overview

The primary tasks in this step are:

- Create a product parameter override file for CMS
- Apply service to CMS (optional)
- Modify, rename, and assemble the sample DMSNGP file
- Build and IPL the new CMS nucleus
- Save and print the CMS load map (optional)
- Rebuild the system assembler.

Procedure

> **WARNING**
>
> Remove all previous-release CMS service from the MAINT 293 and and 295 minidisks before you build the new CMS nucleus.

Note: Due to expansion of the CMS nucleus, you might need to increase the size of your virtual machine.

■ Create a product parameter override file. You can use any file name for the override file (except the product ID, which is reserved for the base product parameter file), but the file type must be $PPF. In the override file, create a CMS parameter override area and define MAINT 490 as the minidisk on which the CMS nucleus is built (BUILD1). Figure 4-1 shows an example of the complete override file. This example uses CMS490 as the name of the override area.

```
:OVERLSTP. CMS490
:OVERLST.
:CMS490. CMS 5664167E
:MDA. UPDATE
BUILD1 490
:END.
```

Figure 4-1. Example of an Override File Containing a CMS Override Area Defining a New BUILD1 Minidisk

When you enter a command that requires you to specify the name of the product parameter file and the name of the CMS component area, enter the names of this override file and override area instead. The CMS 5664167E parameters in the override file are pointers to the base CMS area in the base product parameter file. The BUILD1 490 parameter in the override file updates (overrides) the BUILD1 190 parameter defined in the CMS area in the base file.

For more information about how an override file works, see the *VM/SP Service Guide*.

■ The recommended procedure for applying service to CMS is to complete the VM/SP installation first, then apply the service. However, if you need to apply preventive service, corrective service, or local service to CMS now, use the procedures in the *VM/SP Service Guide*. **Do not** receive or apply service for any other component. After you apply the CMS service, continue with the rest of this installation procedure.

Note: If you apply service to CMS, you might need to rebuild serviced modules before you continue with the installation procedure.

- Edit the CMS nucleus generation profile (the sample file is named DMSNGP SAMPLE).

DMSNGP defines responses to the system prompts that are generated when you build a CMS nucleus. If a response is defined in the DEFNUC macro call in DMSNGP, the system uses that value, and the corresponding prompt does not appear when you build the nucleus. However, to ensure that all necessary system values are defined, if a DMSNGP statement is missing or contains no value (and that parameter has no default value in the DEFNUC macro), or if a DMSNGP statement contains a question mark (?), the corresponding prompt appears when you build the nucleus.

DMSNGP is supplied with most responses predefined. For more information about DMSNGP and the DEFNUC macro, see the *VM/SP Planning Guide and Reference*.

Make the following changes to the file:

Note: On each line, make sure that you have at least one blank inserted between the macro statement and the comment section.

1. Change the SAVESYS statement to **NO**.

   This statement indicates that the CMS system should **not** be saved after the build. You do not want to save CMS at this point in the procedure because you have not yet modified the CMS entry in DMKSNT.

   Note: You could also change this statement to **?** to generate a prompt during the build. If so, you must enter **0** (NO) in response at that time.

2. Change the IPLADDR statement to **490**.

   You want to write the new CMS nucleus to the MAINT 490 minidisk that you defined and formatted in previous steps.

   Note: You could also change this statement to **?** to generate a prompt during the build. If so, you must enter **490** in response at that time.

3. Complete the CYLADDR statement to indicate the cylinder/block address on 490 where the nucleus is written, according to your DASD type:

Device	Address
3330	155 (cyl)
3340	384 (cyl)
3350	70 (cyl)
3375	108 (cyl)
3380	70 (cyl)
FBA	63744 (blk)

4. If you want to build an uppercase English system, change the LANGID statement from AMENG to **UCENG**.

5. Change other statements, such as the version identification and the installation heading, to reflect the responses that you want to use or the prompts that you want to receive. You must place single quotes around any character string that you enter as the new VERSION or INSTID.

- Save your changes, then rename the file (if you are using the sample) and assemble it:

```
rename dmsngp sample b = assemble =

release u

access 490 c/a dmssp *
```

To assemble DMSNGP, you must have access to the CMS control file (DMSSP CNTRL) and the DEFNUC macro (located in DMSSP MACLIB).

```
vmfasm dmsngp dmssp (ctl

:
```

```
DMSNGP TEXT A1 CREATED
```

- Copy the TEXT file to the new MAINT 490 minidisk:

```
access 490 u

copyfile dmsngp text a = = u (replace olddate
```

- Build the new CMS nucleus:

```
access 490 u/u

access 490 c/a dmssp *

spool prt *

vmfbld ppfname cms490 cmsload (nosetup
```

ppfname is the name of the override file that you created earlier in this step.

cms490 is the name of the CMS parameter override area in the override file.

```
SYSTEM LOAD DECK COMPLETE

PUN FILE spoolid TO MAINT COPY 0001 NOHOLD
```

Nucleus Generation Prompts and Responses

Note: Depending on the status of your DMSNGP file, you might receive none, some, or all of the prompts shown in this section (which ends on page 4-19). A particular prompt appears **only** if the corresponding statement in DMSNGP is missing or contains no value (and the DEFNUC macro contains no default value), or if the DMSNGP statement contains a question mark (?). If you do not receive a prompt, the system is using either the value specified in DMSNGP or the default defined in DEFNUC.

```
DMSINQ606R System disk address =
```

ENTER or ***vdev***

190 is the default.

```
DMSINQ615R Y-disk address =
```

ENTER or ***vdev***

19E is the default.

```
DMSINQ640R HELP disk address =
```

ENTER or ***vdev***

19D is the default.

```
DMSINQ764R Language id =
```

ENTER or **uceng**

This response identifies the language ID (*langid*) of your system national language. The default is mixed-case American English (**AMENG**). If you want to build an uppercase English system, enter **uceng**.

```
DMSINQ293R Is this a DBCS language?  Enter 1 (YES) or 0 (NO).
```

ENTER (or **0**) or **1**

The default is **0** (NO); American English and uppercase English are not DBCS (Double-Byte Character Set) languages.

```
DMSINQ295R Language level id =
```

ENTER or ***levelid***

If you load multiple versions of a national language into saved segments, ***levelid*** is a single alphanumeric character (A-Z, 0-9) that must match the level ID operand of the LANGGEN command used to build the language saved segment to be used with this CMS nucleus. It must also match the third character of the language saved segment name defined in the system name table (DMKSNT). The system national language does not use a saved segment. S is the default.

For more information about the LANGGEN command and the language level ID, see *VM System Facilities for Programming*.

```
DMSINQ1104R File system read/write cache buffer size =
```

ENTER or ***nn***

This is the amount of read-ahead/write-behind buffering to be used for file input and file output operations. You can enter an amount from 1 to 28 (K). **12K** is the default.

```
DMSINQ296R Should the installation segment be used? Enter 1 (YES) or 0 (NO).
```

The CMS installation saved segment is an optional saved segment into which you can place frequently-used EXEC files and System Product Editor (XEDIT) macros. When a user IPLs CMS, the CMS installation saved segment is connected to the user's virtual machine. Multiple users can share the same executing copy of an EXEC file or XEDIT macro stored in this saved segment.

You can install the CMS installation saved segment either as a physical saved segment or as a logical saved segment. If you install it as a logical saved segment, you can also include other objects, such as modules, text files, callable services libraries (CSLs), language repositories, and user-defined objects.

You install the CMS installation saved segment **after** you install your base system, but you must indicate **now** if you want to use it. For more information about installing this saved segment, see Chapter 6, "Installing Saved Segments" on page 6-1.

ENTER (or **1**) or **0**

The default is **1** (YES). Enter **0** if you do not want to use this saved segment.

```
DMSINQ310R Installation segment name =
```

This prompt appears only if you accepted the default (or entered **1**) at the previous prompt.

ENTER or ***segname***

Enter a one-to-eight alphanumeric character name for the CMS installation saved segment, or press ENTER to accept the default name, **CMSINST**.

```
DMSINI729R Do you want to save the system?  Enter 1 (YES) or 0 (NO).
```

0

You **do not** want to save CMS now, because you have not yet modified the CMS entry in the system name table (DMKSNT).

```
DMSINI607R Rewrite the nucleus?  Enter 1 (YES) or 0 (NO).
```

1

Enter **1** to write the CMS nucleus on the minidisk that you specify in your response to the next prompt.

```
DMSINI608R IPL device address =
```

490

490 is the new CMS system minidisk that you defined and formatted in previous steps.

```
DMSINI609R Nucleus (CYL or BLK) address =
```

nnnnn

nnnnn is the location on the 490 minidisk where the new CMS nucleus is written. Enter the correct cylinder/block address for your DASD type:

Device	Address
3330	155 (cyl)
3340	384 (cyl)
3350	70 (cyl)
3375	108 (cyl)
3380	70 (cyl)
FBA	63744 (blk)

```
DMSINI610R Also IPL CYL/BLK 0?  Enter 1 (YES) or 0 (NO).
```

1

Enter **1** to write the initial IPL text on cylinder/block 0 of the minidisk where the CMS nucleus is written.

The initial IPL text is a bootstrap program that reads the CMS nucleus from the cylinder/block where the nucleus is written (as defined in your response to prompt DMSINI609R). The initial IPL text is always written on the same cylinder/block as the nucleus. If the initial IPL text is not also written on cylinder/block 0, you must specify the cylinder/block address of the nucleus when you issue IPL commands for this system. For more information, see the description of the IPL command in the *VM/SP CP General User Command Reference*.

```
DMSINI611R Enter version identification:
```

The version identification is displayed each time you IPL the CMS system you are now building.

ENTER or ***version***

You can enter up to 32 descriptive characters to identify this version and level of CMS, or you can press ENTER to accept the default version ID:

VM/SP Release 6 *mm/dd/yy hh:mm:ss*

```
DMSINI612R Enter installation heading:
```

The installation heading appears at the beginning of each output file created using this CMS nucleus.

ENTER or ***heading***

You can enter up to 64 descriptive characters to serve as an installation heading, or you can press ENTER to accept the default heading:

```
VM/SP CONVERSATIONAL MONITOR SYSTEM
```

End of Nucleus Generation Prompts and Responses

```
DMSWSP327I The installation saved segment could not be loaded
```

The CMS installation saved segment (CMSINST) cannot be loaded into your virtual machine now because this saved segment is not installed until after you complete the base VM/SP installation.

This message does not appear if you indicated (in the DMSNGP file or in response to the DMSINQ296R prompt) that you do not want to use this saved segment.

```
VM/SP Release 6 mm/dd/yy hh:mm:ss
```

Or your own version ID.

ENTER

- IPL the new CMS nucleus:

define 190 590

This command redefines the old CMS system minidisk to 590.

define 490 190

This command redefines the new CMS system minidisk to 190.

ipl 190 clear

You must IPL the new CMS system minidisk to use the new system generation tools.

```
PRT FILE spoolid TO MAINT COPY 0001 NOHOLD
```

The spool ID in this message identifies the CMS load map file. Record the spool ID if you plan to save or print the load map.

```
VM/SP Release 6 mm/dd/yy hh:mm:ss
```

Or your own version ID.

access 190 a

access 193 b

The system generation tools are on this minidisk.

Saving and Printing the CMS Load Map

Note: If you are migrating from VM/SP Release 2 or 3, you cannot receive the CMS load map until after you build the new CP nucleus (Step 7).

The CMS load map is in MAINT's virtual reader. To save the load map on the CMS base minidisk (MAINT 193), enter:

#cp spool prt nohold
receive *spoolid* cmsnuc map b

After you save the load map, you can print a copy of it by entering:

print cmsnuc map b

- Rebuild the system assembler using modules for the new release:

  Note: If you applied service to CMS in this step and serviced any of the files used to build the ASSEMBLE module (DMSASM, DMSASD, or DMSASN), you must invoke the CMSGEND EXEC to rebuild the ASSEMBLE module before you invoke ASMGEND:

  cmsgend assemble

asmgend

```
ASSEMBLE XF GEND PROC

ENTER TAFGET DISK MODE FOR ASSEMBLE MODULES
DEFAULTS TO S-DISK IF NONE ENTERED.
```

ENTER

```
ASSEMBLE XF GEND COMPLETE
```

ipl 190 clear

This command reaccesses the CMS system minidisk with the new assembler module that you just created. It also reaccesses MAINT's default A file mode (usually the 191 minidisk).

Step 5. Do Optional Tasks

Overview

Primary tasks:

- Load the CP, CMS, and IPCS source code (optional)
- Print copies of the sample files (optional)
- Create a stand-alone service utility tape (optional).

Procedure

■ Load the CP, CMS, and IPCS source code (optional).

Source code for the CP and IPCS components and source code for some CMS modules is provided on the product tape. Invoke the SPLOAD EXEC to load the source tape files to the destinations specified in the SPLOAD PROFILE:

Tape File	**Destination**
CP SOURCE	MAINT 394
CMS SOURCE	MAINT 393
IPCS SOURCE	MAINT 393

Note: Source code is supplied and loaded down to minidisk in **packed** format. To unpack your source files after they are loaded, use the COPYFILE command with the UNPACK option. For more information about this option, see the *VM/SP CMS Command Reference*.

The recommended MAINT 394 and 393 minidisk sizes (see Table A-2 on page A-3) allow enough space for all of the packed files plus some (but not all) unpacked modules. If you plan to unpack all of your files, you should redefine these minidisks 80% larger.

spload ***group*** **source**

To use this command, the tape drive must be attached to MAINT at 181.

group refers to the GROUP portion of the tape file identifier (GROUP ELEMENT) in the SPLOAD PROFILE: **cp**, **cms**, or **ipcs**.

Enter this command for each source tape file that you want to load.

Notes:

1. If you are loading a minidisk that has not been formatted yet, the SPLOAD EXEC formats the minidisk before continuing with the load.
2. When SPLOAD formats a minidisk, you might receive two device error messages. You can ignore these messages.

```
DMSWSL409I Loading group SOURCE to MAINT vdev
```

- Print copies of the sample files (optional).

  Having printed copies of the sample files on hand could be helpful when you tailor the system definition files in Step 6. Use your normal method for printing files from minidisk to obtain copies of the files that you want. Remember that some sample files have DASD-specific file types (see Step 3).

- Create a stand-alone service utility tape (optional).

  Use the UTILITY EXEC to create a service utility tape containing one or more of the following stand-alone programs:

  - Device Support Facility
  - DASD Dump/Restore (DDR)
  - Format/Allocate (FMT)
  - CP Directory (DIR).

  For more information, see the description of the UTILITY EXEC in the *VM/SP Service Guide*.

Step 6. Tailor the System Definition Files

Overview

Primary tasks:

- Tailor the VM/SP product parameter file
- Tailor the current CP directory
- Tailor the current DMKSYS file
- Tailor the current DMKRIO file
- Tailor the **sample** DMKSNT file
- Tailor the other system definition files (as required).

Procedures

Use the System Product Editor (XEDIT) to tailor these files as required for your installation. Where necessary, you are directed to other books in the VM/SP library for additional information.

Product Parameter File Considerations

■ Tailor the VM/SP product parameter file (5664167E $PPF) if you want to change the supplied system generation parameters. There is a separate area in the product parameter file for each VM/SP component. A component area begins with a component tag (such as :CP.). The *VM/SP Service Guide* describes the function of the tags and records in the product parameter file and also provides a complete listing of the product parameter file shipped with VM/SP.

You can make your changes directly to the statements in the component areas in the product parameter file or define your changes in special override areas. However, the recommended procedure for making most changes is to put the changes in override areas in a separate file called a product parameter override file. For information about creating override areas and override files and how they work, see the *VM/SP Service Guide.*

Examine the following parameters:

– **Minidisk and SFS Directory Assignments**

In each component area, there is a section that begins with a minidisk/directory assignments tag (:MDA.) that defines the minidisks and/or SFS directories assigned to the functional groups (BUILD, DELTA, APPLY, and so on) for that component. Make sure that these assignments reflect your actual system configuration, and that essential items are not commented out (with * in the first column).

– **Building an AP or MP System**

The base product parameter file contains parameters for building a uniprocessor (UP) system, but includes parameters for building an attached processor (AP) or multiprocessor (MP) system in CP override areas called CPAP and CPMP, respectively. In Step 7, when you issue the VMFBLD command to build the CP nucleus, to build an AP or MP system, enter the name of the corresponding override area (CPAP or CPMP) instead of CP.

– **National Language (American English or Uppercase English)**

VM/SP is shipped with mixed-case American English (AMENG) as the default national language. However, files for uppercase English (UCENG) are also provided on the product tape. When you build your VM/SP system, American English is installed by default as the system national language, which is the language used for most system functions. If you want to build an uppercase English system, locate the national language tag (:NLS.) in the CP, CMS, and GCS component areas of the product parameter file and change the language ID from AMENG to **UCENG** (the parameter must be in uppercase).

Note: Some system functions are always in American English regardless of the language installed as the system national language. For example, the language of the installation and system generation tools is American English and cannot be changed.

- **CP Options**

  There are three options that you can select when building the CP nucleus:

  - Small CP
  - CP FRET Trap
  - Virtual = Real.

  These options are selected by the CP control file and/or CP build list specified in the product parameter file. In the IBM-supplied product parameter file, these options are **not** selected. For information about these options and how to select them, see Appendix D, "Special Options for CP" on page D-1.

CP Directory Considerations

■ Compare your current CP directory to the sample directory (DIRECT *devtype[-mmm]*). Using the sample file as a guide, update your current file with any items that you want to include in the new system.

Also consider the following changes:

- **Defining Non-Restricted User Logon Passwords**

  As a system security measure, the new CP includes the Auto-Deactivation of Restricted Passwords (ADRP) feature. After you build the new CP (or if your current CP already includes ADRP), entering the DIRECT command to process a CP directory file causes the system to search the directory for logon passwords that match the list of restricted logon passwords contained in the RPWLIST DATA file that you loaded to MAINT 295 in Step 3. All passwords that match are changed to NOLOG in the directory before the directory is placed on line. You cannot log on to any virtual machine that has a NOLOG password.

  Change all restricted logon passwords in your CP directory to non-restricted passwords. For a list of the restricted passwords, see Appendix B, "Restricted Logon Passwords" on page B-1.

- **Updating the MAINT Virtual Machine**

  Update the MAINT virtual machine to make permanent the temporary changes that you defined in Step 4:

  1. Change MAINT's 190 MDISK to 590.
  2. Change MAINT's 490 MDISK to 190.

- **Adding User IDs**

  Some of the virtual machines defined in the sample directory (such as GCS, TSAFVM, VMSERVS, VMSERVU, AVSVM) might not be included in your current system. Move these entries to your current CP directory.

  If you plan to install additional program products (such as RSCS) that require system user IDs not in the current directory, create directory entries for them now using the format described in the *VM/SP Planning Guide and Reference.*

- **Modifying the System File Pool Server Machine**

  You might want to change the supplied specification of the system file pool server virtual machine (VMSERVS). For example, you might want to increase the minidisk allocation for the system file pool.

  Note: Changing the definition of this machine might also require changes to the VMSYS POOLDEF file. You should not attempt to change any of these files unless you have a thorough knowledge of how SFS works.

  For more information about the configuration of a file pool server machine, see *VM/SP CMS Shared File System Administration.*

- **Modifying the User File Pool Server Machine**

  You might want to change the supplied specification of the user file pool server virtual machine (VMSERVU). For example, you might want to increase the minidisk allocation for the user file pool.

  Note: Changing the definition of this machine might also require changes to the VMSYSU POOLDEF file. You should not attempt to change any of these files unless you have a thorough knowledge of how SFS works.

  For more information about the configuration of a file pool server machine, see *VM/SP CMS Shared File System Administration.*

- **Modifying the GCS Recovery Machine**

  If you are installing the GCS component, you might want to change the supplied specification of the GCS recovery machine (GCS). For more information about GCS, see the *VM/SP Planning Guide and Reference.*

- **Modifying the TSAF Machine**

  If you are installing the TSAF component, you might want to change the supplied specification of the TSAF virtual machine (TSAFVM). Also remember that all user IDs in all systems in the TSAF collection must be unique. For more information about TSAF, see *VM/SP Connectivity Planning, Administration, and Operation.*

- **Modifying the AVS Machine**

  If you are installing the AVS component, you might want to change the supplied specification of the AVS virtual machine (AVSVM). For more information about AVS, see *VM/SP Connectivity Planning, Administration, and Operation.*

- **Restructuring User Classes**

  To restructure user classes, you must make the appropriate changes to the directory and provide OVRD formatted space for the OVERRIDE file. OVRD space must be allocated on the same volume as the directory.

If you plan to restructure your user classes, you should do it **after** the system is installed. For additional information about restructuring user classes, see *VM/SP Administration.*

- **Changing Minidisk Read/Write Passwords**

  If you plan to install optional feature program products on the VM/SP base, the feature program product installation tool (INSTFPP EXEC) assumes that certain minidisks have specific passwords. Do not delete or alter the passwords of minidisks related to these products until after they are installed.

■ If you add minidisks to the directory or change minidisk allocations, use the DISKMAP EXEC to check for gaps and overlaps between minidisk assignments:

diskmap _fn ft_

DISKMAP produces a file called _fn_ DISKMAP that is written to file mode A. For more information about the DISKMAP EXEC, see the *VM/SP Planning Guide and Reference.*

> WARNING
>
> **Do not** use the DIRECT command to place the updated CP directory on line at this time. Because you are using your current CP to build the new CP, you must continue to use the version of the directory that is associated with your current CP until the new CP is built.

DMKSYS Considerations

■ Compare your current DMKSYS file to the sample file (DMKSYS *devtype*). Using the sample file as a guide, update your current file with any items that you want to include in the new system.

Also consider the following changes:

- **Allocating Space for Paging and for Dumps**

  The sample DMKSYS file does not provide space for paging or for dumps. If you plan to use the sample file to build your system, TEMP space is used for paging and for dumps unless you unless you allocate space for these functions. For information about calculating space requirements and allocating space in the SYSOWN macro, see the *VM/SP Planning Guide and Reference.*

- **Restructuring User Classes**

  If you plan to restructure your current user classes, you might need to change the user class definitions in the SYSFCN macro. See the *VM/SP Planning Guide and Reference* for more information about this macro.

- **Defining the Real VMSRES Address**

  If you plan to save an IPLable CP nucleus on tape, you must modify the SYSRES=123 statement in the SYSRES macro to include the real address of the VMSRES volume as the *altaddr*. See the *VM/SP Planning Guide and Reference* for the proper format for this entry.

- **Building an AP or MP System**

  If you plan to generate an AP or MP system, you must change the SYSCOR macro. For more information about this macro, see the *VM/SP Planning Guide and Reference.*

DMKRIO Considerations

- Update your current DMKRIO file to reflect any changes that you made to your I/O device configuration for this installation.

  At second level, make sure that the indicated console address matches the device that you are using.

DMKSNT Considerations

- Due to the number of changes made to the DMKSNT file from previous releases, you should **use the IBM-supplied DMKSNT to build the new system.** Compare your current DMKSNT file to the sample file (DMKSNT *devtype[-mmm]*). Update the sample file with any items from your current file that you want to include in the new system.

  If you make any changes to this file, you must group like macros together in the DMKSNT CSECT. That is, group all NAMESYS macros together, group all NAMENCP macros together, group all NAME3800 macros together, and group all NAMELANG macros together. If you group them incorrectly (for example, by placing a NAMENCP macro between two NAMESYS macros), the system generates an MNOTE indicating that the macros are out of sequence.

  > **WARNING**
  >
  > Make sure that your changes do not cause overlay of other products on your DASD. Also make sure that no physical saved segment or saved system overlays another physical saved segment or saved system in virtual storage unless they are **never** used at the same time. Table 6-1 on page 6-3 shows the layout of physical saved segments and saved systems defined by the sample DMKSNT.

  Consider the following changes:

  - **Using Mixed DASD**

    The sample DMKSNT file assumes that all system DASD volumes are the same device type as the VMSRES volume. If you are using a different device type for any volume, some adjustment in the location of named saved systems and/or physical saved segments on the volume might be necessary. For reference, you might want to look at the sample DMKSNT file for the other DASD type. To load the file from the product tape, enter:

    spload dasdmodl samples dmksnt *devtype[-mmm]*

    where ***devtype[-mmm]*** is the device type (and model designation, if any) of the other DASD. After you load the file, you can browse it (using XEDIT) and/or print it.

  - **Modifying the CMS System Minidisk Pointers**

    Change the SYSCYL (or SYSBLOK) parameter in the CMS NAMESYS macro to define the starting location of the new MAINT 190 minidisk (the MAINT 490 minidisk that you defined in Step 1).

    If the new MAINT 190 minidisk resides on a different physical volume from the original 190 minidisk (now 590), also change the VSYSRES parameter (VSYSRES must point to the new volume).

  - **Adding New Entries**

    If you plan to install national languages or program products that require physical saved segments, you should add the DMKSNT entries now.

Otherwise, you will have to reassemble DMKSNT and rebuild the CP nucleus.

- **Modifying the GCS Entry**

  The sample DMKSNT includes an entry that defines a GCS saved system. If you are installing the GCS component, in your pre-installation planning (Chapter 2) you should have recorded on your Installation Reference Worksheet any changes to be made to this sample GCS entry (such as the load address, the name of the saved system, etc.). Make those changes now to the file.

  If you plan to install more than one GCS nucleus, you must create entries for the additional systems.

■ To verify your changes, use the SNTMAP EXEC, an SNT mapping tool that resides on the MAINT 193 minidisk:

sntmap dmksnt *devtype[-mmm]*

SNTMAP produces two output files on file mode A:

- DASD SNTMAP lists the saved system and physical saved segment names and DASD allocations in order of occurrence on each DASD volume.
- MEMORY SNTMAP lists the saved system and physical saved segment names and DASD allocations according to location in virtual storage.

For more information about the SNTMAP EXEC, see the *VM/SP Planning Guide and Reference.*

■ Rename or erase the current DMKSNT file, then rename the sample file to become the current file.

File Pool Considerations

The generation of a file pool requires data from two file pool definition files:

- The POOLDEF file contains information about the location of the minidisks in the file pool and their function;
- The DMSPARMS file contains start-up parameters for the file pool server machine.

Two sets of files are supplied on the product tape:

- VMSYS POOLDEF and VMSERVS DMSPARMS define a system file pool that is used for loading GCS, TSAF, and AVS code.
- VMSYSU POOLDEF and VMSERVU DMSPARMS define a user file pool that you can build for user files.

Although these files can be tailored, you should not make any changes unless you have a thorough working knowledge of SFS. For more information, see *VM/SP CMS Shared File System Administration.*

Other Considerations

■ If you want to change the forms control buffer macros, tailor DMKFCB ASSEMBLE. For more information, see *VM/SP Administration.*

■ If you want to change the design or contents of system logos, such the system logon screen logo or the printer separator page logo, tailor DMKBOX ASSEMBLE.

- If you have defined new CP commands, see *VM/SP Administration* for information about adding the new commands to CP.
- If you want to modify the universal character set or font offset buffer for the printer(s) in your system, tailor the corresponding module(s):

Printer Type	Module
1403	DMKUCS
3211	DMKUCB
3203	DMKUCC
3289	DMKPIA
3262	DMKPIB

Refer to the module prologues for specific information.

Note: These files were loaded to the MAINT 194 minidisk.

Step 7. Build a New CP Nucleus

Overview

The primary tasks in this step are:

- Back up the current CP nucleus and $DIRECT$ area
- Rename or erase any duplicate system definition files
- Apply service to CP (optional)
- Assemble the tailored system definition files
- Build the new CP nucleus
- Save and print the CP load map (optional)
- Place the tailored CP directory on line
- Shut down the production system and IPL the new VMSRES volume
- Save an IPLable CP nucleus on tape (optional)
- Define and save an additional CP nucleus (optional).

Procedure

■ Use the DDR program to backup your current CP nucleus and $DIRECT$ area. For more information about using DDR, see *VM/SP System Facilities for Programming.*

■ To avoid having duplicate files, rename or erase any previous-release system definition files that you are not using to generate the new system. **To obtain complete function at the new release level, you must use the sample DMKBOX, DMKFCB, and DMKSNT.**

■ The recommended procedure for applying service to CP is to complete the VM/SP installation first, then apply the service. However, if you need to apply preventive service, corrective service, or local service to CP now, use the procedures described in the *VM/SP Service Guide* to receive and/or apply the CP service. **Do not** receive or apply service for any other component. **Do not** use the procedure described in that book to build CP. After you apply the CP service, return to this step and continue with the rest of this installation procedure.

Note: If you apply service to CP, you might need to rebuild serviced modules before you continue with the installation procedure.

■ Set up the assemble access order:

vmfsetup *ppfname* *compname* (access asm	***ppfname*** is the file name of the product parameter file that contains the CP parameters you want to use: • If you using only the base product parameter file, enter **5664167e**. • If you want to use alternative and/or additional parameters defined in a product parameter override file, enter the name of the override file.

compname is the name of the component area in the product parameter file (base file or override file) that contains the CP parameters you want to use:

- If you are using only the base CP area in the base file (to build a uniprocessor system), enter **cp**.
- If you want to use alternative and/or additional CP parameters defined in an override area, either in the base file or in an override file, enter the name of the override area. For example, to use the control file and loadlist defined in the CPAP override area or CPMP override area in the base file (to build an attached processor system or multiprocessor system), enter **cpap** or **cpmp**.

■ Assemble the system definition files that you tailored in Step 6. Assemble the sample DMKBOX, DMKFCB, and DMKSNT even if you did not tailor these files.

vmfasm *fn* *ppfname* *compname* (nosetup

fn is the name of the file to be assembled.

ppfname is the file name of the product parameter file (base file or override file) that contains the CP parameters you want to use.

compname is the name of the component area in the product parameter file (base file or override file) that contains the CP parameters you want to use.

```
DMSUPD181E No update files were found
DMSWAM1907I Assembling fn
```

Note: If any errors occur during the assembly:

1. Examine the flagged statement(s).
2. Edit the file and correct the error(s).
3. Reenter the VMFASM command to assemble the file.

```
ASSEMBLER (XF) DONE
NO STATEMENTS FLAGGED IN THIS ASSEMBLY
DMSWAM1909I fn TEXT A created
```

- Build the new CP nucleus:

vmfbld ***ppfname*** ***compname*** ***bldlist***

ppfname is the file name of the product parameter file (base file or override file) that contains the CP parameters you want to use.

compname is the name of the component area in the product parameter file (base file or override file) that contains the CP parameters you want to use.

bldlist is the file name of the loadlist you want to use to build this nucleus, **if that loadlist is different from the one specified in the product parameter file (base file or override file) you are using**. If you do not want to change the loadlist, you can omit this operand.

```
⋮

NUCLEUS LOADED ON VMSRES --- STARTING CYL/BLK=nnnn  , LAST CYL/BLK USED=nnnn

CP entered; disabled wait PSW '00020000 00000012'
```

Notes:

1. On many processors the PSW message does not display automatically. If the PSW message does not display within about one minute after the NUCLEUS LOADED message, enter:

   #cp display psw

2. If an error occurs during the load:

   a. Inspect the load map and the PSW. A loader error might be indicated in the listing or in the PSW. For a list of the loader wait state codes, see *VM/SP System Messages and Codes.*

   b. Correct the error, then reenter the VMFBLD command.

3. The NUCLEUS LOADED message gives the starting and ending cylinders/blocks for the nucleus. Remember that you are using the **current** CP directory to build this nucleus. Because you made changes to the current directory in Step 1, you should verify that this nucleus has not expanded into another allocated area, such as TEMP, WARM START, and so on, or unpredictable results could occur.

   An easy way to check this is to look at the *fn* DISKMAP file that you created after you tailored the directory. The LAST CYL/BLK USED=*nnnn* portion of the response displayed above must not exceed the $CPNUC$ ending cylinder/block defined in the directory. However, before you can look at the *fn* DISKMAP file, you must exit the WAIT state. Enter:

   #cp ipl 190 clear

Saving and Printing the CP Load Map

The CP load map has been spooled to MAINT's virtual printer. To save the load map on the CP object minidisk (MAINT 194), enter:

```
#cp spool prt nohold
#cp close prt
#cp ipl 190 clear
```

When VM READ appears in the corner of your screen, enter:

```
ENTER
access 194 c
query rdr
```

```
MAINT spoolid M PUN nnnnnnnn 001 NONE STANDARD
```

```
receive spoolid cpnuc map c
```

After you save the load map, you can print a copy of it by entering:

```
print cpnuc map c
```

#cp ipl 190 clear

This command is required only if you did not enter the command to look at the *fn* DISKMAP file or to save the CP load map.

access 295 b

- Place the updated CP directory on line:

access 190 c

Because the new DIRECT MODULE file is on the new CMS system minidisk (formerly 490), make sure that you have this minidisk accessed ahead of the old CMS system minidisk (now 590).

WARNING

When you enter the DIRECT command, the Auto-Deactivation of Restricted Passwords (ADRP) feature in the CP nucleus searches the CP directory for logon passwords that match the list of restricted passwords contained in the RPWLIST DATA file. All passwords that match are changed to NOLOG in the directory. You cannot log on to any virtual machine that has a NOLOG password. Before you enter the DIRECT command, make sure that your directory contains only non-restricted passwords. For a list of the restricted passwords, see Appendix B, "Restricted Logon Passwords" on page B-1.

direct *fn ft*

fn ft is the file ID of your CP directory file.

- Make sure that all users are off the system, then shut down:

drain all

This command stops spooling operations on all real unit record devices. However, any jobs in progress continue processing until they are done. **Be sure to wait until you receive a** `DRAINED` **response for each device.**

⋮

shutdown

⋮

Second Level Only

```
CP entered; disabled wait PSW '000A0000 00000008'
```

End of Second Level Only

- IPL your VMSRES volume:

First Level Only

IPL CLEAR the real address of your VMSRES volume according to the directions in the *Operator's Guide* for your machine.

End of First Level Only

Second Level Only

ipl *vdev* clear

vdev is the first level virtual address of your VMSRES volume, which functions as a "real" address at second level.

End of Second Level Only

```
VM/SP Release 6, Service level nnnn; created on mm/dd/yy at hh:mm:ss

It is now hh:mm:ss EDT day  mm/dd/yy
Change TOD clock (YES|NO) :
```

yes or **no**

At second level, enter **no**, because you do not have access to the physical hardware needed to change the TOD clock.

⋮

At first level, if you replied **yes** to the previous prompt, you are then prompted to enter the date and time:

- Enter the date in the format ***mm/dd/yy***.
- Enter the time in the format ***hh:mm:ss*** or ***hh.mm.ss***.

Check your hardware operation instructions to determine the correct method to enable the TOD clock.

```
DMKCPI971I System is uniprocessor generated
DMKUDR476I System directory loaded from volume VMSRES

DMKCPI974I No valid override file; using system defaults
```

```
DMKCPJ951I CP volid volid not mounted
```

If you receive this message, *volid* refers to a DASD volume listed in the SYSOWN macro in the DMKSYS file. The SYSOWN macro might contain some volumes that you are not using in your system, in which case you can ignore this message.

```
Start ((WARM|CKPT|FORCE|COLD)(DRAIN))|(SHUTDOWN):
```

cold

Because there is no data or accounting information to be recovered, you must request a cold start.

```
AUTO LOGON *** OPERATOR USERS=001 BY SYSTEM
```

You are now logged on to the OPERATOR virtual machine.

```
DMKCPJ952I nnnnnK system storage
```

For some processors, *nnnnn*K might be less than your actual system storage.

⋮

```
DMKCPI966I Initialization complete
```

■ Log on to the MAINT virtual machine by doing **one** of the following:

- Disconnect OPERATOR and log on to the same device as MAINT.

  or

- Enable another device and log on as MAINT (leaving OPERATOR logged on to the primary device).

Saving an IPLable CP Nucleus on Tape

After you log on as MAINT, you can use the UTILITY EXEC to save a copy of your CP nucleus on tape for recovery purposes. Enter:

```
utility nuctape
```

Notes:

1. To save the nucleus, you must have a scratch tape mounted on a tape drive in read/write mode attached to MAINT at virtual address 181.
2. The saved nucleus is not usable unless the **real** address of VMSRES is properly defined in DMKSYS (see page 4-27).

Defining More Than One CP Nucleus

You can improve system availability if you define and save more than one copy of your CP nucleus. Then, if the primary nucleus is damaged or unavailable, the system operator can select an alternate nucleus to IPL. For more information, see the *VM/SP Planning Guide and Reference*.

IPLing a CP Nucleus Saved on Tape

To retrieve a CP nucleus that you have saved on tape:

- If your system is down, IPL the **real** address of the tape drive.
- If your system is running, mount the tape at virtual address MAINT 181 and enter:

  ipl 181 clear

The new CP is written over the old CP nucleus at the address and location defined in the CP directory and DMKSYS.

Go back to page 4-35 and complete the processing for this step, beginning with the following prompt:

```
VM/SP Release 6, Service Level nnnn; created on mm/dd/yy at hh:mm:ss

It is now hh:mm:ss EDT day  mm/dd/yy
Change TOD clock (YES|NO) :
```

Step 8. Load HELP (Optional)

Overview

The primary tasks in this step are:

- Load the system HELP files
- Convert the files to uppercase (if required).

Procedure

■ Load the system HELP files:

```
VM/SP Release 6 mm/dd/yy hh:mm:ss
```

Or your own version ID.

spload help files

To use this command, the tape drive must be attached to MAINT at 181.

This command loads the HELP FILES tape file to the minidisk specified in SPLOAD PROFILE. The destination defined in the sample SPLOAD PROFILE is MAINT 19D. If you want to load to an address other than 19D, modify the HELP FILES entry in SPLOAD PROFILE before you issue the SPLOAD command. For additional information about the entries in this file, see "SPLOAD PROFILE Syntax" on page 8-19.

```
DMSWSL409I Loading HELP FILES to MAINT 19D
```

Note: Because there are many HELP files, the load operation could take 15 minutes or more.

```
Ready;
```

Converting the HELP Files to Uppercase

If you are building an uppercase English system, convert the HELP files from mixed case (as loaded) to uppercase by entering the following commands:

access *vdev fm*

copyfile * * *fm* = = = (upcase olddate replace

where ***vdev*** is the address of the minidisk where you loaded the HELP files and ***fm*** is the file mode at which you access that minidisk.

Saving the HELP Minidisk Directory in a Saved Segment

To improve system performance, you can save the HELP minidisk directory (file status table) in a saved segment. However, you must wait until after you have saved the CMS nucleus in a named saved system. Both procedures are described in Step 11.

Step 9. Update the SYSTEM NETID File (Optional)

Overview

If you have a Remote Spooling Communications Subsystem (RSCS) networking virtual machine defined in the CP directory, you should now update the SYSTEM NETID file located on the CMS system minidisk (MAINT 190). This file is referenced when you use CMS commands to communicate across the network.

The records in the SYSTEM NETID file have the format:

```
cpuid  nodeid  netid
```

where:

cpuid
is the CPU serial number.

nodeid
is the local node ID of the RSCS virtual machine, as defined in the LOCAL statement of the RSCS configuration file on the RSCS 191 (MAINT 59F) minidisk.

netid
is the user ID of the RSCS virtual machine, as defined in the CP directory.

When you use CMS commands to communicate via the network, the SYSTEM NETID file is referenced as follows:

1. To transmit notes, files, and messages, the NOTE, SENDFILE, and TELL commands call the IDENTIFY command.
2. The IDENTIFY command calls the QUERY CPUID command to retrieve the CPU serial number, then searches the SYSTEM NETID file for a matching serial number.
3. The IDENTIFY command calls the QUERY USERID command to retrieve the node identification, then compares it to the node in the SYSTEM NETID record.

   If there is a conflict in nodes between the SYSTEM NETID file and the response from QUERY USERID, the node in SYSTEM NETID takes precedence.

Procedure

Note: If you are installing an MP system, create a record in the SYSTEM NETID file for each processor that you want to be able to IPL.

If you are installing an AP system, create a SYSTEM NETID record only for the master processor.

■ To update the SYSTEM NETID file, enter:

set emsg on	This command provides message numbers for messages displayed in this step.

access 190 b

This command gives you write access to the CMS system minidisk.

query cpuid

This command obtains your CPU serial number.

```
CPUID = FFssssssdddd0000
```

ssssss is your CPU serial number; *dddd* is your processor device type.

xedit system netid b2

```
*CPUID NODEID NETID
013773 GDLPDP RSCS
```

This is a sample SYSTEM NETID file. Enter your CPU serial number (*ssssss*, obtained above) as the CPUID entry in this file. Replace the sample NODEID and NETID entries with the correct information for your installation.

file

release b

Note: These changes go into effect when you save your CMS system in Step 11.

Step 10. Load IPCS and Complete the CUSTOMER PROFILE

Overview

The primary tasks in this step are:

- Load the IPCS code
- Update the CUSTOMER PROFILE file.

Procedure

■ Load the IPCS code to the destinations specified in the SPLOAD PROFILE:

spload ipcs delta

This command loads the IPCS DELTA tape file, which contains IPCS delta service code (serviced text decks and update files).

```
DMSWSL409I Loading IPCS DELTA to MAINT 293
Ready;
```

spload ipcs apply

This command loads the IPCS APPLY tape file, which contains IPCS apply service code (auxiliary control files and update shells).

```
DMSWSL409I Loading IPCS APPLY to MAINT 392
Ready;
```

spload ipcs object

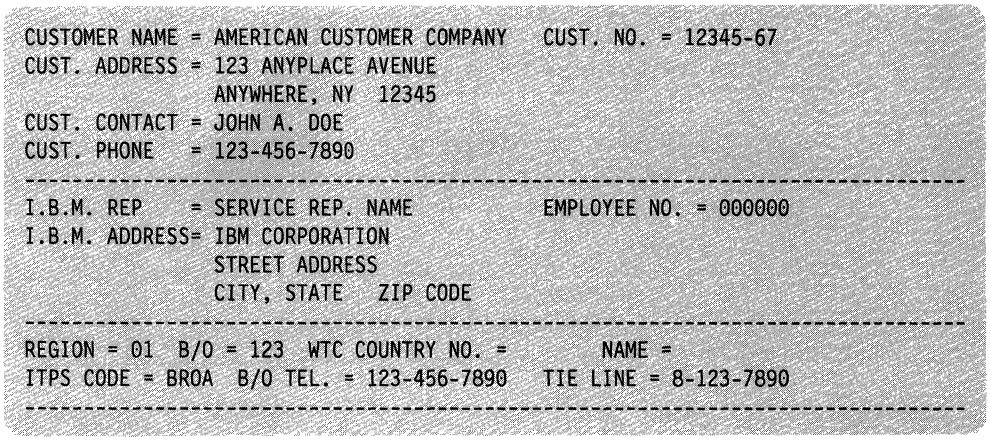
This command loads the IPCS OBJECT tape file, which contains IPCS object code.

```
DMSWSL409I Loading IPCS OBJECT to MAINT 193
Ready;
```

■ Edit the CUSTOMER PROFILE file:

access 193 r

xedit customer profile



```
CUSTOMER NAME = AMERICAN CUSTOMER COMPANY   CUST. NO. = 12345-67
CUST. ADDRESS = 123 ANYPLACE AVENUE
                ANYWHERE, NY  12345
CUST. CONTACT = JOHN A. DOE
CUST. PHONE   = 123-456-7890
--------------------------------------------------------------------------------
I.B.M. REP    = SERVICE REP. NAME           EMPLOYEE NO. = 000000
I.B.M. ADDRESS= IBM CORPORATION
                STREET ADDRESS
                CITY, STATE   ZIP CODE
--------------------------------------------------------------------------------
REGION = 01  B/O = 123  WTC COUNTRY NO. =        NAME =
ITPS CODE = BROA  B/O TEL. = 123-456-7890   TIE LINE = 8-123-7890
--------------------------------------------------------------------------------
```

The sample file (shown above) contains all of the standard fields that you need to submit Authorized Program Analysis Reports (APARs). As loaded from the product tape, these fields contain sample data to show the type of information required. Replace the sample data with the correct information for your installation.

If you need additional information to complete the CUSTOMER PROFILE, contact your IBM Branch Office.

■ Save the edited file:

file

Notes:

1. These changes go into effect when you save your CMS system in Step 11.
2. For information about using IPCS, see the *VM/SP Interactive Problem Control System Guide and Reference.*

Step 11. Save CMS and HELP

Overview

The primary tasks in this step are:

- Save the CMS nucleus
- Modify DMSNGP
- Enlarge the CMS nucleus to contain the Y minidisk directory (if required)
- Save the HELP minidisk directory (optional).

Procedure

■ Save the CMS nucleus:

ipl 190 clear parm savesys cms

This command saves the CMS nucleus as a named saved system using the name, size, and location parameters defined in DMKSNT.

```
SYSTEM SAVED
DMSWSP327I The installation saved segment could not be loaded
```

The CMS installation saved segment is not loaded and saved until after you complete the base installation.

This message does not appear if you indicated in the DMSNGP file, or in response to the DMSINQ296R prompt (see page 4-17), that you do not want to use this saved segment.

```
VM/SP Release 6 mm/dd/yy hh:mm:ss
```

Or your own version ID.

ENTER

■ Modify the DMSNGP ASSEMBLE file and change the SAVESYS and IPLADDR statements back to their original values:

– Change the SAVESYS statement to **YES**.

– Change the IPLADDR statement to **190**.

Enlarging the CMS Nucleus to Contain the Y Minidisk Directory

The next time you enter the **ipl cms** command to IPL the CMS saved system, if you receive the message:

```
DMSACC723I Y (19E) R/O
```

the Y minidisk directory (file status table, or Y-STAT) could not fit within the CMS nucleus that you just saved. In that case, every user who IPLs CMS gets the Y minidisk (19E) accessed in non-shared user free space, which is less efficient. You might want to enlarge your CMS nucleus to contain the Y-STAT. See Appendix F, "Enlarging the CMS Nucleus to Contain the Y Minidisk Directory (Y-STAT)" on page F-1.

Saving the HELP Minidisk Directory in a Saved Segment

If you loaded the system HELP files in Step 8, you can improve system performance by saving the HELP minidisk directory (file status table, or FST) in either a logical saved segment or a physical saved segment.

If you save the HELP directory in a logical saved segment, that saved segment can reside with other logical saved segments in a single physical saved segment. For more information about defining and saving a logical saved segment, see the *VM/SP Application Development Guide for CMS.*

If you would prefer to save the HELP directory in its own physical saved segment, the sample DMKSNT file contains a NAMESYS entry for a physical saved segment named HELP. To save the MAINT 19D minidisk directory in the HELP physical saved segment:

1. Define your virtual storage **less** than the address at which the HELP saved segment is to be loaded. For example, the sample DMKSNT defines the load address as 13248K; therefore, define your storage as 12M (12288K).

   To remove the disabled wait condition after you define your storage, enter:

   ipl cms

2. Enter the following commands to initialize and save the segment:

   savefd init 19d *label* help
   savefd save 19d *label* help

   where *label* is the CMS label assigned to the 19D minidisk.

3. Define your virtual storage back to 16M, then enter:

   ipl cms

For more information about the SAVEFD command, see the *VM/SP Application Development Guide for CMS.*

Step 12. Build the System File Pool

Overview

If you plan to install any of the remaining VM/SP components (GCS, TSAF, or AVS), you must build the system file pool, because code for these components is loaded (in later steps) to system file pool directories.

The name of the system file pool is VMSYS. The user ID for the system file pool server is VMSERVS.

To build the system file pool, you invoke the ITASK EXEC with the FILESERV VMSYS operands. ITASK:

- Accesses the MAINT 295 minidisk and obtains the logon password for the VMSERVS virtual machine and the write password for the AUTOLOG1 191 minidisk from the VMUSERS DIRECT file.
- Links to the VMSERVS 191 minidisk and copies the following files from MAINT 295:

  ```
  VMSYS POOLDEF
  VMSERVS DMSPARMS
  SERVER PROFILE
  ```

- Creates a tool called RUNSERVE EXEC on VMSERVS 191, then releases and detaches VMSERVS 191.
- Issues the AUTOLOG VMSERVS command to log the VMSERVS virtual machine on in a disconnected state.

  Note: MAINT is designated as the secondary user on the CONSOLE statement in the VMSERVS entry in the CP directory. The Single Console Image Facility allows the secondary user (MAINT) to become the controlling virtual machine when the primary user (VMSERVS) is disconnected. MAINT receives all console output from VMSERVS, and MAINT can issue commands on VMSERVS by using the CP SEND command. For more information about the Single Console Image Facility, see *VM/SP System Facilities for Programming.*

- (On VMSERVS) Accesses the MAINT 293 and MAINT 193 minidisks.
- (On VMSERVS) Invokes the RUNSERVE EXEC, which issues the FILESERV GENERATE VMSYS POOLDEF command to define and initialize the VMSYS file pool.
- (On VMSERVS) Renames SERVER PROFILE to PROFILE EXEC. This file contains the following commands:

  ```
  CP SET EMSG ON
  CP SET RUN ON
  SET AUTOREAD OFF
  ACCESS 293 B
  ACCESS 193 C
  EXEC FILESERV START
  ```

- (On VMSERVS) Issues the FILESERV START command to start the VMSERVS server machine using the start-up parameters contained in the VMSERVS DMSPARMS file. In this file, MAINT is designated as the first administrator of the VMSYS file pool.

- Links to the AUTOLOG1 191 minidisk and creates a PROFILE EXEC file that includes an AUTOLOG VMSERVS command, then releases and detaches AUTOLOG1 191.
- Issues the SET FILEPOOL VMSYS command to designate VMSYS as the default file pool for any file pool commands that MAINT issues.
- Issues the ENROLL USER command to enroll MAINT in the VMSYS file pool.
- Copies the SFS sample files from the MAINT 193 minidisk into the VMSYS:MAINT.SAMPLES directory.
- Issues the ENROLL PUBLIC command to let all users on this processor connect to the file pool.
- Issues GRANT AUTHORITY commands for PUBLIC to give read authority on the SAMPLES directory and on the individual sample files to all users on this processor.

Procedure

Notes:

1. In this step, ITASK looks for the VMUSERS DIRECT directory file. If your directory file has another name, create a temporary copy of your directory called VMUSERS DIRECT. If you plan to build a user file pool (in Step 13), keep the temporary directory file to use in that step. Otherwise, you can erase the file after the completion of this step.
2. If you have not already defined and formatted the VMSERVS 191 and AUTOLOG1 191 minidisks, you must do so before you invoke ITASK.

■ Build the system file pool:

itask fileserv vmsys

Note: The Single Console Image Facility routes console output from the disconnected VMSERVS server machine to MAINT with the prefix VMSERVS:.

⋮

```
VMSERVS: Ready for operator communications
Ready;
```

You might receive these messages in the reverse order.

Note: For information about enrolling users in the system file pool and other administrator tasks, see *VM/SP CMS Shared File System Administration.* However, you can wait until you complete the base VM/SP installation.

Step 13. Build the User File Pool (Optional)

Overview

The name of the sample user file pool is VMSYSU. The user ID for the user file pool server is VMSERVU.

To build the user file pool, you invoke the ITASK EXEC with the FILESERV VMSYSU operands. ITASK:

- Accesses the MAINT 295 minidisk and obtains the logon password for the VMSERVU virtual machine and the write password for the AUTOLOG1 191 minidisk from the VMUSERS DIRECT file.
- Links to the VMSERVU 191 minidisk and copies the following files from MAINT 295:

```
VMSYSU POOLDEF
VMSERVU DMSPARMS
SERVER PROFILE
```

- Creates a tool called RUNSERVE EXEC on VMSERVU 191, then releases and detaches VMSERVU 191.
- Issues the AUTOLOG VMSERVU command to log the VMSERVU virtual machine on in a disconnected state.

  Note: MAINT is designated as the secondary user on the CONSOLE statement in the VMSERVU entry in the CP directory. The Single Console Image Facility allows the secondary user (MAINT) to become the controlling virtual machine when the primary user (VMSERVU) is disconnected. MAINT receives all console output from VMSERVU, and MAINT can issue commands on VMSERVU by using the CP SEND command. For more information about the Single Console Image Facility, see *VM/SP System Facilities for Programming*.
- (On VMSERVU) Accesses the MAINT 293 and MAINT 193 minidisks.
- (On VMSERVU) Invokes the RUNSERVE EXEC, which issues the FILESERV GENERATE VMSYSU POOLDEF command to define and initialize the VMSYSU file pool.
- (On VMSERVU) Renames SERVER PROFILE to PROFILE EXEC. This file contains the following commands:

```
CP SET EMSG ON
CP SET RUN ON
SET AUTOREAD OFF
ACCESS 293 B
ACCESS 193 C
EXEC FILESERV START
```

- (On VMSERVU) Issues the FILESERV START command to start the VMSERVU server machine using the start-up parameters contained in the VMSERVU DMSPARMS file. In this file, MAINT is designated as the first administrator of the VMSYSU file pool.
- Links to the AUTOLOG1 191 minidisk and creates a PROFILE EXEC file that includes an AUTOLOG VMSERVU command, then releases and detaches AUTOLOG1 191.

Procedure

Notes:

1. In this step, ITASK looks for the VMUSERS DIRECT directory file. If your directory file has another name, create a temporary copy of your directory called VMUSERS DIRECT. You can erase the temporary file after the completion of this step.
2. If you have not already defined and formatted the VMSERVU 191 minidisk, you must do so before you invoke ITASK.

■ Build the user file pool:

itask fileserv vmsysu

Note: The Single Console Image Facility routes console output from the disconnected VMSERVU server machine to MAINT with the prefix `VMSERVU:`.

⋮

```
VMSERVU: Ready for operator communications
Ready;
```

You might receive these messages in the reverse order.

Note: For information about enrolling users in the user file pool and other administrator tasks, see *VM/SP CMS Shared File System Administration.* However, you can wait until you complete the base VM/SP installation.

Step 14. Load, Build, and Save GCS (Optional)

Overview

You must install GCS if you plan to install SNA products or RSCS Version 2.

Before you begin this step, make sure that you read "Planning for the Group Control System (GCS)" on page 2-4.

The primary tasks in this step are:

- Format the GCS system minidisk (if not done already)
- Load the GCS code
- Create a GCS configuration file (if required)
- Build and save the GCS nucleus
- Save and print the GCS load map (optional)
- Reconfigure GCS (if required).

Procedure

> **WARNING**
>
> Remove all previous-release GCS service from the MAINT 295 minidisk before you build the new GCS nucleus.

■ If the GCS system minidisk (MAINT 595) has not been formatted yet, format it now. The GCS nucleus will be written on this minidisk.

■ Load the GCS code to the destinations specified in the SPLOAD PROFILE:

Note: If the destination is an SFS directory that has not been created yet, the SPLOAD EXEC creates the directory before continuing with the load.

spload gcs delta

This command loads the GCS DELTA tape file, which contains GCS delta service code (serviced text decks and update files).

```
DMSWSL409I Loading GCS DELTA to VMSYS:MAINT.GCS.PUTSERV
```

spload gcs apply

This command loads the GCS APPLY tape file, which contains GCS apply service code (auxiliary control files and update shells).

```
DMSWSL409I Loading GCS APPLY to VMSYS:MAINT.GCS.APPLY
```

spload gcs interface

This command loads the GCS INTERFACE tape file, which contains files that enable GCS to interface with IPCS.

```
DMSWSL409I Loading GCS INTERFACE to MAINT 193
```

spload gcs object

This command loads the GCS OBJECT tape file, which contains GCS object code, including the GCSLOAD EXEC and GROUP EXEC files.

```
DMSWSL409I Loading GCS OBJECT to VMSYS:MAINT.GCS.OBJECT
```

■ Set up the GROUP EXEC messages in storage.

Notes:

1. If you are **rebuilding** a GCS nucleus, and your current configuration file does **not** require changes, you can skip that part of this step. Enter the following command:

   vmfsetup 5664167e gcs (access bld

   to set up your access order and then go directly to the nucleus build operation on page 4-59.

2. If you do not have a full screen display device, you cannot use the GROUP EXEC to create the configuration file, because you cannot display the panels. Therefore, you can skip this operation. Use the GCS build macros described in the *VM/SP Group Control System Command and Macro Reference* to build the configuration file manually. Then continue this step with the nucleus build operation on page 4-59

access 595 b

access vmsys:maint.gcs.object c

copyfile csimes text c csiume = a

This command creates a temporary GCS message file (containing the messages for the GROUP EXEC) that has the file name required by the SET LANGUAGE command.

set language ameng (add csi user

This command sets American English (AMENG) as the language of the temporary GCS message file and places the file in user storage.

■ Invoke the GROUP EXEC to create your GCS configuration file.

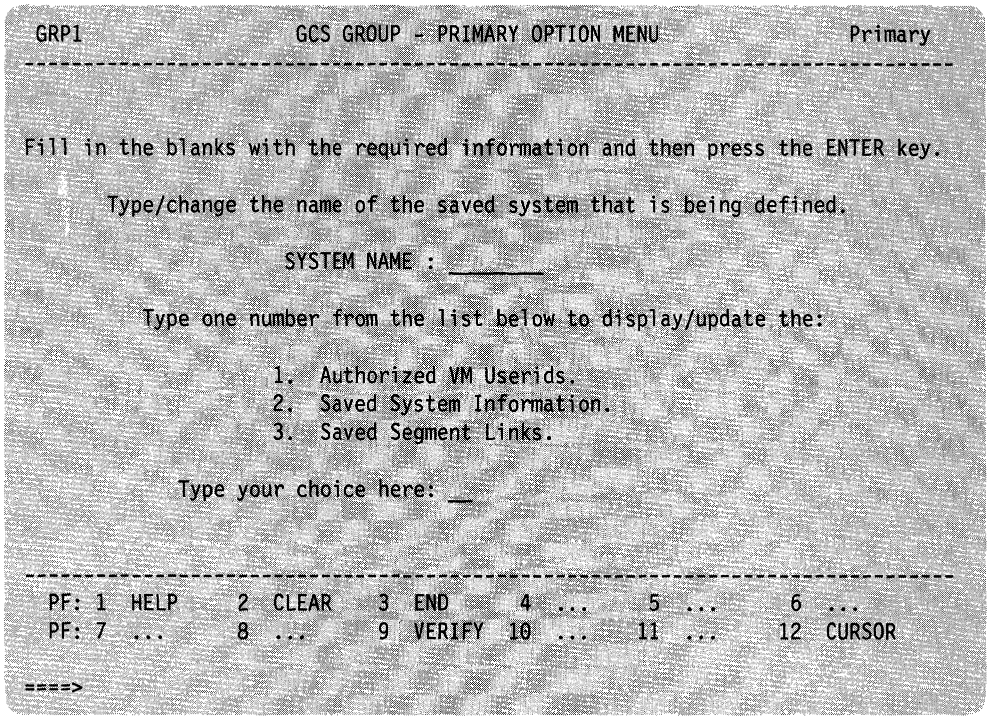
Use your Installation Reference Worksheet to help you complete the GROUP EXEC panels. For more information, see "Planning for the Group Control System (GCS)" on page 2-4.

group ***systemname***

systemname is the name of the GCS configuration file that you are creating. This command calls the Primary Option Menu. ***systemname*** is optional; if specified, it must match the SYSNAME entry in the DMKSNT file (the sample SYSNAME entry is **GCS**). If you specify ***systemname*** here, the Primary Option Menu appears with the SYSTEM NAME field filled in. If you do not specify ***systemname*** here, you must complete the SYSTEM NAME field on the Primary Option Menu.

Table 4-3. Function Keys Used with the GROUP EXEC Panels

KEY	FUNCTION
PF1 HELP	Shows information about a certain panel.
PF2 CLEAR	Clears the input areas where you enter information.
PF3 END	Leaves the present panel and returns you to a previous one. • If you press PF3 on the Primary Option Menu, you return to CMS. • If you press PF3 on any other screen, you return to the Primary Option Menu. • If you press PF3 after you enter information, but before you save it, you get the message: `577E File has been changed; type QQUIT to quit anyway`
PF5 REFRESH	Fills in the panel's input areas with the values you last saved there.
PF6 SAVE	Saves information that you entered on the panel (for the configuration file *systemname* GROUP).
PF7 PREVIOUS	Returns to the previous panel, if there is one.
PF8 NEXT	Moves ahead to the next panel, if there is one.
PF9 VERIFY	Checks to see if information that you entered on the panel also exists somewhere else (in a directory entry, for instance). If you press PF9, the GROUP EXEC runs a check and issues a message telling you how "valid" your information is.
PF12 CURSOR	Moves the cursor to the panel's command line.
PF4, PF10, PF11	Not Used
ENTER	Saves information that you entered and processes any valid CP or CMS command typed on the command line. Two specific commands you can enter are: 1. QQUIT, entered from the Primary Option Menu, returns you to CMS. If entered on any other panel, it returns you to the Primary Option Menu. 2. CANCEL, entered on any panel, returns you to CMS.



```
 GRP1                    GCS GROUP - PRIMARY OPTION MENU               Primary
------------------------------------------------------------------------------


Fill in the blanks with the required information and then press the ENTER key.

       Type/change the name of the saved system that is being defined.

                      SYSTEM NAME : ________

          Type one number from the list below to display/update the:

                     1.  Authorized VM Userids.
                     2.  Saved System Information.
                     3.  Saved Segment Links.

             Type your choice here: __


------------------------------------------------------------------------------
  PF: 1  HELP     2  CLEAR    3  END      4  ...     5  ...      6  ...
  PF: 7  ...      8  ...      9  VERIFY  10  ...    11  ...     12  CURSOR

====>
```

■ Complete the Primary Option Menu:

1. **Fill in or change the SYSTEM NAME.**
   If you invoked the GROUP EXEC with the ***systemname*** operand, this panel appears with the SYSTEM NAME already filled in.

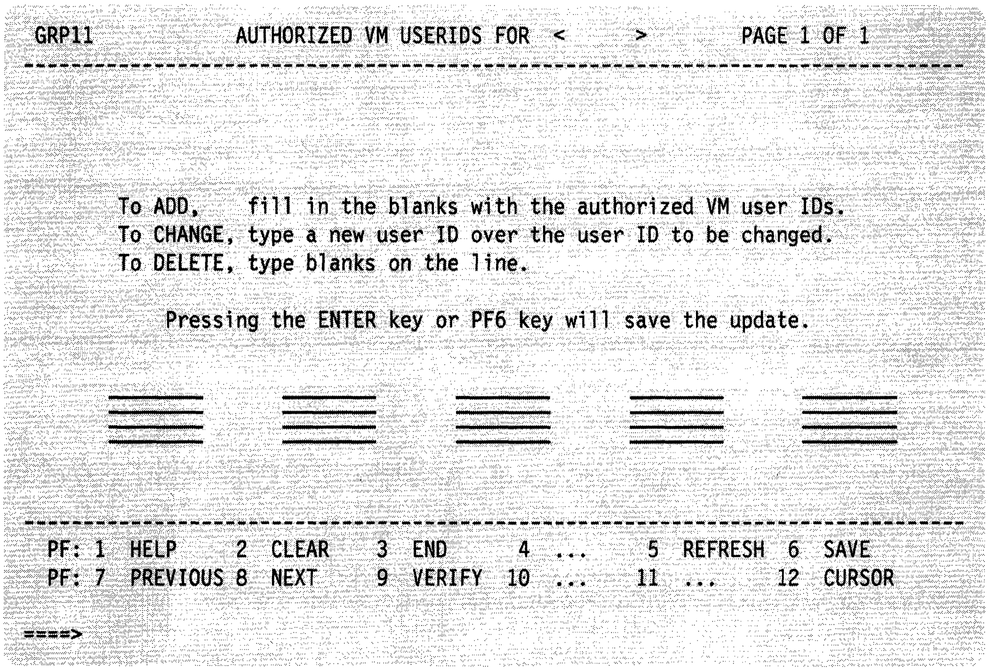
2. **Verify your input before selecting another screen.**
   Press PF9 to verify that this SYSTEM NAME has a valid entry in your system name table (DMKSNT). You receive two messages:

   a. Tells whether the SYSTEM NAME exists
   b. Asks you to enter a selection number.

   Note: If a configuration file for this SYSTEM NAME already exists (for example, MYGCS GROUP), the GROUP EXEC makes a backup copy of it, with a different file type, before you start making changes:

MYGCS OLDGROUP	(original file, new file type)
MYGCS GROUP	(same file, ready for changes)

3. **Select the next panel (1, 2, or 3) and press ENTER.**
   The first time you leave the Primary Option Menu, select panel 1; the next time, panel 2; the third time, panel 3.



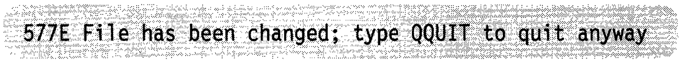
```
 GRP11          AUTHORIZED VM USERIDS FOR <      >       PAGE 1 OF 1
--------------------------------------------------------------------------------



         To ADD,    fill in the blanks with the authorized VM user IDs.
         To CHANGE, type a new user ID over the user ID to be changed.
         To DELETE, type blanks on the line.

             Pressing the ENTER key or PF6 key will save the update.


       ________     ________     ________     ________     ________
       ________     ________     ________     ________     ________
       ________     ________     ________     ________     ________
       ________     ________     ________     ________     ________


--------------------------------------------------------------------------------
 PF: 1  HELP     2  CLEAR    3  END      4 ...     5  REFRESH  6  SAVE
 PF: 7  PREVIOUS 8  NEXT     9  VERIFY  10 ...    11 ...     12  CURSOR

====>
```

- Complete the Authorized User IDs panel:

  1. **Type in the user ID(s)**.
     Press ENTER or PF6 after typing each user ID to save your information and advance to the next space at the same time. Every time you press either of these keys, the GROUP EXEC tells you how many user IDs it has processed. Use PF7 and PF8 if you have more than one screenful of user IDs.

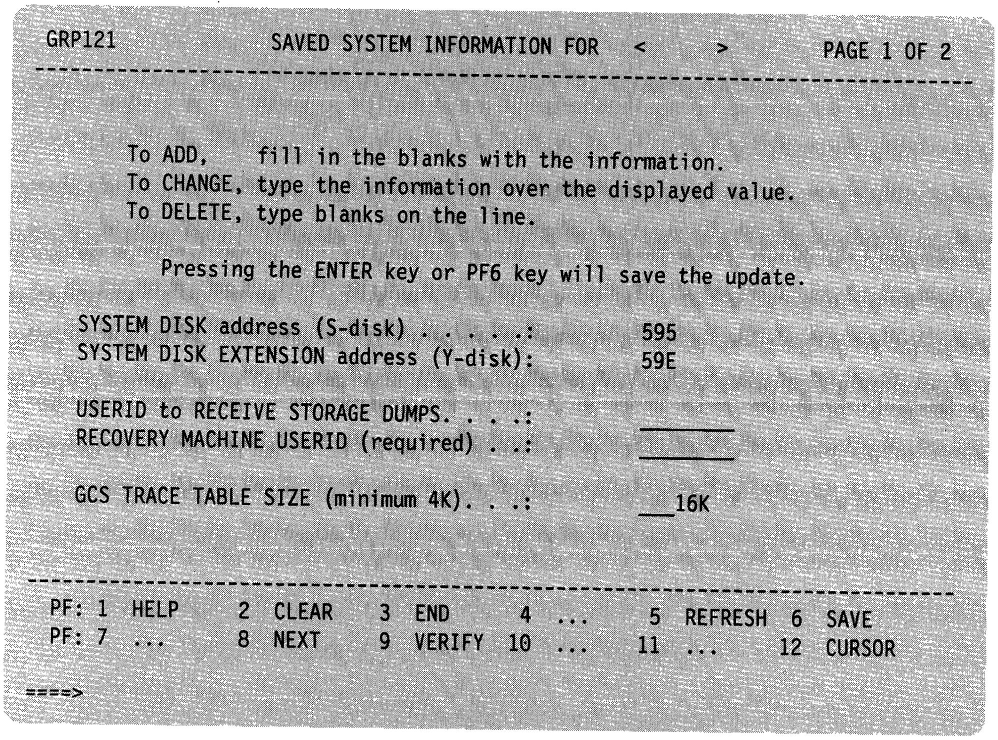
  2. **Verify your input**.
     Press PF9 . The GROUP EXEC tells you how many user IDs on the page being viewed are valid or invalid and places arrows beside any user IDs not found in the CP directory.

  3. **Return to the Primary Option Menu**.
     Press PF3 . If you forgot to press ENTER or PF6 , you will remain on this screen and see the message:

     ```
     577E File has been changed; type QQUIT to quit anyway
     ```

     To continue, simply press PF6 . You will receive a message telling you how many authorized user IDs have been processed. Press PF3 again to return to the Primary Option Menu.



```
 GRP121              SAVED SYSTEM INFORMATION FOR  <       >        PAGE 1 OF 2
-------------------------------------------------------------------------------


          To ADD,    fill in the blanks with the information.
          To CHANGE, type the information over the displayed value.
          To DELETE, type blanks on the line.

             Pressing the ENTER key or PF6 key will save the update.

      SYSTEM DISK address (S-disk) . . . . .:          595
      SYSTEM DISK EXTENSION address (Y-disk):          59E

      USERID to RECEIVE STORAGE DUMPS. . . .:          ________
      RECOVERY MACHINE USERID (required) . .:          ________

      GCS TRACE TABLE SIZE (minimum 4K). . .:          ___16K


-------------------------------------------------------------------------------
 PF: 1  HELP     2  CLEAR    3  END     4  ...     5  REFRESH  6  SAVE
 PF: 7  ...      8  NEXT     9  VERIFY  10 ...     11 ...      12 CURSOR

====>
```

- Complete the first Saved System Information panel:

1. **Specify your disk addresses.**
   The default 595 and 59E virtual addresses are already saved and verified. If you must choose different addresses, type over the default values. Make sure the new information is correct; the GROUP EXEC does not prevent you from saving invalid information.

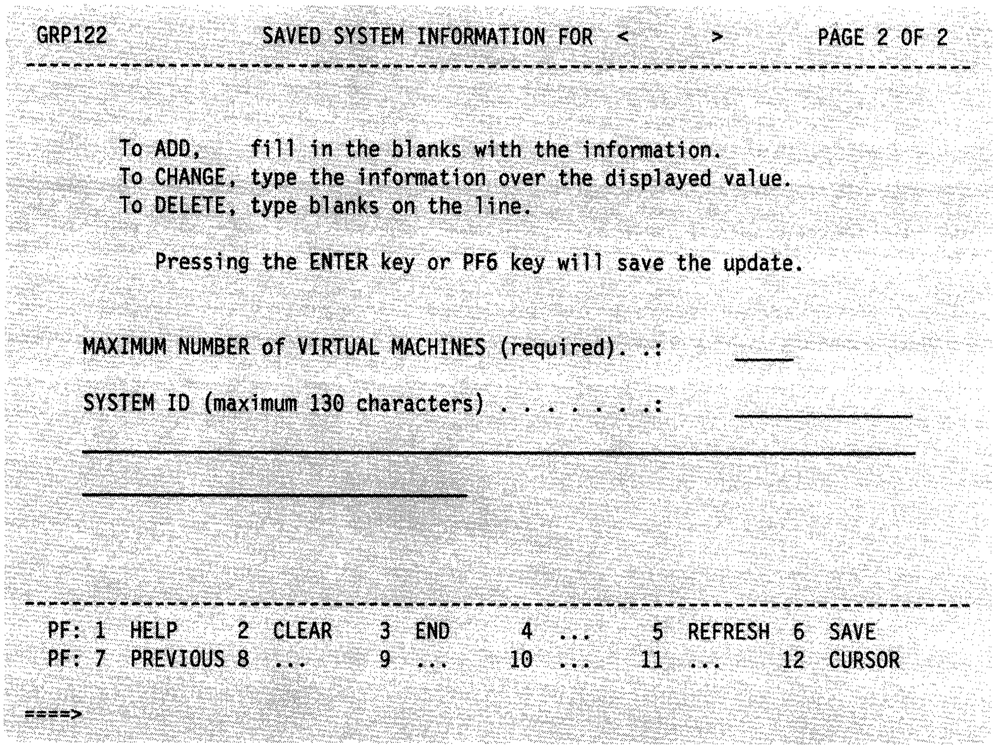
2. **Name an authorized dump receiver.**
   Name an authorized recovery machine.
   GCS requires you to enter a recovery machine user ID. The GROUP EXEC tells you whether that user ID is valid, but it does not prevent you from saving an invalid entry.

3. **Specify a trace table size.**
   By default, the GROUP EXEC saves a value of 16K. If you decide to save a different amount, simply type the new value over the default value.

4. **Verify your input.**
   Press PF9 to verify that the dump receiver and recovery machine user IDs have valid entries in the CP directory. If your dump receiver also appears on the other panel of Authorized VM user IDs, the GROUP EXEC verifies it as a "Valid Authorized" user ID. Make sure any new SYSTEM DISK or TRACE TABLE information is correct; the GROUP EXEC does not prevent you from saving invalid information.

5. **Save your input.**
   If you are satisfied with your choices, press PF6 or ENTER to save the information.

6. **Go on to the second panel.**
   Press PF8 to continue on the second Saved System Information panel.



```
GRP122              SAVED SYSTEM INFORMATION FOR <       >        PAGE 2 OF 2
--------------------------------------------------------------------------------


       To ADD,    fill in the blanks with the information.
       To CHANGE, type the information over the displayed value.
       To DELETE, type blanks on the line.

          Pressing the ENTER key or PF6 key will save the update.


    MAXIMUM NUMBER of VIRTUAL MACHINES (required). .:     ____

    SYSTEM ID (maximum 130 characters) . . . . . . .:     _______________
    ______________________________________________________________________

    ________________________________



--------------------------------------------------------------------------------
 PF: 1  HELP      2  CLEAR    3  END      4  ...     5  REFRESH  6  SAVE
 PF: 7  PREVIOUS  8  ...      9  ...     10  ...    11  ...     12  CURSOR

====>
```

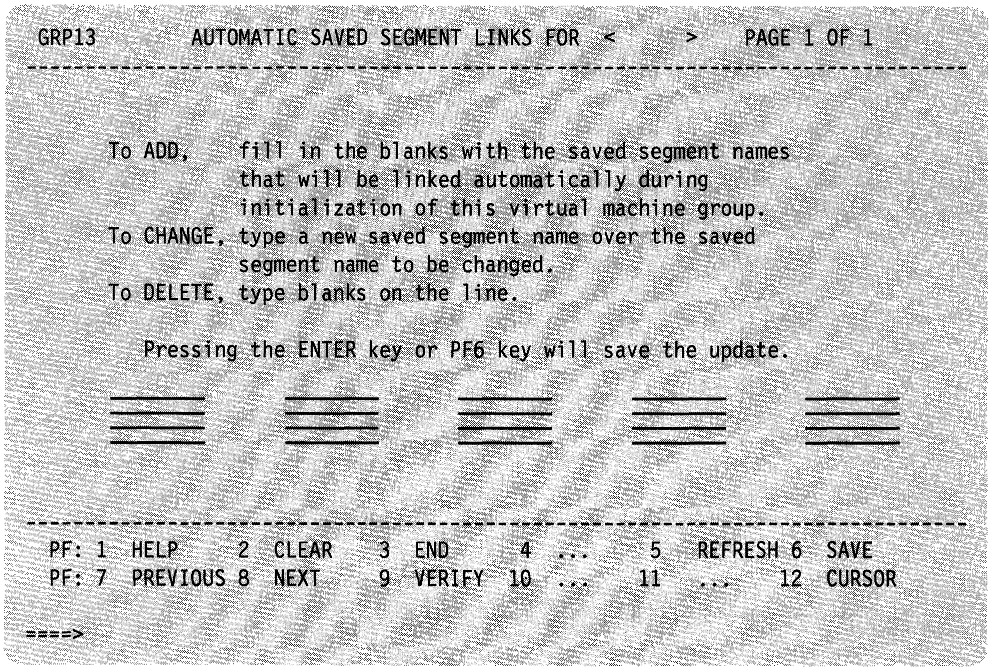
- Complete the second Saved System Information panel:

  1. **Specify a MAXIMUM NUMBER.**
     Simply type a value of **1** or more, and press ENTER. The GROUP EXEC responds with a message *only* if you enter an invalid value.

  2. **Type in your SYSTEM ID text.**
     This is optional. Move the cursor to the SYSTEM ID space (if it is not already there) and enter your text. The GROUP EXEC does not let you enter any more than 130 characters.

  3. **Save your input and return to the Primary Option Menu.**
     Press ENTER to save your information and then press PF3.

Step 14



```
GRP13        AUTOMATIC SAVED SEGMENT LINKS FOR  <       >    PAGE 1 OF 1
------------------------------------------------------------------------------


      To ADD,    fill in the blanks with the saved segment names
                 that will be linked automatically during
                 initialization of this virtual machine group.
      To CHANGE, type a new saved segment name over the saved
                 segment name to be changed.
      To DELETE, type blanks on the line.

         Pressing the ENTER key or PF6 key will save the update.

      ________     ________     ________     ________     ________
      ________     ________     ________     ________     ________
      ________     ________     ________     ________     ________
      ________     ________     ________     ________     ________


------------------------------------------------------------------------------
 PF: 1  HELP     2  CLEAR    3  END     4  ...     5   REFRESH 6  SAVE
 PF: 7  PREVIOUS 8  NEXT     9  VERIFY  10 ...     11  ...     12 CURSOR

====>
```

■ Complete the Automatic Saved Segment Links panel:

Note: Do not include CMSBAM or CMSVSAM here. For more information about these VSAM segments, see Chapter 6.

1. **Type in the segment name(s).**
   Press ENTER or PF6 after typing each segment name to save your information and advance to the next space at the same time. Every time you press either of these keys, the EXEC tells you how many segment names it has processed. Use PF7 and PF8 if you more than one screen of names.

2. **Verify your input.**
   Press PF9. The GROUP EXEC tells you how many segment names on the page being viewed are valid or invalid and places arrows beside any names it does not find in your system name table (DMKSNT).

3. **Return to the Primary Option Menu.**
   If you have not saved your input yet, press PF6. Press PF3 to return to the Primary Option Menu.

- After you provide the necessary input on all panels and return to the Primary Option Menu for the last time, press PF3 to exit from the GROUP EXEC.
- Remove the temporary GCS message file from user storage and erase it from file mode A:

set language ameng (delete csi user

erase csiume text a

- Invoke the ITASK EXEC to build and save the GCS nucleus. ITASK:
  - Modifies a copy of the GCS loadlist (GCSLOAD EXEC) and updates the entry that contains the name of the GCS configuration file (the default is GCS) to match the name of the configuration file that you just created (the default is also GCS), then files the modified loadlist on MAINT 295
  - Renames the GCS configuration file type from GROUP to ASSEMBLE
  - Calls the VMFASM EXEC to assemble the configuration file
  - Calls the VMFBLD EXEC to build the GCS nucleus.

itask build gcs ***systemname***

systemname is the name of the GCS configuration file that you just created. If you do not specify a name, the default is **GCS**.

```
DMSUPD181E No update files were found
DMSOPN002W File DMKSP MACLIB * not found
DMSWAM1907I Assembling systemname
```

⋮

If You Receive an Assembly Error

1. Reestablish the nucleus build access order:

   vmfsetup 5664167e gcs (access bld

2. Check your configuration file (which ITASK has renamed to *systemname* ASSEMBLE) for incorrect or incomplete entries. For information about the required fields, see "Planning for the Group Control System (GCS)" on page 2-4.

3. Correct the configuration file or go through the GROUP EXEC panels again to build a new file.

   Note: If you decide to build a new file, you must use the same *systemname*. After you build the new file (*systemname* GROUP), rename or erase the old file (*systemname* ASSEMBLE), then rename the file type of the new file from GROUP to ASSEMBLE.

4. Assemble the configuration file:

   vmfasm *systemname* 5664167e gcs (ppf

5. Build the nucleus:

   vmfbld 5664167e gcs gcsload

⋮

```
MSG FROM MAINT: CSIINI134I systemname has nnnnnn bytes of available common free storage

SYSTEM SAVED
```

```
PRT FILE spoolid TO MAINT COPY 001 NOHOLD
```

This message indicates that the GCS load map file has been sent to MAINT's virtual reader. If you plan to save or print the GCS load map, record the *spoolid.*

```
SYSTEM RESET
DMKDSP450W CP entered;  disabled wait PSW '00020000 00000000'
```

■ IPL CMS:

ipl cms

```
VM/SP Release 6 mm/dd/yy hh:mm:ss
```

Or your own version ID.

ENTER

```
Ready;
```

Saving and Printing the GCS Load Map

The GCS load map has been sent to MAINT's virtual reader. To save the load map in the GCS object directory (VMSYS:MAINT.GCS.OBJECT), enter:

```
access vmsys:maint.gcs.object e
receive spoolid gcsnuc map e
```

After you save the load map, you can print a copy of it by entering:

```
print gcsnuc map e
```

Reconfiguring GCS

After you install GCS, you might want to alter the GCS configuration:

- To modify GCS (to add new authorized users, for example), you must modify the existing GCS configuration file and rebuild the GCS nucleus.
- To move or enlarge GCS, see Appendix H, "Example of Alternate GCS Nucleus Placement" on page H-1.
- To install multiple GCS nuclei, you must build a new configuration file for each additional nucleus. Additional systems must be defined in DMKSNT and in the CP directory.

To modify an existing system or add a new system, you can follow a similar sequence of steps. If you are modifying your system, do only the steps marked with an asterisk (*). If you are installing a new system, do only the steps marked with a plus (+).

1. * + Reestablish the nucleus build access order:

   ```
   vmfsetup 5664167e gcs (access bld
   ```

2. * + (Full screen display devices only) Set up the GROUP EXEC messages in storage:

   ```
   copyfile csimes text b csiume = a
   set language ameng (add csi user
   ```

3. *Rename the file type of the existing configuration file:

   ```
   rename systemname assemble a = group =
   ```

4. * + Build or modify the configuration file:

   - If you are using a full-screen display device, enter:

     ```
     group systemname
     ```

     and complete the panels to define the system parameters. If this a configuration file for an additional nucleus, ***systemname*** must be unique.

   - If you are **not** using a full-screen display device, you must build the configuration file manually, using the build macros described in the *VM/SP Group Control System Command and Macro Reference.*

5. * + (Full-screen display devices only) Remove the temporary GCS message file from user storage and erase it from file mode A:

```
set language ameng (delete csi user
erase csiume text a
```

6. + Edit the GCS loadlist:

```
xedit gcsload exec
```

and change the configuration file entry (&1 &2 &3 *fn*, where *fn* is the name of the configuration file for the previously-generated GCS nucleus) to specify the name of the configuration file for this nucleus.

7. * + Rename the file type of the configuration file:

```
rename systemname group a = assemble =
```

8. * + Assemble the configuration file:

```
vmfasm systemname 5664167e gcs (ppf
```

9. * + Build and save the GCS nucleus:

```
vmfbld 5664167e gcs gcsload
```

10. * + Save or print the GCS load map. Remember to give each load map file a different name.

Step 15. Load TSAF (Optional)

Overview

The Transparent Services Access Facility (TSAF) is shipped as a ready-to-run system. In this step, you load the TSAF code from the product tape; no build is required. For more information about TSAF, see *VM/SP Connectivity Planning, Administration, and Operation.*

Procedure

■ Load the TSAF code to the destinations specified in the SPLOAD PROFILE:

Note: If the destination is an SFS directory that has not been created yet, the SPLOAD EXEC creates the directory before continuing with the load.

spload tsaf delta

This command loads the TSAF DELTA tape file, which contains TSAF delta service code (serviced text decks and update files).

```
DMSWSL409I Loading TSAF DELTA to VMSYS:MAINT.TSAF.PUTSERV
```

spload tsaf apply

This command loads the TSAF APPLY tape file, which contains TSAF apply service code (auxiliary control files and update shells).

```
DMSWSL409I Loading TSAF APPLY to VMSYS:MAINT.TSAF.APPLY
```

spload tsaf interface

This command loads the TSAF INTERFACE tape file, which contains files that enable TSAF to interface with IPCS.

```
DMSWSL409I Loading TSAF INTERFACE to MAINT 193
```

spload tsaf object

This command loads the TSAF OBJECT tape file, which contains the ready-to-run TSAF module.

```
DMSWSL409I Loading TSAF OBJECT to VMSYS:MAINT.TSAF.OBJECT
```

Step 16. Load AVS (Optional)

Overview

APPC/VM VTAM Support (AVS) is shipped as a LOADLIB; no build is required. The primary tasks in this step are:

- Load the AVS code from the product tape
- Copy certain AVS files to the AVSVM 191 minidisk.

For more information about AVS, see *VM/SP Connectivity Planning, Administration, and Operation.*

Procedure

■ Load the AVS code to the destinations specified in the SPLOAD PROFILE:

Note: If the destination is an SFS directory that has not been created yet, the SPLOAD EXEC creates the directory before continuing with the load.

spload avs delta

This command loads the AVS DELTA tape file, which contains AVS delta service code (serviced text decks and update files).

```
DMSWSL409I Loading AVS DELTA to VMSYS:MAINT.AVS.PUTSERV
```

spload avs apply

This command loads the AVS APPLY tape file, which contains AVS apply service code (auxiliary control files and update shells).

```
DMSWSL409I Loading AVS APPLY to VMSYS:MAINT.AVS.APPLY
```

spload avs interface

This command loads the AVS INTERFACE tape file, which contains files that enable AVS to interface with IPCS.

```
DMSWSL409I Loading AVS INTERFACE to MAINT 193
```

spload avs object

This command loads the AVS OBJECT tape file, which contains the ready-to-run AVS module.

```
DMSWSL409I Loading AVS OBJECT to VMSYS:MAINT.AVS.OBJECT
```

■ Copy the AVS files from the object directory to the AVSVM 191 minidisk:

Note: If you did not define and format the AVSVM 191 minidisk in Steps 1 and 2, you must do so now.

```
access vmsys:maint.avs.object f

link avsvm 191 091 wr password

access 091 g

copyfile gcs profile f = = g

copyfile agw loadlib f = = g

release f

release g (det
```

Step 17. Load the Source Feature Tape (Optional)

Overview

Source code for the GCS and TSAF components and additional CMS modules is provided on a separately-orderable feature tape. Destinations for these tape files are specified in the SPLOAD PROFILE:

Tape File	**Destination**
GCS SOURCE	VMSYS:MAINT.GCS.SOURCE
TSAF SOURCE	VMSYS:MAINT.TSAF.SOURCE
CMSFT SOURCE	MAINT 393

Note: All source files are loaded down in **packed** format. After the files are loaded, you can unpack individual files by using the COPYFILE command with the UNPACK option. For more information about this option, see the *VM/SP CMS Command Reference.*

The recommended MAINT 393 minidisk size (see Appendix A, "Minidisks and SFS Directories Reserved for MAINT" on page A-1) allows enough space for all of the packed files plus some (but not all) unpacked modules. If you plan to unpack all of your files, you should redefine this minidisk 80% larger.

Procedure

- Mount the source feature tape on device 181.
- Load the source code:

itask load *group*src

group refers to the GROUP portion of the tape file identifier (GROUP ELEMENT) used in the SPLOAD PROFILE: **gcs**, **tsaf**, or **cmsft**.

```
DMSWSL409I Loading group SOURCE to dirname|userid vdev
```

- Enter the previous command for each source tape file that you want to load.

Step 18. Install EREP (Optional)

Overview

If you ordered the Environmental Record Editing and Printing program (EREP), this licensed program is supplied on a separate distribution tape. In this step, you invoke the INSTFPP EXEC installation tool to load the EREP files to the EREP minidisk (MAINT 201 is the default).

Procedure

- Mount the EREP tape.
- Load the EREP code:

instfpp or **instfpp (all** The default is **all**.

ENTER

```
REWIND COMPLETE
DMKVDD040E Device 001 does not exist
NOW FORMATTING TEMPORARY WORKDISK "001" WITH FILEMODE "C."
FORMATTING OF TEMPORARY WORKDISK "001/C" IS COMPLETE.

nnnn-nnn VM/EREP VER. n REL. n MOD. LEVEL n SL nnn
DO YOU WISH TO INSTALL THIS PRODUCT NOW (YES OR NO)?
```

yes

```
PRT FILE nnnn TO MAINT  COPY nnn NOHOLD

xxxxxxxx MEMO HAS BEEN PRINTED

THE DEFAULT MINIDISK FOR CPEREP INTERFACE FILES ON EREP n.n.n IS:
*______ _______ ______*  ( 201 )  *______ _______ ______*

    ENTER   A NULL LINE TO ACCEPT DEFAULT
                    OR
             NEW MINIDISK CUU
```

ENTER or ***vdev*** Press ENTER to accept the default (**201**) or enter the new minidisk address (***vdev***).

```
LOADING CPEREP INTERFACE FILES FROM DISTRIBUTION TAPE.....
CPEREP MODULE SUCCESSFULLY CREATED
LOADING EREP TXTLIB FILES FROM DISTRIBUTION TAPE.....

xxxxxxxx - VM/EREP n.n.n INSTALLATION COMPLETED SUCCESSFULLY

NOW UPDATING THE PROD LEVEL P FILE...


LEAVING INSTFPP EXEC...
REWIND COMPLETE
```

Supplementary Procedures

You have now completed the base VM/SP installation. However, you might also want to complete the following supplementary procedures:

- **Do a full system backup.**
- **Verify your installed system.**

  To make sure that CP and CMS are working properly, you can use the Installation Verification Procedure (IVP). See Chapter 5, "Verifying Your Installed VM/SP System" on page 5-1.
- **Install saved segments.**

  For procedures to install the CMSDOS, CMSBAM, CMSVSAM, CMSAMS, CMSINST, CMSVMLIB, and CMSFILES saved segments, see Chapter 6, "Installing Saved Segments" on page 6-1.
- **Install a new system national language.**

  If you received a feature tape for a language that you want to install as the new system national language (replacing American English), see Chapter 7, "Installing a New System National Language" on page 7-1.
- **Install additional national languages.**

  If you received a feature tape for a language that you want to install on your system as an option for selected users to supplement or override the system national language, see *VM System Facilities for Programming.*
- **Apply Service.**

  If a PUT (Program Update Tape) was included with your shipment, you might want to apply some or all of the service updates that it contains. For information about VM/SP service, see the *VM/SP Service Guide.*

Chapter 5. Verifying Your Installed VM/SP System

Overview

The Installation Verification Procedure (IVP) is optional, but recommended for the formal completion of the VM/SP installation process. The IVP consists of two files that use the EXEC facility of CMS to verify that CP and CMS are working properly.

The IVP tests verify the following CP functions:

- Automatic warm start after abnormal termination of VM/SP
- Correct EC level for machines with ECPS/VSE
- Message transmission to the system operator
- Spooled data transfer to other virtual machines
- I/O spooling
- Disk I/O support
- Off-line I/O operations
- Paging operations
- Task dispatching and scheduling
- Multiple virtual machine support.

The IVP tests verify the following CMS functions:

- Creating and modifying files with the XEDIT command
- Copying files
- Printing and punching CMS files
- Formatting minidisks
- Assembling executable programs
- Executing user programs
- Stacking and unstacking command and data input from the terminal
- Using multilevel nested EXEC procedures
- Communicating with the user from EXEC procedures
- Creating and executing user-written commands
- Processing normal CMS commands
- Issuing commands to CP.

The IVP **does not** verify system facilities such as preferred execution options, virtual=real, OS ISAM, and the Remote Spooling Communications Subsystem (RSCS).

The IVP requires operator intervention only to start the IVP tests and to make certain decisions. All file creation, erasure, management, and logoff of the virtual machines (with the exception of the system operator) at test completion takes place without operator or user action. The IVP tests use only the system-provided facilities to create, assemble, and erase all unique test programs.

Requirements

All VM/SP configurations are supported. The IVP executes under the control of CMS.

For the **full-function** IVP, you need:

- The system operator's console and virtual machine
- One additional terminal
- The IVPM1 and IVPM2 virtual machines (defined in the sample CP directory), **with NOLOG entries changed to valid non-restricted passwords.**
- OPERATOR's reader and punch defined to the same spool class.

The facilities required for the IVPM1 and IVPM2 virtual machines are:

- The system assembler
- One R/W minidisk (usually 191) accessed as file mode A
- 16M of virtual storage for IVPM1; 3M for IVPM2.

Variations

Using Virtual Machines Other Than IVPM1 and IVPM2

You can execute the IVP with virtual machines other than IVPM1 and IVPM2. Simply specify the user ID of the desired virtual machine when you enter the IVP 1 and/or IVP 2 command. This option is shown in the procedure.

Using a Single Virtual Machine

You can run a variation of the IVP in a single virtual machine. In this variation, inter-machine functions, such as transferring data between virtual machines, are not verified. Inter-machine transfer of data is simulated by transferring virtual punched output to the same virtual machine's virtual card reader. The abnormal termination test is not run. The single-machine procedure begins on page 5-9.

Procedures

Full-Function (Two Virtual Machine) Procedure

- Log on to the system console as OPERATOR and set up OPERATOR's virtual machine:

ipl cms

```
VM/SP Release 6 mm/dd/yy hh:mm:ss
```

Or your own version identification.

ENTER

terminal mode vm

This command enables you to enter CMS commands.

link maint 193 193 rr

The IVP EXECs are located on this minidisk.

```
Enter read password:
```

password

Enter the minidisk readshare password defined in the directory.

access 193 t

- Invoke the IVP:

ivp

```
*** ARE YOU THE SYSTEM OPERATOR?  ENTER ''YES'' OR ''NO'':
```

yes

If you reply **no**, you default to the single virtual machine verification procedure (see page 5-9).

```
*** STARTING TEST SECTION 0
CP QUERY V ALL
```

⋮

```
*** THIS PORTION OF IVP NOW GOING TO SLEEP.
CP SLEEP
```

- Log on to your second terminal as IVPM1:

logon ivpm1

```
Enter password:
```

password

Enter the logon password that you defined for this virtual machine in the CP directory during the installation procedure.

```
LOGON AT hh:mm:ss EST day  mm/dd/yy
```

- Define your storage and IPL CMS:

define storage 16m

```
STORAGE=16384K
```

ipl cms

```
VM/SP Release 6 mm/dd/yy hh:mm:ss
```

Or your own version identification.

Formatting the IVPM1 191 Minidisk

- If the IVPM1 191 minidisk is not already formatted, format it now:

access (nodisk

This command prevents CMS from automatically accessing the IVPM1 191 minidisk. If CMS tried to access 191, you would receive an error message, because 191 has not been initialized.

format 191 a

By default, this command formats the minidisk using the block size that is optimal for performance and data storage for the DASD type. If you want to format the minidisk with a specific block size, use the BLKSIZE option. For more information about block size defaults and options, see the *VM/SP CMS Command Reference*.

```
FORMAT will erase all files on disk A(191).
Do you wish to continue?  Enter 1 (YES) or 0 (NO).
```

1

```
Enter disk label:
```

label

label is the 1-to-6-character label, such as **ivp1**, that you assign to this minidisk. Select any label that you want.

```
Formatting disk A
nnnnn cylinders|FB-512 blocks  formatted on disk A(191)
```

nnnnn is the number of cylinders or blocks assigned to this minidisk in the CP directory.

End of Formatting the IVPM1 191 Minidisk

ENTER

This entry is necessary only if you did not format the IVPM1 191 minidisk.

- Invoke the first IVP procedure:

access 193 t

The IVP tests are on this minidisk.

ivp 1 ***userid1***

Enter ***userid1*** only if you want to use a virtual machine other than IVPM1 to perform these tests.

```
*** STARTING TEST SECTION 1
CP SPOOL D TO IVPM2
```

⋮

```
*** THIS PORTION OF IVP NOW DISCONNECTING.
CP DISC HOLD
```

- Log on as IVPM2 (using the same terminal):

logon ivpm2

```
Enter password:
```

password

Enter the logon password that you defined for this virtual machine in the CP directory during the installation procedure.

```
LOGON AT hh:mm:ss EST day  mm/dd/yy
```

- IPL CMS:

ipl cms

```
VM/SP Release 6 mm/dd/yy hh:mm:ss
```

Or your own version identification.

Formatting the IVPM2 191 Minidisk

- If the IVPM2 191 minidisk is not already formatted, perform the following sequence:

access (nodisk

This command prevents CMS from automatically accessing the IVPM2 191 minidisk. If CMS tried to access 191, you would receive an error message, because 191 has not been initialized.

format 191 a

By default, this command formats the minidisk using the block size that is optimal for performance and data storage for the DASD type. If you want to format the minidisk with a specific block size, use the BLKSIZE option. For more information about block size defaults and options, see the *VM/SP CMS Command Reference.*

```
FORMAT will erase all files on disk A(191).
Do you wish to continue?  Enter 1 (YES) or 0 (NO).
```

1

```
Enter disk label:
```

label

label is the 1-to-6-character label, such as **ivp2**, that you assign to this minidisk. Select any label that you want.

```
Formatting disk A
nnnnn cylinders|FB-512 blocks  formatted on disk A(191)
```

nnnnn is the number of cylinders or blocks assigned to this minidisk in the CP directory.

End of Formatting the IVPM2 191 Minidisk

ENTER

This entry is necessary only if you did not format the IVPM2 191 minidisk.

- Invoke the second IVP procedure:

access 193 t

The IVP tests are on this minidisk.

ivp 2 *userid2*

Enter ***userid2*** only if you want to use a virtual machine other than IVPM2 to perform these tests.

```
*** STARTING TEST SECTION 6
```

⋮

At the end of this phase, IVPM2 displays, punches, and prints a file that contains the messages generated by IVPM1 after it disconnected. See "Interpreting the IVP Test Results" on page 5-10.

⋮

```
*** IVP PROCEDURE FINISHED
```

ENTER

- The system abnormal termination test immediately follows **on the system console**, where the OPERATOR virtual machine is logged on. This test, which consists of a forced ABEND dump of VM/SP and a subsequent warm start, is **optional but recommended**. If you do not want to conduct this test, reply **no** to the next prompt.

```
*** STARTING SYSTEM ABORT ROUTINE
*** ENTER "GO" TO CONTINUE OR "NO" TO QUIT
```

go

```
*** THIS IS THE LAST STEP OF THE IVP PROCEDURE.
*** FOLLOWING SYSTEM RESTART (WARM START), START SPOOLING DEVICES.

MSG FROM OPERATOR: MANUALLY DEPRESS CPU RESTART KEY TO ABORT SYSTEM.
```

At first level, this operation depends upon the type of console that you have. See your hardware documentation.

At second level, press the PA1 key and enter:

%cp system restart

```
DMKDMP908I SYSTEM FAILURE; CODE PSA002 PROCESSOR nn
```

Note: There is a 20-30 second pause here before the next message is issued.

```
VM/SP SYSTEM RESTART DUE TO SYSTEM FAILURE
```

⋮

```
AUTO LOGON *** OPERATOR  USERS=001 BY SYSTEM
```

You are automatically logged on to the OPERATOR virtual machine.

```
DMKCPI966I Initialization complete.
```

⋮

Single Virtual Machine Procedure

- Log on to the selected virtual machine.
- IPL CMS:

ipl cms

```
VM/SP Release 6 mm/dd/yy hh:mm:ss
```

Or your own version ID.

ENTER

- Link to and access the MAINT 193 minidisk:

link maint 193 193 rr

The IVP EXECs are located on this minidisk.

```
Enter read password:
```

password

Enter the minidisk readshare password defined in the directory.

access 193 t

- Invoke the IVP:

ivp *

⋮

Note: The IVP tests run in the single virtual machine without additional user input. See "Interpreting the IVP Test Results" on page 5-10.

Interpreting the IVP Test Results

Messages at the end of the IVP test indicate successful completion. If any errors are detected by the IVP, call IBM for software support, because an error usually indicates a serious malfunction of the generated system. The IVP procedure identifies each command being tested just before the command is executed.

Error messages are displayed in a four-line format. For example:

```
*** IVP FAILURE HAS OCCURRED ***
*** COMMAND: STATE IVPTST *
*** EXPECTED RETURN CODE 28
*** RECEIVED RETURN CODE 0
```

These messages indicate that the CMS STATE command had a return code of 0 instead of the expected 28.

All information messages that originate within the IVP are preceded by three asterisks (***).

If any command fails, the IVP procedure terminates. Follow the instructions (if any are given) to log off the virtual machine.

Chapter 6. Installing Saved Segments

This chapter contains a general discussion of saved segments and provides installation procedures for the following saved segments:

- CMSDOS
- CMSBAM
- CMSVSAM
- CMSAMS
- CMSINST
- CMSVMLIB
- CMSFILES.

Overview

Saved segments are areas of virtual storage that:

- Have been saved in real storage
- Have unique names associated with them
- Contain read-only data or reentrant code
- Can be shared by multiple users.

Saved segments provide an efficient means of fetching programs and other data by merely connecting the saved segments to a user's virtual machine.

There are two types of saved segments:

- Physical saved segments
- Logical saved segments.

Physical Saved Segments

A segment is a 64K area of virtual storage that can be allocated to a virtual machine or to CP. You can load data into an area consisting of one or more 64K segments and then save a copy of the area on DASD (real storage). This copy, called a physical saved segment, is assigned a unique name. Other virtual machines can then access the data by connecting the named physical saved segment to their address space.

The procedure for installing a physical saved segment includes the following operations:

1. Reserving the allocated DASD space with a dummy minidisk under the $SAVSYS$ user ID in the CP directory. This space must be on a CP-owned volume (listed in the SYSOWN macro in the DMKSNT file).
2. Formatting the minidisk with the Format/Allocate program and allocating the space as PERM.
3. Defining the saved segment in the DMKSNT file.
4. Assembling DMKSNT.
5. Building a new CP nucleus.
6. Loading the data to be saved into the designated area of virtual storage.
7. Saving the virtual storage area in real storage.

Defining Physical Saved Segments

The name, size, and virtual storage location of all physical saved segments are defined in the system name table (DMKSNT). DMKSNT also indicates which 64K segments within each physical saved segment are shared and which are non-shared. Shared segments contain read-only data or reentrant code — information that is shared by all virtual machines that attach the saved segment. Non-shared segments usually contain non-reentrant code — information that cannot be shared because it is unique to each virtual machine that attaches the saved segment. For information about the structure and function of DMKSNT, see the *VM/SP Planning Guide and Reference*.

The sample DMKSNT file shipped on the VM/SP Product Tape contains entries for physical saved segments named CMSAMS, CMSBAM, CMSDOS, CMSFILES, CMSINST, CMSVMLIB, CMSVSAM, and HELP. Table 6-1 on page 6-3 shows the suggested layout of these physical saved segments in virtual storage. DMKSNT also contains entries for named saved systems (systems saved in real storage that can be IPLed by name, such as CMS and GCS), and could include entries for physical saved segments or named saved systems required by optional feature program products. To determine the actual load addresses for your installation, check the entries coded in your DMKSNT file.

Note: For information about installing CMSDOS, CMSBAM, CMSVSAM, and CMSAMS at locations other than those defined in the supplied DMKSNT file, see Appendix I, "Considerations for Placement of the DOS-Related Saved Segments" on page I-1.

The location of physical saved segments should be carefully planned. The address range of a physical saved segment should be either wholly inside or wholly outside the address range of any virtual machine that might attach it. If the saved segment overlaps the Page Allocation Table located at the top of a virtual machine's address space, that virtual machine cannot attach the saved segment until the overlap is eliminated by redefining the size of the virtual machine either larger or smaller.

If you make changes to DMKSNT, you must group like macros together in the DMKSNT CSECT. That is, group all NAMESYS macros together, group all NAMENCP macros together, group all NAME3800 macros together, and group all NAMELANG macros together. If you group them incorrectly (for example, by placing a NAMENCP macro between two NAMESYS macros), the system generates an MNOTE indicating that the macros are out of sequence.

WARNING

If you make any changes to the supplied DMKSNT file, make sure that your changes do not cause overlay of other products on your DASD. Also make sure that no physical saved segment or saved system overlays another physical saved segment or saved system in storage unless they are **never** used at the same time.

To verify your changes, you can use the SNTMAP EXEC, an SNT mapping tool that resides on the MAINT 193 minidisk. The SNTMAP EXEC works with a VM/SP base file called $DASD$ CONSTS, which contains DASD-specific information, to provide two listings:

- DASD SNTMAP is a map of the locations of the NAMESYS resources in the SAVESYS areas of the DASD volumes.

- MEMORY SNTMAP is a map of all defined physical saved segments in virtual storage.

For more information about coding the macros in DMKSNT and using the SNTMAP EXEC, see the *VM/SP Planning Guide and Reference.*

Table 6-1 (Page 1 of 2). Suggested Virtual Storage Layout for Physical Saved Segments and Saved Systems. **See the latest VM/SP Memo-to-Users for any changes to this layout.**

Location	Name	Function
16384K		End of Storage
14336K	CMS	Contains the CMS nucleus code for the file system, device I/O interrupt handlers, debug, etc., and the S and Y minidisk file status tables (S-STAT and Y-STAT). E00000[1] 3584[2] 224-255[3]
13888K		Inter-Segment Gap
13568K	CMSINST	Contains EXECs and System Product Editor macros. D40000 3392 212-215
13248K	HELP	Contains HELP minidisk directory (FST). CF0000 3312 207-211
13184K	CMSDOS	Contains VSE simulation routines. CE0000 3296 206
12480K		Inter-Segment Gap
12288K	CMSBAM	Contains VSE routines. C00000 3072 192-194
11840K	CMSVSAM	Contains CMS/VSAM support. Segments 185-190 are shared; segment 191 is non-shared. B90000 2960 185-191
11264K	CMSAMS	Contains CMS/AMS support. Segments 176-181 are shared; segments 182-184 are non-shared. B00000 2816 176-184
11008K	CMSVMLIB	Contains VMLIB logical saved segment that contains VMLIB callable services library (CSL). AC0000 2752 172-175
9984K	CMSFILES	Contains DMSDAC and DMSSAC logical saved segments that contain CMS Shared File System (SFS) server code. 9C0000 2496 156-171
5120K		Inter-Segment Gap
4096K	GCS	Contains GCS supervisor code and common storage. 400000 1024 64-79

Table 6-1 (Page 2 of 2). Suggested Virtual Storage Layout for Physical Saved Segments and Saved Systems. **See the latest VM/SP Memo-to-Users for any changes to this layout.**				
Location	**Name**	**Function**		
		CMS virtual machine area.		
0K		000000	0	0-63

Notes:
[1]Hexadecimal load address
[2]Starting page number
[3]Segment numbers

Logical Saved Segments

The SEGGEN command loads and saves a physical saved segment that is composed of one or more logical saved segments. A logical saved segment has a unique name and can contain multiple objects (modules, text files, execs, callable services libraries, language repositories, and user-defined objects) or a single shared minidisk directory (file status table). You can pack as many objects as can fit into a logical saved segment, and you can pack as many logical saved segments as can fit into the physical saved segment. Virtual machines can then access these objects by connecting the named logical saved segments to their address space.

Although the name, size, and location of the physical saved segment must be defined in DMKSNT, this type of physical saved segment is essentially just a container. You cannot attach the logical saved segments by attaching the physical saved segment; you must attach each logical saved segment individually. The contents of the physical saved segment are defined in a physical segment definition file that lists the names of logical segment definition files. A logical segment definition file contains records that define the objects contained in the logical saved segment.

Storing your data in logical saved segments gives you two advantages over using physical saved segments:

- You can store more types of objects in each saved segment.
- Unless the parameters of the container (the physical saved segment) need to be changed (enlarged or moved), you can add, move, change, and delete logical saved segments without having to rebuild CP.

For more information about using the SEGGEN command and defining logical saved segments, see the *VM/SP Application Development Guide for CMS.*

Loading and Saving Data in a Saved Segment

There are three ways to load and save data in a saved segment:

- Use the SEGGEN command to load and save the data in a logical saved segment contained in a physical saved segment. A logical saved segment can contain many types of data. The installation procedures provided in this chapter for the CMSVMLIB and CMSFILES saved segments use the SEGGEN command.

  For more information about using the SEGGEN command and defining logical saved segments, see the *VM/SP Application Development Guide for CMS.*

- Use a tool that loads and saves a specific type of data in a physical saved segment:

  - The SAVEFD command loads and saves a shared minidisk directory (file status table).
  - The LANGGEN command loads and saves national language text files.
  - The DOSGEN EXEC loads and saves CMS/DOS text files.
  - The SAMGEN EXEC loads and saves simulated VSE modules.
  - The VSAMGEN EXEC loads and saves VSAM and AMS files.
  - The DCSSGEN command loads and saves EXEC files and XEDIT macros.

  For more information about using SAVEFD, see *VM/SP Administration.* For more information about using LANGGEN, see *VM/SP System Facilities for Programming.* DOSGEN, SAMGEN, VSAMGEN, and DCSSGEN are described in the next section of this chapter.

- Manually load and save the data in a physical saved segment:

  1. Load the data into virtual storage with the LOAD command or the STORE command.

  2. Assign storage protection with the SETKEY command.

  3. Save the virtual storage area as a physical saved segment with the SAVESYS command.

  For more information about the LOAD and SETKEY commands, see the *VM/SP CMS Command Reference.* For more information about the STORE command, see the *VM/SP CP General User Command Reference.* For more information about the SAVESYS command, see the *VM/SP CP System Command Reference.*

Procedures

This section contains procedures for installing the following **physical saved segments**:

- CMSDOS (page 6-7)
- CMSBAM (page 6-9)
- CMSVSAM and CMSAMS (page 6-12)
- CMSINST (page 6-16)
- CMSVMLIB (page 6-21)
- CMSFILES (page 6-23).

Installation of the HELP saved segment is part of the VM/SP base installation procedure. To install saved segments required by optional feature program products, see the program product documentation.

Physical saved segments have the following general installation requirements:

- Build your VM/SP system with enough reserve DASD space to contain the physical saved segments that you want to install.

- Make sure that the physical saved segments are defined correctly in the DMKSNT file.

Alternatively, you can install CMSDOS and CMSINST as **logical saved segments** in a new or existing physical saved segment created with the SEGGEN command:

1. Define a physical saved segment in DMKSNT that is large enough to contain the logical saved segments that you want to install. This physical saved segment must be defined on a CP-owned volume and must have a unique name.

2. Define a minidisk for the physical saved segment in the $SAVSYS$ user ID in the CP directory.

3. Format the minidisk and allocate the space as PERM.

4. Reassemble DMKSNT and rebuild CP to make the physical saved segment known to CP.

5. See the *VM/SP Application Development Guide for CMS* for information about building the physical and logical segment definition files and using the SEGGEN command to save the physical and logical segments.

Installing CMSDOS

Overview

The CMSDOS saved segment contains text files needed to create a CMS/DOS environment that simulates DOS/VSE (Disk Operating System/Virtual Storage Extended) under CMS.

You can install CMSDOS as a logical saved segment or as a physical saved segment. To install CMSDOS as a logical saved segment, see the *VM/SP Application Development Guide for CMS* for information about building the segment definition files and using the SEGGEN command.

To install CMSDOS as a physical saved segment, use the DOSGEN EXEC procedure described in this section. The CMSDOS physical saved segment consists of a single 64K segment. You invoke DOSGEN with the hexadecimal load address and the name that you want to assign to this saved segment. The load address and name must match the ones coded in DMKSNT.

DOSGEN:

- Checks the entered command for the required parameters
- Issues a SEGMENT RESERVE command to make sure that space for the segment is available
- Checks that the specified virtual address contains valid characters and that it is greater than X'20000' and less than 16M
- Looks for a read/write file mode A on which to write the CMS loader work file
- Loads all the text files needed for VSE simulation, starting at the address specified
- Assigns a storage protection key of X'D'
- Saves the segment
- Writes the load map to file mode A as LOAD MAP A5.

Notes:

1. CMSDOS must be defined at a higher address than CMSBAM, CMSVSAM, and CMSAMS.
2. Before you load and save CMSDOS, you must define your virtual machine with at least 512K free storage above the end of CMSDOS. This provides room for the loader tables, which occupy the top pages of virtual storage. After you load and save CMSDOS, the loader tables and the 512K free storage are no longer required.
3. For information about installing CMSDOS at a location other than the one defined in the supplied DMKSNT file, see Appendix I, "Considerations for Placement of the DOS-Related Saved Segments" on page I-1.

Procedure

- Log on as MAINT (if you have not already done so).
- If you changed the size of the MAINT virtual machine from the default (16M), redefine your storage:

define storage 16M

```
STORAGE=16384K
CP entered; disabled wait PSW '00020000 00000000'
```

- Set up your system:

ipl 190 clear

```
VM/SP Release 6 mm/dd/yy hh:mm:ss
```

Or your own version ID.

access (noprof

This command suppresses execution of MAINT's PROFILE EXEC.

segment reserve cmsdos

This command makes sure that space for the CMSDOS segment is available.

access 293 t
access 193 u

set emsg on

This command sets all error messages on.

- Invoke DOSGEN EXEC:

dosgen ce0000 cmsdos

This address and name must match the address and name coded in DMKSNT. This example uses the address shown in Table 6-1 on page 6-3.

```
SYSTEM SAVED
DMSWGN715I DOSGEN COMPLETE
```

- If you want to save the LOAD MAP file, rename it and copy it from file mode A to the 193 minidisk (which you accessed as U). For example:

copy load map a cmsdos segmap u

Installing CMSBAM

Overview

The CMSBAM system contains the simulated VSE modules necessary to support Sequential Access Method (SAM) data management (DTFSD), the ESERV utility program, and Virtual Storage Extended/Virtual Storage Access Method (VSE/VSAM). CMSBAM is installed as a physical saved segment (it cannot be installed as a logical saved segment) and consists of three 64K segments, as shown in Table 6-1 on page 6-3.

To install CMSBAM, use the SAMGEN EXEC procedure. Before you invoke the SAMGEN EXEC, you must define a virtual machine large enough to contain CMSBAM. To provide room for the loader tables, the size of the virtual machine should be at least 512K greater than the location at which you intend to save CMSBAM. However, do not define your storage larger than the address at which you loaded CMSDOS.

SAMGEN EXEC performs the following operations:

- Prompts you for the virtual storage location where the system will be loaded and saved.
- Prompts you for the name to be used for the saved system.
- Issues the SEGMENT RESERVE command for that system name to reserve the required virtual storage, then checks to make sure that the space was reserved. (If the SEGMENT RESERVE fails, SAMGEN stops with an error message.)
- Issues the SEGMENT RELEASE command to release the reserved space.
- Issues the FETCH command to load the simulated VSE phases from the CMSBAM DOSLIB file, which is supplied as part of VM/SP, into storage at the designated address.
- Issues the SETKEY command to assign a storage protection key of X'15'.
- Issues the SAVESYS command to save the system.

Note: For information about installing CMSBAM at a location other than the one defined in the supplied DMKSNT file, see Appendix I, "Considerations for Placement of the DOS-Related Saved Segments" on page I-1.

Procedure

- Log on as MAINT (if you have not already done so).
- Install CMSDOS (if not already installed).
- If you changed the size of the MAINT virtual machine from the default (16M), redefine your storage:

define storage 16M

```
STORAGE=16384K
CP entered;  disabled wait PSW '00020000 00000000'
```

- Set up your system:

ipl 190 clear

```
VM/SP Release 6 mm/dd/yy hh:mm:ss
```

Or your own version ID.

access (noprof

This command suppresses execution of MAINT's PROFILE EXEC.

- Invoke the SAMGEN EXEC procedure:

set dos on

This command places the CMS virtual machine in a CMS/DOS environment.

Note: If you receive the following message, you can ignore it:

```
DMSSET402W DMSLBR not in CMSBAM segment; ESERV support not available.
```

```
DMSSET1101I 100K DOS partition defined at hexadecimal location 020000
```

access 293 t
access 193 u

samgen

```
DMSSGN363R ENTER LOCATION WHERE CMSBAM WILL BE LOADED AND SAVED:
```

c00000

This address must match the address coded in DMKSNT. This example uses the address shown in Table 6-1 on page 6-3.

```
DMSSGN366R ENTER NAME OF SYSTEM TO BE SAVED:
```

cmsbam

This name must match the name coded in DMKSNT.

```
DMSSGN364I FETCHING CMSBAM...
```

⋮

```
SYSTEM SAVED
DMSSGN365I SYSTEM CMSBAM SAVED.
```

Installing CMSVSAM and CMSAMS

Overview

The CMSVSAM and CMSAMS saved segments allow you to use the Virtual Storage Access Method (VSAM) and Access Method Services (AMS) under CMS. CMSVSAM and CMSAMS can be installed only as physical saved segments. CMSVSAM consists of seven 64K segments, one of which is non-shared. CMSAMS consists of nine 64K segments, three of which are non-shared.

To install CMSVSAM and CMSAMS, use the VSAMGEN EXEC procedure. Before you can invoke the VSAMGEN EXEC, you must:

- Mount the VSAM product tape at device address 181.
- Define your virtual storage size large enough to contain the CMSDOS, CMSBAM, CMSVSAM, and CMSAMS saved segments plus at least one segment to contain the CMS control blocks and free storage used during the installation process.
- Access as file mode A a read/write minidisk that is large enough to contain all of the VSAM and AMS text decks required to generate the saved segments. The amount of space required depends on the minidisk device type, as shown below:

Device	Minidisk
3330	30 cyl
3340	70 cyl
3350	15 cyl
3375	22 cyl
3380	14 cyl
FBA	13440 blk

You can invoke the VSAMGEN EXEC to perform three distinct functions, provided certain input requirements are met:

VSAMGEN Function	Required Material
Read the VSAM product tape and create the necessary text files on file mode A.	VSAM product tape
Link edit the text files to create the CMSVSAM and/or CMSAMS DOSLIBs.	Text files
Create either the CMSVSAM or CMSAMS segments from the associated DOSLIB.	DOSLIBs

Based on the function chosen, VSAMGEN verifies that the necessary input is available. If not, you receive appropriate error messages that identify the specific problem, and processing is terminated.

Note: For information about installing CMSVSAM and CMSAMS at locations other than those defined in the supplied DMKSNT file, see Appendix I, "Considerations for Placement of the DOS-Related Saved Segments" on page I-1.

Procedure

- Log on as MAINT (if you have not already done so).
- Install CMSDOS and CMSBAM (if not already installed).
- If you changed the size of the MAINT virtual machine from the default (16M), redefine your storage:

define storage 16M

```
STORAGE=16384K
CP entered; disabled wait PSW '00020000 00000000'
```

- Set up your system:

ipl 190 clear

```
VM/SP Release 6 mm/dd/yy hh:mm:ss
```

Or your own version ID.

access (noprof

This command suppresses execution of MAINT's PROFILE EXEC.

- Access a read/write minidisk of the required size as file mode A.
- Mount the VSAM product tape on a tape drive attached to MAINT at virtual address 181. (If you do not have the proper class authority to attach a tape, the system operator might have to do it for you.)
- Invoke the VSAMGEN EXEC procedure:

set dos on

This command places the CMS virtual machine in a CMS/DOS environment.

```
DMSSET1101I 100K DOS partition defined at hexadecimal location 020000
```

segment reserve cmsvsam
segment reserve cmsams

These commands make sure that space for the CMSVSAM and CMSAMS segments is available.

access 293 t
access 193 u

vsamgen

```
SELECT ONE OF THE FOLLOWING FUNCTIONS BY ENTERING THE NUMBER:

 1. INSTALL AMS           (READ VSAM PRODUCT TAPE, BUILD DOSLIB, CREATE SEGMENT)
 2. INSTALL VSAM          (READ VSAM PRODUCT TAPE, BUILD DOSLIB, CREATE SEGMENT)
 3. INSTALL VSAM AND AMS  (READ VSAM PRODUCT TAPE, BUILD DOSLIB, CREATE SEGMENT)
 4. BUILD  AMS            (BUILD DOSLIB, CREATE SEGMENT)
 5. BUILD  VSAM           (BUILD DOSLIB, CREATE SEGMENT)
 6. BUILD  VSAM AND AMS   (BUILD DOSLIB, CREATE SEGMENT)
 7. RESTART AMS           (CREATE SEGMENT)
 8. RESTART VSAM          (CREATE SEGMENT)
 9. RESTART VSAM AND AMS  (CREATE SEGMENT)
10. QUIT                  (EXIT VSAMGEN EXECUTION)

       ENTER RESPONSE...
```

n

If this is the initial installation of VSAM, select function **1**, **2**, or **3**.

If the text files have already been created from the VSAM product tape and are currently on the accessed file mode A minidisk, select function **4**, **5**, or **6**.

If the DOSLIBs currently reside on an accessed minidisk, select function **7**, **8**, or **9**. In this case, it is not necessary to have the text files available.

Note: Messages labeled 2101I are from the linkage editor and can be ignored.

```
DMSVGN366R ENTER NAME OF CMSVSAM SYSTEM TO BE SAVED:
```

cmsvsam

This name must match the name coded in DMKSNT.

```
DMSVGN363R ENTER LOCATION WHERE CMSVSAM WILL BE LOADED AND SAVED:
```

b90000

This address must match the address coded in DMKSNT. This example uses the address shown in Table 6-1 on page 6-3.

```
DMSVGN364I FETCHING CMSVSAM ...
```

⋮

```
DMSVGN365I SYSTEM CMSVSAM SAVED.
```

⋮

```
DMSVGN366R ENTER NAME OF CMSAMS SYSTEM TO BE SAVED:
```

cmsams

This name must match the name coded in DMKSNT.

```
DMSVGN363R ENTER LOCATION WHERE CMSAMS WILL BE LOADED AND SAVED:
```

b00000

This address must match the address coded in DMKSNT. This example uses the address shown in Table 6-1 on page 6-3.

```
DMSVGN364I FETCHING CMSAMS ...
```

⋮

```
DMSVGN365I SYSTEM CMSAMS SAVED.
```

⋮

Installing CMSINST

Overview

The CMS installation saved segment (CMSINST is the default name) is designed to contain EXEC files and System Product Editor (XEDIT) macros. If a frequently-used exec or editor macro resides in this saved segment, multiple users can share the same executing copy.

There are two ways that you can install CMSINST:

- With the SEGGEN command, you can build CMSINST as a **logical saved segment** in which you can also place other objects, such as modules, text files, callable services libraries (CSLs), language repositories, and user-defined objects. For more information about using SEGGEN, see the *VM Application Development Guide for CMS.*
- With the DCSSGEN command (described below), you can build CMSINST as a **physical saved segment** that contains **only** execs and editor macros. The CMSINST physical saved segment, which is already defined for you in the sample DMKSNT file, consists of four 64K segments.

Before you enter the DCSSGEN command, you must build a loadlist file that contains a list of the execs and editor macros to be loaded into the CMSINST saved segment. **The loadlist must be a fixed-format file with a logical record length of 80.** Each record in the file must contain the file ID of one exec or editor macro or a comment. DCSSGEN processes the records sequentially.

The format of a DCSSGEN loadlist entry is:

```
fn  ft  [fm  [execname  [exectype] ] ]
```

where:

fn
: is the file name of the exec or editor macro to be loaded.

ft
: is the file type of the exec or editor macro to be loaded.

fm
: is the file mode of the exec or editor macro to be loaded. If the file mode is specified as *, DCSSGEN loads the first file in the search order that satisfies the file name and file type qualifications.

execname
: is the file name to be assigned to the loaded exec or editor macro. The default is '=', which means that the present file name is to be used.

exectype
: is the file type to be assigned to the loaded exec or editor macro. The default is '=', which means that the present file type is to be used.

The file name and file type of the exec or editor macro can each be from one to eight characters. The valid characters are A-Z, a-z, 0-9, $, #, @, +, - (hyphen), : (colon), and _ (underscore). The *execname* and *exectype* can also be from one to eight characters. However, they are not limited to the file name and file type character set. The only characters not valid within an *execname* and *exectype* are =, *,), (, and X'FF'.

To enter a comment in the loadlist, type an asterisk (*) in column one followed by the text of the comment.

For example, your loadlist entries might look like this sample:

```
* Rename RDRLIST EXEC to MAIL EXEC
RDRLIST EXEC * MAIL =
FILELIST EXEC S
SYSPROF EXEC S
PARSE XEDIT S
DISCARD EXEC S
NOTE EXEC S
PROFNOTE XEDIT S
ALL XEDIT S
```

Figure 6-1. Example of a DCSSGEN loadlist

Before you process your loadlist, you should remove the comments and unnecessary blanks from the source program to conserve storage space. The EXECUPDT command with the NOCOMMENTS option removes all comments and leading blanks. One comment line containing the execname and exectype is inserted at the beginning of the file. If the source file contains Double-Byte Character Set (DBCS) characters, also specify the ETMODE option. For more information about the EXECUPDT command, see the *VM/SP CMS Command Reference.*

When you enter the DCSSGEN command, you specify the file ID of your loadlist. You also specify the name that you want to assign to the physical saved segment. This name must match the name coded in DMKSNT. The default name is CMSINST.

You should be aware that DCSSGEN might overlay system storage. To avoid subsequent problems, you should re-IPL your virtual machine immediately after DCSSGEN finishes processing. If your CMS session abends during execution of the DCSSGEN command:

1. Re-IPL your virtual machine.
2. Access only those minidisks required to execute the DCSSGEN procedure.
3. Reenter the DCSSGEN command.

If the problems persists, you might have to change the definition of CMSINST in DMKSNT and move the saved segment to a different area within your virtual machine address space.

DCSSGEN performs the following operations:

- Processes the loadlist file, sequentially loading each exec and editor macro into storage
- Saves the storage area as a physical saved segment
- Writes a load map to file mode A as *dcssname* DCSSMAP A.

In the load map file, the records copied from your loadlist file are left-justified. The records created during the build process are indented five spaces. Comments are also copied from your loadlist file, with an asterisk (*) in column one followed by the text.

The load map file (CMSINST DCSSMAP A) for the sample loadlist described above would look like this:

```
* RENAME RDRLIST EXEC TO MAIL EXEC
RDRLIST EXEC * MAIL =
    15:41:59 10/22/85 copy of RDRLIST  EXEC     S loaded as MAIL     EXEC
    EXISBLK - 280000   FBLOCK - 280100   LENGTH - 001C40
FILELIST EXEC S
    15:41:58 10/22/85 copy of FILELIST EXEC     S loaded as FILELIST EXEC
    EXISBLK - 280020   FBLOCK - 281D40   LENGTH - 0018C8
SYSPROF EXEC S
     7:30:18 11/26/85 copy of SYSPROF  EXEC     S loaded as SYSPROF  EXEC
    EXISBLK - 280040   FBLOCK - 283608   LENGTH - 002178
PARSE XEDIT S
     8:47:55 12/18/84 copy of PARSE    XEDIT    S loaded as PARSE    XEDIT
    EXISBLK - 280060   FBLOCK - 285780   LENGTH - 0024A0
DISCARD EXEC S
    15:40:32 10/22/85 copy of DISCARD  EXEC     S loaded as DISCARD  EXEC
    EXISBLK - 280080   FBLOCK - 287C20   LENGTH - 0012B0
NOTE EXEC S
    11:03:53 10/24/85 copy of NOTE     EXEC     S loaded as NOTE     EXEC
    EXISBLK - 2800A0   FBLOCK - 288ED0   LENGTH - 005310
PROFNOTE XEDIT S
    15:41:55 10/22/85 copy of PROFNOTE XEDIT    S loaded as PROFNOTE XEDIT
    EXISBLK - 2800C0   FBLOCK - 28E1E0   LENGTH - 000980
ALL XEDIT S
    15:41:05 10/22/85 copy of ALL      XEDIT    S loaded as ALL      XEDIT
    EXISBLK - 2800E0   FBLOCK - 28EB60   LENGTH - 001298
    *** End of Source List ***
    CMSINST built at 15:56:34 on 12/02/85
```

Figure 6-2. Example of a DCSSGEN load map

Procedure

- Log on as MAINT (if you have not already done so).
- If you changed the size of the MAINT virtual machine from the default (16M), redefine your storage:

define storage 16m

```
STORAGE=16384K
CP entered; disabled wait PSW '00020000 00000000'
```

- Set up your system:

ipl 190 clear

```
VM/SP Release 6 mm/dd/yy hh:mm:ss
```

Or your own version ID.

access (noprof

This command suppresses execution of MAINT's PROFILE EXEC.

segment reserve cmsinst

This command makes sure that space for the CMSINST segment is available.

access 293 b
access 193 c

DCSSGEN MODULE is on this minidisk.

segment release cmsinst

- Build the loadlist file that contains the list of execs and XEDIT macros to be loaded into the saved segment. A sample loadlist file named CMSINST EXECLIST was loaded to the MAINT 193 minidisk (B). You can use the sample file "as is," or modify the sample file, or create a new file.
- Invoke DCSSGEN MODULE:

dcssgen ***fn ft fm*** ***segname***

fn ft fm is the file ID of the loadlist file.

segname is the name of the saved segment. This name must match the name coded in DMKSNT. If you do not specify a name, the default is CMSINST.

```
SYSTEM SAVED
```

Note: When you built your CMS nucleus, if you indicated in the DMSNGP file or in answer to the DMSINQ296R prompt that you wanted to use the CMS installation saved segment (the default is YES), then this saved segment is used each time a user IPLs CMS with virtual storage defined less than the starting address of the segment (13568K). If you previously indicated that you did not want to use the CMS installation segment but now want to use it, you must modify the DMSNGP file to indicate that the segment should be used, then assemble the modified DMSNGP and rebuild CMS.

Installing CMSVMLIB

Overview

The VMLIB callable services library (CSL) is shipped as a module that is loaded to the CMS system minidisk (MAINT 190). However, text decks and template files for VMLIB are also shipped (and loaded to MAINT 193) so you can load VMLIB into a saved segment. The members of VMLIB are defined in the VMLIB CSLCNTRL file.

This procedure shows you how to load VMLIB into a logical saved segment also named VMLIB that is defined in a physical saved segment named CMSVMLIB. CMSVMLIB is defined in the sample DMKSNT file and consists of four 64K segments, as shown in Table 6-1 on page 6-3.

To install CMSVMLIB:

1. Use the CSLGEN EXEC to build a version of the VMLIB CSL that can be loaded into a logical saved segment.
2. Use the SEGGEN EXEC to build the saved segments.

   The physical segment definition file (CMSVMLIB PSEG) and the logical segment definition file (VMLIB LSEG) are already defined for you on the MAINT 193 minidisk. For more information about physical and logical segment definition files, see the *VM/SP Application Development Guide for CMS.*

Procedure

- Log on as MAINT (if you have not already done so).
- If you changed the size of the MAINT virtual machine from the default (16M), redefine your storage:

define storage 16M

```
STORAGE=16384K
CP entered;  disabled wait PSW '00020000 00000000'
```

- Set up your system:

ipl 190 clear

```
VM/SP Release 6 mm/dd/yy hh:mm:ss
```

Or your own version ID.

access (noprof

This command suppresses execution of MAINT's PROFILE EXEC.

access 293 b
access 193 c

These minidisks contain the latest VMLIB text decks and template files, the physical segment definition file (CMSVMLIB PSEG), and the logical segment definition file (VMLIB LSEG).

- Build the CSL file:

cslgen seg vmlib from vmlib cslcntrl (to a

This command uses the VMLIB CSLCNTRL file as input to build a CSL file called VMLIB CSLSEG on file mode A. For information about the CSLGEN command, see the *VM/SP Application Development Guide for CMS.*

- Build the saved segments:

seggen cmsvmlib pseg (gen map

SEGGEN uses the CMSVMLIB PSEG file as input to generate the saved segments and produces a segment load map on file mode A. For information about the SEGGEN command, see the *VM/SP Application Development Guide for CMS.*

```
SYSTEM SAVED
```

Installing CMSFILES

Overview

The DMSDAC MODULE and DMSSAC MODULE files (located on the MAINT 193 minidisk) contain CMS Shared File System (SFS) file pool server code. If you have multiple SFS server machines on your system, you can improve system performance by loading these modules into a saved segment.

This procedure shows you how to load DMSDAC and DMSSAC into logical saved segments named DMSDAC and DMSSAC that are defined in a physical saved segment named CMSFILES. CMSFILES is defined in the sample DMKSNT file and consists of sixteen 64K segments, as shown in Table 6-1 on page 6-3.

To build the saved segments, use the SEGGEN command. The physical segment definition file (CMSFILES PSEG) and logical segment definition files (DMSDAC LSEG and DMSSAC LSEG) are already defined for you on the MAINT 193 minidisk. For more information about physical and logical segment definition files, see the *VM/SP Application Development Guide for CMS.*

Procedure

- Log on as MAINT (if you have not already done so).
- If you changed the size of the MAINT virtual machine from the default (16M), redefine your storage:

define storage 16M

```
STORAGE=16384K
CP entered; disabled wait PSW '00020000 00000000'
```

- Set up your system:

ipl 190 clear

```
VM/SP Release 6 mm/dd/yy hh:mm:ss
```

Or your own version ID.

access (noprof

This command suppresses execution of MAINT's PROFILE EXEC.

access 293 b
access 193 c

These minidisks contain the latest DMSDAC MODULE and DMSSAC MODULE files, the physical segment definition file (CMSFILES PSEG), and the logical segment definition files (DMSDAC LSEG and DMSSAC LSEG).

- Build the saved segments:

seggen cmsfiles pseg (gen map

SEGGEN uses the CMSFILES PSEG file as input to generate the saved segments and produces a segment load map on file mode A. For information about the SEGGEN command, see the *VM/SP Application Development Guide for CMS.*

```
SYSTEM SAVED
```

Chapter 7. Installing a New System National Language

Overview

VM/SP is shipped with mixed-case American English as the default national language. The product tape also contains files for uppercase English. During base installation you can install either American English (by default) or uppercase English as the system national language.

When VM/SP is completely installed, the system national language is automatically set for CP and all virtual machines on the system. A user's virtual machine is set to the system national language during logon, unless the directory entry for that virtual machine contains an OPTION LANG statement that overrides the system national language.

Note: Some messages, panels, and commands use American English regardless of the language installed as the system national language. For example, the language of the installation and system generation tools is American English and cannot be changed.

You can also order other national languages to install on your VM/SP system. VM/SP national languages (except American English and uppercase English) are distributed on national language feature tapes. The files on a national language feature tape contain translated information. Each national language supported by VM/SP is identified by a unique 1-to-5 character language identifier (*langid*). Table 7-1 shows the supported national languages and their corresponding language IDs.

Table 7-1. Language IDs for Supported National Languages

Language	***langid***
American English (mixed-case)	AMENG
Kanji (Japan)	KANJI
Uppercase English	UCENG
Brazilian Portuguese	PORTG
French	FRANC
German	GER
Spanish	ESPAN
Chinese Simplified	HANZI

There are two ways that you can install another national language on your system:

- As a replacement for the current system national language. Use the procedure described in this chapter.
- As an option available to selected system users. See *VM System Facilities for Programming*.

Installing a new system national language involves the following steps:

1. Load the national language files.

   If you are installing any national language other then AMENG or UCENG, these files are on a national language feature tape. If you are changing the system national language from AMENG to UCENG, or UCENG to AMENG, files for both national languages were loaded from the product tape during base installation. However, you must reload the HELP files from the product tape to have the correct version of these files (mixed-case or uppercase) on your system.

2. Tailor the system generation files and update the CMS translation tables (if necessary).

3. Build a new CP nucleus containing the CP language files.

4. Shut down the system and do a warm start to bring the new CP nucleus on line.

5. Build and save a new CMS nucleus containing the CMS language files.

6. (If GCS is installed) Build a new GCS configuration file (optional).

7. (If GCS is installed) Build and save a new GCS nucleus containing the GCS language files.

Contents of a National Language Feature Tape

A national language feature tape contains source files and object files. Figure 7-1 shows the layout of the tape files.

File 1:	Header
File 2:	CMS base
File 3:	CP
File 4:	HELP
File 5:	CMS source
File 6:	TSAF
File 7:	GCS
File 8:	AVS

Figure 7-1. National Language Feature Tape File Layout

The file names of national language source and object files have a one- or two-character **country code** suffix as the seventh and eighth characters to identify the national language and the language ID. The recognized values for the country code are shown in Table 7-2. These values are stored in the VMFNLS LANGLIST file.

Note: The American English versions of translatable national language files have six-character file names; they do not use the country code.

Table 7-2 (Page 1 of 2). National Language Country Codes

Language	***langid***	**Country Code (*cc*)**
Kanji (Japan)	KANJI	A
Uppercase English	UCENG	B
Brazilian Portuguese	PORTG	C

Table 7-2 (Page 2 of 2). National Language Country Codes

Language	***langid***	**Country Code (*cc*)**
French	FRANC	D
German	GER	E
Spanish	ESPAN	F
Chinese Simplified	HANZI	G

Source Files

National language source files are not in machine-readable form, but they can be updated. These source files are then converted into object files that the system can read.

File ID	**Description**
DMKMES*cc* REPOS	CP message repository, containing the translated versions of CP system messages.
DMSMES*cc* REPOS	CMS message repository, containing the translated versions of CMS system messages.
DMSSPA*cc* DLCS	CMS command syntax file, containing the translated syntax definitions for CMS commands.
DMSTRT*cc* ASSEMBLE	CMS uppercase translation table, mapping lowercase alphabetic characters to uppercase for the language.
CSIMES*cc* REPOS	GCS message repository, containing the translated versions of GCS messages.
ATSUME*cc* REPOS	TSAF message repository, containing the translated versions of TSAF messages.
AGWCMR*cc* REPOS	AVS message repository, containing the translated versions of AVS messages.
xxxxxxxx HELP*zzzz*	HELP files, where *xxxxxxxx* is the message number or the name of the command and *zzzz* is the name of the HELP facility component. HELP files have the same file IDs for all languages; therefore, the HELP files for each language installed on your system must reside on a different minidisk. Table 7-3 on page 7-4 shows the recommended minidisks for the supported national languages. **Note:** These minidisks are not defined in the base CP directory.

Table 7-3. Recommended Minidisks for National Language HELP Files

Language	Minidisk
KANJI	MAINT 401
Uppercase English	MAINT 402
Brazilian Portuguese	MAINT 403
French	MAINT 404
German	MAINT 405
Spanish	MAINT 406
Chinese Simplified (HANZI)	MAINT 407

Object Files

National language object files have been converted to machine-readable form. These files are used to build the new CP, CMS, and GCS nuclei during the installation procedure. The text file produced for the TSAF message repository is loaded into the TSAF virtual machine's storage using the SET LANGUAGE command.

File ID	**Description**
DMKMES*cc* TEXT	CP message repository, created by GENMSG.
DMSMES*cc* TEXT	CMS message repository, created by GENMSG.
DMSSPA*cc* TEXT	CMS system command syntax definition file, created by CONVERT COMMANDS.
DMSSSY*cc* TEXT	CMS system national language translation and synonym table, created by CONVERT COMMANDS.
DMSTRT*cc* TEXT	CMS uppercase translation table, created by VMFASM.
CSIMES*cc* TEXT	GCS message repository, created by GENMSG.
ATSUME*cc* TEXT	TSAF message repository, created by GENMSG.
AGWCMR*cc* TEXT	AVS message repository, created by GENMSG.

Procedure

Notes:

1. You must be logged on to the MAINT virtual machine.
2. Some languages have character sets that require special hardware. Be sure that all display devices in your configuration can properly display the character set of the new system national language.

Step 1. Load the Language Files from the Language Feature Tape (All Languages Except AMENG or UCENG)

- Mount the national language feature tape on device 181.
- Load the language files:

itask load lang ***comp***

comp identifies the language files that you want to load from the language tape. Enter **all** to load all of the files. Enter **allobj** to load only the TEXT, REPOS, and HELP*zzzz* files. If you do not have all of the VM/SP components installed in your system, enter **cp**, **cms**, **gcs**, **tsaf**, **avs**, **help**, or **cmssrc** to load the files for the specified component. Enter an ITASK LOAD command for each component that you have installed.

ITASK calls the SPLOAD EXEC to load the language files to the destinations specified in the SPLOAD PROFILE, as indicated in Table 7-4 and Table 7-5 on page 7-6.

Notes:

1. REPOS files for source are loaded to the object minidisk or SFS directory for the corresponding component.
2. You are prompted for the address of the minidisk where you want the HELP files loaded.

Table 7-4. National Language Source File Destinations

Source File	Minidisk or Directory
DMKMES*cc* REPOS	MAINT 194
DMSMES*cc* REPOS	MAINT 193
DMSSPA*cc* DLCS	MAINT 393
DMSTRT*cc* ASSEMBLE	MAINT 393
CSIMES*cc* REPOS	VMSYS:MAINT.GCS.OBJECT
ATSUME*cc* REPOS	VMSYS:MAINT.TSAF.OBJECT
AGWCMR*cc* REPOS	VMSYS:MAINT.AVS.OBJECT
xxxxxxxx HELP*zzzz*	MAINT *vdev* (unique HELP minidisk for this language)

Table 7-5. National Language Object File Destinations

Object File	Minidisk or Directory
DMKMES*cc* TEXT	MAINT 194
DMSMES*cc* TEXT	MAINT 193
DMSSPA*cc* TEXT	MAINT 193
DMSSSY*cc* TEXT	MAINT 193
DMSTRT*cc* TEXT	MAINT 193
CSIMES*cc* TEXT	VMSYS:MAINT.GCS.OBJECT
ATSUME*cc* TEXT	VMSYS:MAINT.TSAF.OBJECT
AGWCMR*cc* TEXT	VMSYS:MAINT.AVS.OBJECT

Step 1A. Load the HELP Files from the Product Tape (AMENG or UCENG Only)

- Tailor the SPLOAD PROFILE and define a new target minidisk for the HELP FILES tape file **unless you want your existing HELP files overlaid**. The recommended minidisk for each national language is listed in Table 7-3 on page 7-4.

WARNING

The recommended minidisks are not defined in the supplied CP directory. You must define the minidisk you want to use. Also make sure that you define the minidisk in the section of the SPLOAD PROFILE that matches the density of your product tape. Otherwise, you will overlay your existing HELP files.

- Mount the product tape on device 181.
- Load the HELP files:

```
spload help files
```

- (UCENG only) Convert the HELP files to uppercase:

```
access vdev fm
```

vdev is the address of the minidisk where you loaded the HELP files.

fm is any available file mode.

```
copyfile * * fm = = = (upcase olddate replace
```

fm is the file mode of the minidisk where you loaded the HELP files.

Step 2. Modify the System Generation Files and Update the CMS Translation Tables (If Necessary)

- Tailor the VM/SP product parameter file or the product parameter override file you are using to build the new system:
  - If you are building an AMENG or UCENG system, change the language specified in the national language statements (:NLS.) in the CP, CMS, and GCS areas of the base product parameter file, or define these changes in an override file. For information about creating and using override files, see the *VM/SP Service Guide.*)
  - If you are building a system for any other national language, an override file for that language is supplied on the feature tape. Tailor this file (which has the same name as the feature) if you want to change any other system generation parameters (for example, to build an AP or MP system, or to select CP options, and so on).
- Modify the CMS nucleus generation profile (DMSNGP ASSEMBLE):

```
xedit dmsngp assemble
   :
```

Change the LANGID statement to identify the new system national language.

If you loaded HELP files to a minidisk other than 19D, and you want to use those files as the system HELP files, change the HELP statement to identify the new address.

If the new language contains double-byte character set data, change the DBCS statement to YES.

```
file
```

```
access 295 a
```

vmfasm dmsngp _ppfname_ _compname_ (ppf

ppfname is the file name of the product parameter file that contains the CMS parameters you want to use:

- If you are building an AMENG or UCENG system using only the base product parameter file, enter **5664167e**.
- If you are building an AMENG or UCENG system using alternative and/or additional parameters defined in a product parameter override file, enter the name of the override file.
- If you are building a system for any other national language, enter the name of the override file supplied on the feature tape for that language.

compname is the name of the component area in the product parameter file (base file or override file) that contains the CMS parameters you want to use:

- If you are building an AMENG or UCENG system using only the base CMS area in the base file, enter **cms**.
- If you are building an AMENG or UCENG system using alternative and/or additional CMS parameters defined in an override area, either in the base file or in an override file, enter the name of the override area.
- If you are building a system for any other national language, enter the name of the CMS override area in the national language override file supplied on the feature tape for that language.

access 191 a

Updating the CMS Translation Tables

The DMSTRT*cc* ASSEMBLE file that you loaded from the language feature tape in Step 1 contains read-only system translation tables used by other CMS modules for:

- Uppercase translation
- 327x workstation support (displayable characters, APL, TEXT, and so on).

You might need to update the translation tables in this file to match the language customization generated for your 3174/3274 controllers. You should create a language saved segment or CMS named saved system for each national language that you install that requires customization of your 3174/3274 controllers.

For example, you can edit DMSTRT*cc* using the XEDIT command with the UPDATE option:

```
access 393 c
xedit dmstrtcc assemble c (update
```

where *cc* is the country code for the national language you are installing.

This option creates a file named DMSTRT*cc* UPDATE that contains your changes; the source file is not changed. For more information about using the XEDIT command with the UPDATE option, see the *VM/SP CMS Command Reference* and/or the *VM/SP System Product Editor Command and Macro Reference*.

Next, create an auxiliary control file that points to your UPDATE file, and then identify the auxiliary control file in the CMS control file (DMSSP). For more information about control files and auxiliary control files, see the *VM/SP Service Guide*.

Finally, use the VMFNLS EXEC to generate the associated object file (DMSTRT*cc* TEXT) that includes your changes:

```
vmfnls dmstrtcc assemble ppfname compname (ppf
```

For more information about the VMFNLS EXEC, see the *VM/SP Service Guide*.

Step 3. Build a New CP Nucleus Containing the CP Language Files

- Build the CP nucleus:

vmfbld *ppfname* *compname* *bldlist*

ppfname is the file name of the product parameter file that contains the CP parameters you want to use:

- If you are building an AMENG or UCENG system using only the base product parameter file, enter **5664167e**.
- If you are building an AMENG or UCENG system using alternative and/or additional parameters defined in a product parameter override file, enter the name of the override file.
- If you are building a system for any other national language, enter the name of the override file supplied on the feature tape for that language.

compname is the name of the component area in the product parameter file (base file or override file) that contains the CP parameters you want to use:

- If you are building an AMENG or UCENG system using only the base CP area in the base file, enter **cp**.
- If you are building an AMENG or UCENG system using alternative and/or additional CP parameters defined in an override area, either in the base file (such as CPAP or CPMP) or in an override file, enter the name of the override area.
- If you are building a system for any other national language, enter the name of the CP override area in the national language override file supplied on the feature tape for that language.

bldlist is the file name of the loadlist you want to use to build this nucleus, **if that loadlist is different from the one specified in the product parameter file (base file or override file) you are using**. If you do not want to change the loadlist, you can omit this operand.

⋮

```
NUCLEUS LOADED ON VMSRES --- STARTING CYL/BLK=nnnn  , LAST CYL/BLK USED=nnnn

CP entered; disabled wait PSW '00020000 00000012'
```

Notes:

1. These messages indicate that the nucleus has loaded correctly. However, on many processors the PSW message does not display automatically. If the PSW message does not display within about one minute, enter:

   #cp display psw

2. The `NUCLEUS LOADED` message gives the starting and ending cylinders/blocks for the nucleus. If you have added minidisks or changed minidisk allocations since the initial CP build, you should verify that this nucleus has not expanded into another allocated area such as TEMP, WARM START, and so on, or unpredictable results may occur.

   An easy way to check this is to look at the VMUSERS DISKMAP file that you created after you tailored the directory. Make sure that `LAST CYL/BLK USED=`*nnnn* portion of the response displayed above does not exceed the $CPNUC$ ending cylinder/block defined in the directory. To look at this file, you must first enter the following command to exit the WAIT state:

   #cp ipl 190 clear

Saving and Printing the CP Load Map

The CP load map has been spooled to MAINT's virtual printer. To save the load map on the CP object minidisk (MAINT 194), enter:

#cp spool prt nohold
#cp close prt
#cp ipl 190 clear

When `VM READ` appears in the corner of your screen, enter:

ENTER
access 194 c
query rdr

```
MAINT spoolid PRT nnnnnnnn 001 NONE STANDARD OFF
```

receive *spoolid* cpnuc map c

After you save the load map, you can print a copy of it by entering:

print cpnuc map c

#cp ipl 190 clear

This command is required only if you did not issue the command to look at the VMUSERS DISKMAP file or to save the CP load map.

Step 4. Shut Down and Do a Warm Start

Note: If you have been using two terminals for the installation, go back to the primary console (the terminal where the OPERATOR virtual machine is logged on) and do the shutdown from there. After the shutdown and warm start are complete, OPERATOR is automatically logged on to the console again, and you can log on to the second terminal as MAINT.

- ■ Shut down the system:

drain all

This command stops spooling operations on all real unit record devices. However, any jobs in progress continue processing until they are done. **Be sure to wait until you receive a** `DRAINED` **response for each device.**

⋮

shutdown

```
SYSTEM WARM START DATA SAVED
SYSTEM SHUTDOWN COMPLETE
```

Second Level Only

```
CP entered;  disabled wait PSW  '000A0000 00000008'
```

End of Second Level Only

First Level Only

- ■ IPL CLEAR the real address of your VMSRES volume according to the directions in the *Operator's Guide* for your machine.

End of First Level Only

Second Level Only

- ■ IPL your VMSRES volume:

ipl *vdev* clear

vdev is the first level virtual address of your VMSRES volume, which functions as a "real" address at second level.

End of Second Level Only

```
VM/SP Release 6, Service level nnnn; created on mm/dd/yy at hh:mm:ss
```

⋮

```
Start ((WARM|CKPT|FORCE|COLD)(DRAIN))|(SHUTDOWN):
```

warm

```
AUTO LOGON *** OPERATOR USERS=001 BY SYSTEM
```

You are now logged on to the OPERATOR virtual machine.

```
DMKCPJ952I nnnnnK system storage
```

For some processors, *nnnnn*K might be less than your actual system storage.

⋮

```
DMKCPI966I Initialization complete
```

- Log on to the MAINT virtual machine by doing **one** of the following:
  - At the primary console, enter **#cp disconn** to disconnect the OPERATOR virtual machine, then log on to the same device as MAINT.

    or

  - Log on as MAINT at your previously defined second terminal.
- IPL CMS:

ipl cms

```
VM/SP Release 6 mm/dd/yy hh:mm:ss
```

Or your own version ID.

ENTER

Step 5. Build and Save a New CMS Nucleus Containing the CMS Language Files

■ Build the CMS nucleus:

vmfbld *ppfname* *compname* *bldlist*

ppfname is the file name of the product parameter file that contains the CMS parameters you want to use:

- If you are building an AMENG or UCENG system using only the base product parameter file, enter **5664167e**.
- If you are building an AMENG or UCENG system using alternative and/or additional parameters defined in a product parameter override file, enter the name of the override file.
- If you are building a system for any other national language, enter the name of the override file supplied on the feature tape for that language.

compname is the name of the component area in the product parameter file (base file or override file) that contains the CMS parameters you want to use:

- If you are building an AMENG or UCENG system using only the base CMS area in the base file, enter **cms**.
- If you are building an AMENG or UCENG system using alternative and/or additional CMS parameters defined in an override area, either in the base or in an override file, enter the name of the override area.
- If you are building a system for any other national language, enter the name of the CMS override area in the national language override file supplied on the feature tape for that language.

bldlist is the file name of the loadlist you want to use to build this nucleus. Enter **cmsload** or the name of an alternative loadlist.

Nucleus Generation Prompts and Responses

Note: Depending on the status of your DMSNGP file, you might receive none, some, or all of the prompts shown in this section (which ends on page 7-18). A particular prompt appears **only** if the corresponding statement in DMSNGP is missing or contains no value (and the DEFNUC macro contains no default value), or if the DMSNGP statement contains a question mark (?). If you do not receive a prompt, the system is using either the value specified in DMSNGP or the default defined in DEFNUC.

```
DMSINQ606R System disk address =
```

ENTER or *vdev*

190 is the default.

```
DMSINQ615R Y-disk address =
```

ENTER or *vdev*

19E is the default.

```
DMSINQ640R HELP disk address =
```

ENTER or *vdev*

19D is the default.

```
DMSINQ764R Language id =
```

langid

langid is the identifier for the new system national language that you are installing. Text decks for this language are loaded into the CMS nucleus.

```
DMSINQ293R Is this a DBCS language?  Enter 1 (YES) or 0 (NO).
```

ENTER (or **0**) or **1**

The default is **0** (NO). Enter **1** if the language that you are installing contains double-byte character set (DBCS) data.

```
DMSINQ295R Language level id =
```

ENTER or ***levelid***

If you load multiple versions of a national language into saved segments, ***levelid*** is a single alphanumeric character (A-Z, 0-9) that must match the level ID operand of the LANGGEN command used to build the language saved segment to be used with this CMS nucleus. It must also match the third character of the language saved segment name defined in the system name table (DMKSNT). S is the default.

For more information about the LANGGEN command and the language level ID, see *VM System Facilities for Programming*.

```
DMSINQ1104R File system read/write cache buffer size =
```

ENTER or *nn*

This is the amount of read-ahead and write-behind buffering to be used for file input and file output operations. You can enter an amount from 1 to 28 (K). **12K** is the default.

```
DMSINQ296R Should the installation segment be used?  Enter 1 (YES) or 0 (NO).
```

The CMS installation saved segment is an optional saved segment into which you can place frequently-used EXEC files and System Product Editor (XEDIT) macros. Installation of this segment is optional.

ENTER (or **1**) or **0**

The default is **1** (YES). Enter **0** if you do not want to use the segment.

```
DMSINQ310R Installation segment name =
```

This prompt appears only if you accepted the default (or entered **1**) at the previous prompt.

ENTER or *segname*

Enter a one-to-eight alphanumeric character name for the CMS installation saved segment, or press ENTER to accept the default name, **CMSINST**.

```
DMSINI729R Do you want to save the system?  Enter 1 (YES) or 0 (NO).
```

ENTER

The default is **1** (YES).

```
DMSINI730R Saved system name =
```

ENTER or *systemname*

The default system name is **CMS**. This name must match the name defined in DMKSNT.

```
DMSINI607R Rewrite the nucleus?  Enter 1 (YES) or 0 (NO).
```

1

Enter **1** to write the CMS nucleus on the minidisk that you specify in your response to the next prompt.

```
DMSINI608R IPL device address =
```

ENTER

The default is the address of the CMS system minidisk, which you entered in response to prompt DMSINQ606R.

```
DMSINI609R Nucleus (CYL or BLK) address =
```

nnnnn

nnnnn is the cylinder/block location on 190 where the nucleus is written. Enter the correct location for your VMSRES DASD type:

Device	Location
3330	155 (cyl)
3340	384 (cyl)
3350	70 (cyl)
3375	108 (cyl)
3380	70 (cyl)
FBA	63744 (blk)

```
DMSINI610R Also IPL CYL/BLK 0?  Enter 1 (YES) or 0 (NO).
```

1

Enter **1** to write the initial IPL text on cylinder/block 0 of the minidisk where the CMS nucleus is written.

The initial IPL text is a bootstrap program that reads the CMS nucleus from the cylinder/block where the nucleus is written (as defined in your response to prompt DMSINI609R). The initial IPL text is always written on the same cylinder/block as the nucleus. If the initial IPL text is not also written on cylinder/block 0, you must specify the cylinder/block address of the nucleus when you issue IPL commands for this system. For more information, see the description of the IPL command in the *VM/SP CP General User Command Reference.*

```
DMSINI611R Enter version identification:
```

The version identification is displayed each time that you IPL the CMS system you are now generating.

ENTER or ***version***

You can enter up to 32 descriptive characters to identify this version and level of CMS, or you can press ENTER to accept the default version ID:

VM/SP Release 6 *mm/dd/yy hh:mm:ss*

```
DMSINI612R Enter installation heading:
```

The installation heading appears at the beginning of each output file created using this CMS nucleus.

ENTER or *heading*

You can enter up to 64 descriptive characters to define your own installation heading, or you can press ENTER to accept the default heading:

```
VM/SP CONVERSATIONAL MONITOR SYSTEM
```

End of Nucleus Generation Prompts and Responses

⋮

```
SYSTEM SAVED
VM/SP Release 6 mm/dd/yy hh:mm:ss
```

Or your own version ID.

ENTER

Your CMS system is saved. From now on, instead of IPLing the MAINT 190 system minidisk, enter **ipl cms**.

⋮

```
Ready;
```

Saving and Printing the CMS Load Map

The CMS load map has been spooled to MAINT's virtual printer. To save the load map on the CMS base minidisk (MAINT 193), enter:

#cp spool prt nohold
#cp close prt

```
PRT FILE spoolid TO MAINT COPY 0001 NOHOLD
```

access 193 r
receive ***spoolid*** **cmsnuc map r**

After you save the load map, you can print a copy of it by entering:

print cmsnuc map r

Step 6. Build a New GCS Configuration File (Optional)

If you have GCS installed, you might want to build a GCS configuration file for this language. For example, you might want to add or change authorized user IDs, or you might want to change the text of your system ID to appear in the new language.

If you do not want to build a new configuration file, go to "Step 7. Build and Save a New GCS Nucleus Containing the GCS Language Files" on page 7-22.

- Set up the build access order:

vmfsetup ***ppfname*** ***compname*** **(access bld**

ppfname is the file name of the product parameter file that contains the GCS parameters you want to use:

- If you are building an AMENG or UCENG system using only the base product parameter file, enter **5664167e**.
- If you are building an AMENG or UCENG system using alternative and/or additional parameters defined in a product parameter override file, enter the name of the override file.
- If you are building a system for any other national language, enter the name of the override file supplied on the feature tape for that language.

compname is the name of the component area in the product parameter file (base file or override file) that contains the GCS parameters you want to use:

- If you are building an AMENG or UCENG system using only the base GCS area in the base file, enter **gcs**.
- If you are building an AMENG or UCENG system using alternative and/or additional GCS parameters defined in an override area, either in the base file or in an override file, enter the name of the override area.
- If you are building a system for any other national language, enter the name of the GCS override area in the national language override file supplied on the feature tape for that language.

■ Set up the GROUP EXEC messages in storage:

Note: If you do not have a full screen display device, you cannot use the GROUP EXEC to build the configuration file. Therefore, **do not** set up the GROUP EXEC messages in storage. Use the build macros described in the *VM/SP Group Control System Command and Macro Reference* to build the configuration file manually. Then continue the language installation by modifying the GCS loadlist (see page 7-20).

copyfile csimes*cc* text b csiume = a

cc is the country code that identifies your **current** system national language, **not** the new language that you are installing. This command creates a temporary GCS message file (containing the messages for the GROUP EXEC) that has the file name required by the SET LANGUAGE command.

set language *langid* (add csi user

langid identifies your **current** system national language. This command sets *langid* as the language of the temporary GCS message file and places the file in user storage.

- Invoke the GROUP EXEC:

group *systemname*

This command assigns ***systemname*** as the name of the GCS configuration file that you are building and invokes the Primary Option Menu. ***systemname*** is optional; if specified, it must match the SYSNAME entry for this GCS system in the DMKSNT file. If you specify the ***systemname*** here, the Primary Option Menu appears with the SYSTEM NAME field filled in. If you do not specify the ***systemname*** here, you must complete the SYSTEM NAME field on the Primary Option Menu.

Note: The GROUP EXEC panels are not shown here. For guidance, refer to the procedure that you used to install your base system.

If you used the Starter System procedure, see page 3-114.

If you used the existing VM/SP system procedure, see page 4-53.

- After you complete the panels and exit from the GROUP EXEC, remove the temporary GCS message file from user storage and erase it from file mode A:

set language *langid* (delete csi user

langid identifies your **current** system national language, **not** the one you are installing.

erase csiume*cc* text a

cc identifies your **current** system national language, **not** the one you are installing.

- Modify the GCS loadlist:

copyfile gcsload exec b = = a (replace

This command writes a copy of the loadlist on MAINT 295 and replaces the loadlist used to generate the previous GCS nucleus. **Do not** modify the source file in VMSYS:MAINT.GCS.OBJECT.

xedit gcsload exec a

⋮

Locate the configuration file entry (&1 &2 &3 GCS) in the loadlist and change the file name from the default (GCS) to ***systemname***.

- ■ Rename and assemble the configuration file:

rename *systemname* group a = assemble =

systemname is the name of the configuration file.

```
vmfasm systemname ppfname compname (ppf
```

systemname is the name of the configuration file.

ppfname is the file name of the product parameter file that contains the GCS parameters you want to use:

- If you are building an AMENG or UCENG system using only the base product parameter file, enter **5664167e**.
- If you are building an AMENG or UCENG system using alternative and/or additional parameters defined in a product parameter override file, enter the name of the override file.
- If you are building a system for any other national language, enter the name of the override file supplied on the feature tape for that language.

compname is the name of the component area in the product parameter file (base file or override file) that contains the GCS parameters you want to use:

- If you are building an AMENG or UCENG system using only the base GCS area in the base file, enter **gcs**.
- If you are building an AMENG or UCENG system using alternative and/or additional GCS parameters defined in an override area, either in the base file or in an override file, enter the name of the override area.
- If you are building a system for any other national language, enter the name of the GCS override area in the national language override file supplied on the feature tape for that language.

Step 7. Build and Save a New GCS Nucleus Containing the GCS Language Files

■ Build and save GCS:

vmfbld ***ppfname*** ***compname*** **gcsload**

ppfname is the file name of the product parameter file that contains the GCS parameters you want to use:

- If you are building an AMENG or UCENG system using only the base product parameter file, enter **5664167e**.
- If you are building an AMENG or UCENG system using alternative and/or additional parameters defined in a product parameter override file, enter the name of the override file.
- If you are building a system for any other national language, enter the name of the override file supplied on the feature tape for that language.

compname is the name of the component area in the product parameter file (base file or override file) that contains the GCS parameters you want to use:

- If you are building an AMENG or UCENG system using only the base GCS area in the base file, enter **gcs**.
- If you are building an AMENG or UCENG system using alternative and/or additional GCS parameters defined in an override area, either in the base file or in an override file, enter the name of the override area.
- If you are building a system for any other national language, enter the name of the GCS override area in the national language override file supplied on the feature tape for that language.

⋮

```
SYSTEM SAVED
```

```
PRT FILE spoolid TO MAINT COPY 001 NOHOLD
```

This message indicates that the GCS load map file has been sent to MAINT's virtual reader. If you plan to save or print the GCS load map, record the *spoolid.*

```
SYSTEM RESET
CP entered;  disabled wait PSW '00020000 00000000'
```

■ IPL CMS:

ipl cms

```
VM/SP Release 6 mm/dd/yy hh:mm:ss
```

Or your own version ID.

ENTER

Saving and Printing the GCS Load Map

The GCS load map has been sent to MAINT's virtual reader. To save the load map in the GCS object directory (VMSYS:MAINT.GCS.OBJECT), enter:

```
access vmsys:maint.gcs.object e
receive spoolid gcsnuc map e
```

After you save the load map, you can print a copy of it by entering:

```
print gcsnuc map e
```

Installing the New Language in Multiple GCS Systems

If you have multiple GCS systems installed in your VM/SP system, repeat Steps 6 and 7, building a new GCS configuration file for each GCS nucleus.

Chapter 8. Tools

VM/SP provides various tools to help you do installation and system generation tasks. Tools used primarily for installation are described in this book. Tools used primarily for service are described in the *VM/SP Service Guide.* Tools used when modifying the system definition files are described in the *VM/SP Planning Guide and Reference.* Table 8-1 lists the tools that are described in each book.

Table 8-1 (Page 1 of 2). System Generation Tools

Tool	Book	Function
ASMGEND	SERVICE	Builds the VM/370 system assembler and creates the auxiliary directory.
ASM3705	INSTALL	Assembles 37xx control program source code.
CMSGEND	SERVICE	Builds CMS command modules and load libraries (LOADLIBs).
DCSSGEN	INSTALL[1]	Builds the CMS installation saved segment (CMSINST).
DIRECGEN	INSTALL	Sets up CP directory entries for optional feature program products.
DIRECT	PLANNING	Processes the control statements in a CP directory file.
DISKMAP	PLANNING	Summarizes the MDISK statements in a CP directory file.
DOSGEN	INSTALL[1]	Builds the CMSDOS physical saved segment.
EXPAND	SERVICE	Adds space to a program in object deck form.
GENSERVE	SERVICE	Builds CMS Shared File System (SFS) file pool server load modules.
GENTSAF	SERVICE	Builds the RUNTSAF module and creates a TSAF load map.
GEN3705	INSTALL	Generates an EXEC that assembles and link-edits the 37xx control program.
GROUP	INSTALL[2]	Builds a GCS configuration file.
INSTFPP	INSTALL	Installs optional feature program products.
ITASK	INSTALL	Performs various installation and system generation tasks by invoking other execs and commands.
PASSMOD	INSTALL	Changes logon passwords for user IDs in the CP directory when installing optional feature program products.
PRELOAD	SERVICE	Collects multiple text files and reformats them into a single text file.
SAMGEN	INSTALL[1]	Builds the CMSBAM physical saved segment.
SAVENCP	INSTALL	Reads a 37xx control program load module into virtual storage and extracts the control information required by CP.
SNTMAP	PLANNING	Processes the macro definitions in a system name table (SNT) file.
SPLOAD	INSTALL	Loads files from the VM/SP product tape, source feature tape, or national language feature tape.
UTILITY	SERVICE	Performs installation and service utility tasks: obtains DASD information; creates stand-alone service programs on minidisk; writes an IPLable copy of the CP nucleus on tape; prints system definition files; creates a stand-alone service utility tape.
VMFAPPLY	SERVICE	Creates and/or updates auxiliary control files for the PTFs on the service tape.

Table 8-1 (Page 2 of 2). System Generation Tools

Tool	Book	Function
VMFASM	SERVICE	Updates an ASSEMBLE source file according to entries in a control file, then assembles the updated file to produce an object file.
VMFBLD	SERVICE	Copies and renames PTF numbered text decks, applies patches, builds objects (nuclei, SFS load modules, RUNTSAF MODULE).
VMFDOS	INSTALL	Creates CMS files for VSE modules from VSE library distribution tape or SYSIN tape.
VMFHASM	SERVICE	Updates an ASSEMBLE source file according to entries in a control file, then assembles the updated file with the H-Assembler (an optional product not included with the VM/SP base) to produce an object file.
VMFLKED	SERVICE	Link-edits modules into a load library (LOADLIB).
VMFLOAD	SERVICE	Punches a nucleus or stand-alone dump load deck.
VMFMAC	SERVICE	Builds macro libraries (MACLIBs) containing macro and copy files.
VMFMERGE	SERVICE	Applies PTFs to SNA products.
VMFNLS	SERVICE	Applies updates to national language files and compiles the updated versions.
VMFOVER	SERVICE	Creates a temporary product parameter file containing parameters for a single component and applies component parameter overrides.
VMFPLC2	SERVICE	Loads files from tape, dumps files to tape, and controls various tape drive operations.
VMFREC	SERVICE	Receives program update service or corrective service from tape.
VMFREMOV	SERVICE	Removes PTFs from SNA products.
VMFSETUP	SERVICE	Generates a minidisk and SFS directory access order.
VMFTXT	SERVICE	Builds a text library (TXTLIB) containing text decks.
VMFZAP	SERVICE	Applies ZAPs to SNA products.
VRSIZE	SERVICE	Builds a DMKSLC TEXT file used to generate a virtual=real (V=R) area when building the CP nucleus.
VSAMGEN	INSTALL[1]	Builds the CMSVSAM and CMSAMS physical saved segments.
VSEVSAM	INSTALL	Loads VSE/VSAM assembler macros from tape and builds the VSEVSAM MACLIB.
ZAP	SERVICE	Modifies or dumps MODULE, LOADLIB, or TXTLIB files.
ZAPTEXT	SERVICE	Modifies or dumps individual text files.

Notes:

[1] Described in Chapter 6, "Installing Saved Segments."

[2] Described in Chapter 3, "Installing VM/SP Using the Starter System."

ASM3705 MODULE

Use the ASM3705 command to assemble 3704/3705 control program macro instructions.

Format

```
ASM3705   fn  [ (options . . .) ]

          Options:

          [XREF (FULL) ]  [RENT  ]  [DECK  ]  [LOAD  ]
          [XREF (SHORT)]  [NORENT]  [NODECK]  [NOLOAD]
          [NOXREF      ]

          [LIST  ]   [LINECOUN (55)]   [PRINT  ]
          [NOLIST]   [LINECOUN (nn)]   [DISK   ]
                                       [NOPRINT]
```

Operand

fn

specifies the file name of the source file to be assembled. The file type must be ASM3705. This source file contains 3704/3705 control program macro instructions, which must be in the form of fixed-length 80-character records.

Options

Notes:

1. All of the 3705 XF Assembler options are supported and can be used with the ASM3705 command, with the exception of ALIGN/NOALIGN and TEST/NOTEST.
2. If you specify duplicate or conflicting options, the last one you enter on the command line is the one in effect.

XREF (FULL)
includes in the assembler listing a cross-reference table of all symbols used in the assembly. This includes symbols that are defined but never referenced. The assembler listing also contains a cross-reference table of literals used in the assembly.

XREF (SHORT)
includes in the assembler listing a cross-reference table of all symbols used in the assembly. Any symbols defined but not referenced are not included in the table. The assembler listing contains a cross-reference table of literals used in the assembly.

NOXREF
does not print the cross-reference table.

RENT
checks the source file to see if it satisfies reentrancy requirements.

NORENT
suppresses the check for satisfaction of reentrancy requirements.

DECK
spools the output object module, *fn* TEXT, to the punch.

NODECK
suppresses the spooling of the output object module, *fn* TEXT, to the punch.

LOAD
creates a TEXT file for the program that was assembled.

NOLOAD
suppresses the creation of a TEXT file for the program that was assembled.

LIST
produces a LISTING file.

NOLIST
produces no LISTING file.

PRINT
spools the LISTING file to the printer.

DISK
writes the LISTING file in virtual direct access storage.

NOPRINT
produces no LISTING file.

LINECOUN nn
specifies the number of lines per output printer page. A default of 55 lines is assumed.

Files Created by the ASM3705 Command

This command creates three temporary work files and one or two permanent files.

Temporary Work Files:

fn SYSUT1
fn SYSUT2
fn SYSUT3

Any existing files with the same file identifiers are erased at the beginning of the assembly. These files are placed on the read/write minidisk with the most available space. Work space is automatically allocated as needed during the assembly and returned to available status when the assembly is complete. Insufficient space causes abnormal termination of the assembly.

Permanent Files:

fn TEXT
fn LISTING

The *fn* TEXT file contains the output object module if the LOAD option is in effect. The *fn* LISTING file contains a listing of source statements, assembled machine code, and other associated information based on the options selected. This file is created unless the NOPRINT or NOLIST options are selected. The LISTING and TEXT files are placed on:

1. The minidisk from which the source file was read,
2. Its parent, or
3. The primary minidisk, unless you created a file definition for these files placing them on a non-DASD.

Failure to get sufficient space for these files results in abnormal termination of the assembly.

Usage Notes

1. The assembly procedure can be very lengthy. The amount of time the Assembler takes depends upon the macro options you select and the number of users on the VM/SP system.

2. The LISTING file produced by the assembly is quite large. If you let the ASM3705 command option default to DISK, much of the space on your file mode A minidisk is used. Therefore, IBM recommends that you specify the PRINT option when you issue the ASM3705 command. Also, there are many macro expansions that make the LISTING file larger. IBM recommends that you insert a PRINT NOGEN assembler statement in front of the first macro instruction in the input file to suppress the printing of the macro expansions and reduce the size of the LISTING file.

3. You should examine the output of the Stage 1 assembly carefully and produce a list of resource IDs, with their characteristics, for the operations personnel. The cross-reference list for operations should include:

   - Resource ID
   - Type of resource (line or terminal)
   - Type of line (EP-mode or variable)
   - Location

GEN3705 MODULE

Use the GEN3705 command to process the text file built by the ASM3705 command. GEN3705 creates an assembler source file for each job step in the input file and then creates an EXEC procedure to assemble and link edit the source files. Command options let you determine whether or not GEN3705 includes a command in the EXEC procedure to save a copy of the load module on minidisk, or if GEN3705 invokes the EXEC procedure automatically.

Format

```
GEN3705     fn  ft [fm]   [(options ...)]

            Options:
               [RUN  ]   [SAVE  ]
               [NORUN]   [NOSAVE]
```

Operands

fn

is the file name of the text file. The file must contain fixed-length 80-character records.

ft

is the file type of the text file. The file type is normally TEXT.

fm

is the file mode of the text file.

Options

Note: If duplicate or conflicting options are specified, the last option entered on the command line is in effect.

RUN

causes the output EXEC file to be executed at the conclusion of the GEN3705 processing.

NORUN

suppresses the execution of the output EXEC file.

SAVE

includes a SAVENCP command in the output EXEC file to create a page-format copy of the 3704/3705 control program on a VM/SP CP-owned volume.

If you are generating a 3705 Emulator control program with a Type 4 channel adapter, do not use the SAVE option; an error message will result from the SAVENCP command. In this case, you must enter the SAVENCP command yourself, specifying the CAMOD option.

NOSAVE

does not include the SAVENCP command in the output EXEC file.

Files Created by the GEN3705 Command

Three types of permanent files are created when the GEN3705 command successfully executes: ASM3705, TEXT, and EXEC files.

*fname*00 ASM3705	*fname*L0 TEXT	*fname* EXEC
*fname*01 ASM3705	*fname*L0 TEXT	
. .	. .	
. .	. .	
. .	. .	
fnamenn ASM3705	*fname*L*n* TEXT	

A separate ASM3705 file is created for each assembly job step in the input text file. Each ASM3705 file created by GEN3705 is given a unique file name of the form *fnamenn*. The first six characters of the input file name are concatenated with a two-digit number. For example, if the input file is NCP320 TEXT, the output files are NCP32000 ASM3705, NCP32001 ASM3705, ..., NCP320nn ASM3705. These files are used as input to the 3705 Assembler when it is invoked by the EXEC procedure that GEN3705 creates.

The GEN3705 program creates several TEXT files. These files contain only linkage editor control statements, those statements necessary to build the load module file for the 3704/3705 control program. Each of the TEXT files created is given a unique file name of the form *fname*L*n*. The first six characters of the input file name are concatenated with the letter L and a one-digit number. For example, if the input file is NCP320 TEXT, the linkage editor output files are NCP320L0 TEXT, NCP320L1 TEXT, ..., NCP320Ln TEXT.

The file names assigned to the linkage editor and assembler files must be different. If the file names are the same, when the ASM3705 files are later assembled, TEXT files will be produced that will have file identifiers that conflict with the linkage editor files.

The EXEC macro file created contains the CMS commands necessary to invoke the ASM3705 command for each of the ASM3705 files, and to subsequently invoke the linkage editor for each of the Assembler TEXT files. If the SAVE option is specified, the EXEC file also issues the SAVENCP command, which loads the 3704/3705 control program image into virtual storage and creates the page-format copy of it on a CP-owned volume. The file name of the GEN3705 input file is used as the *ncpname* operand for the SAVENCP command.

Usage Notes

1. IBM recommends that you specify the RUN option. Using the RUN option, GEN3705 stacks a CMS command line to cause the EXEC file to execute following the completion of the GEN3705 program. This technique minimizes the virtual storage overhead during the EXEC file execution.
2. The EXEC procedure contains CMS commands that:
   - Assemble the 3705 source files (ASM3705 commands)
   - Build the TXTLIB that the 3705 Assembler needs (TXTLIB commands)
   - Define all the necessary files, such as the SYSLIB and SYSLMOD files, load libraries, and text libraries (FILEDEF commands)
   - Link edit the 3705 text files creating a load module (LKED commands).

3. If you do not specify the SAVE option, you have to explicitly enter the SAVENCP command. If you do specify the SAVE option, be sure that the input file has the same file name as the entry reserved in the system name table. The system name table is created when a NAMENCP macro is issued during a VM/SP system generation. The NAMENCP macro and the building of the system name table are described in the *VM/SP Planning Guide and Reference.*

ITASK EXEC

Use the ITASK EXEC to do installation and system generation tasks. ITASK is the primary installation tool used in the Starter System procedure. By automatically invoking other EXECs and commands, ITASK lets you complete the installation process with fewer entries and decisions.

Format

```
ITASK  { ALLOCATE
       { ASSEMBLE  { {ALLCP  }  [CP  ] }
       {           { {DMKBOX }  [CPAP] }
       {           { {DMKFCB }  [CPMP] }
       {           { {DMKRIO }         }
       {           { {DMKSNT }         }
       {           { {DMKSYS }         }
       {           {                   }
       {           {  DMSNGP   [CMS]   }
       { BASEIDS
       { BUILD     { CMS                      }
       {           { {CP  }                   }
       {           { {CPAP} [NOASSEM]         }
       {           { {CPMP}                   }
       {           { GCS   [systemname]       }
       {           {       [GCS       ]       }
       { FILESERV  {VMSYS }
       {           {VMSYSU}
       { LOAD      { AVS                      }
       {           { CMS                      }
       {           { CMSFTSRC                 }
       {           { CMSSRC                   }
       {           { CP                       }
       {           { CPSRC                    }
       {           { GCS                      }
       {           { GCSSRC                   }
       {           { HELP                     }
       {           { HPO                      }
       {           { IPCS                     }
       {           { IPCSSRC                  }
       {           { LANG {ALL    }           }
       {           {      {ALLOBJ }           }
       {           {      {AVS    }           }
       {           {      {CMS    }           }
       {           {      {CMSSRC }           }
       {           {      {CP     }           }
       {           {      {GCS    }           }
       {           {      {HELP   }           }
       {           {      {TSAF   }           }
       {           { TSAF                     }
       {           { TSAFSRC                  }
```

Operands

ALLOCATE

allocates space on the DASD volumes identified in the $ALLOC$ user ID in the CP directory.

You are prompted for the real address of each DASD volume, and you have the option to skip any volume that you do not want to allocate. Allocation is done according to the entries in the CP directory; all space not specifically allocated is allocated as PERM.

ASSEMBLE

calls the VMFASM EXEC to assemble the specified system definition file(s).

When assembling any of these files, you can enter an additional operand to specify the name of the area in the product parameter file that contains the name of the control file.

When assembling CP files, the operand also refers to the type of system being built. You can enter CP (for a uniprocessor system), CPAP (for an attached processor system), or CPMP (for a multiprocessor system). If you do not specify the type of system, CP (uniprocessor) is the default.

When assembling the CMS file, CMS is the default.

ALLCP

assembles DMKBOX, DMKFCB, DMKRIO, DMKSNT, and DMKSYS.

DMKBOX

assembles only DMKBOX.

DMKFCB

assembles only DMKFCB.

DMKRIO

assembles only DMKRIO

DMKSNT

assembles only DMKSNT.

DMKSYS

assembles only DMKSYS.

DMSNGP

assembles only DMSNGP.

BASEIDS

issues the CMS FORMAT command to format the remaining minidisks defined in the base CP directory that did not have code loaded to them during the installation process or are not associated with any particular base component.

The minidisks formatted during this procedure are: AUTOLOG1 191; AVSVM 191; CMSBATCH 195; CMSUSER 191; EREP 191; MAINT 201, 319, and 595; OPERATNS 191 and 193; VMSERVS 191; and VMSERVU 191.

BUILD

builds the specified nucleus.

CMS

calls the VMFASM EXEC to assemble the CMS nucleus generation profile (DMSNGP ASSEMBLE) and then calls the VMFBLD EXEC to build the CMS nucleus.

The CMS nucleus is saved only if specified in the DMSNGP file or in response to the DMSINI729R prompt during the nucleus build.

CP
accesses the MAINT 295 minidisk for the system definition files; calls the VMFASM EXEC to assemble the DMKBOX, DMKFCB, DMKSNT, DMKSYS, and DMKRIO files; and calls the VMFBLD EXEC to build the uniprocessor (UP) CP nucleus.

CPAP
accesses the MAINT 295 minidisk for the system definition files; calls the VMFASM EXEC to assemble the DMKBOX, DMKFCB, DMKSNT, DMKSYS, and DMKRIO files; and calls the VMFBLD EXEC to build the attached processor (AP) CP nucleus.

CPMP
accesses the MAINT 295 minidisk for the system definition files; calls the VMFASM EXEC to assemble the DMKBOX, DMKFCB, DMKSNT, DMKSYS, and DMKRIO files; and calls the VMFBLD EXEC to build the multiprocessor (MP) CP nucleus.

NOASSEM
does not assemble DMKBOX, DMKFCB, DMKSNT, DMKSYS, and DMKRIO before building the CP, CPAP, or CPMP nucleus.

GCS
modifies a copy of the GCS build list (GCSLOAD EXEC) and changes the default configuration file entry (GCS) to match the file name of the GCS configuration file; writes the modified build list on the MAINT 295 minidisk; renames the file type of the configuration file from GROUP to ASSEMBLE; calls the VMFASM EXEC to assemble the configuration file; and calls the VMFBLD EXEC to build and save the GCS nucleus.

systemname
is the file name of the GCS configuration file. This name must match the SYSNAME entry for this GCS system in the DMKSNT file. If you do not specify a name, the default is GCS.

FILESERV
builds the system file pool (VMSYS) or the user file pool (VMSYSU) and starts the corresponding server machine (VMSERVS or VMSERVU) by executing the following sequence:

1. Accesses the MAINT 295 minidisk and obtains the logon password for the server and the write password for the AUTOLOG1 191 minidisk from the VMUSERS DIRECT file.

2. Links to the server's 191 minidisk and copies the (VMSYS or VMSYSU) POOLDEF file, the (VMSERVS or VMSERVU) DMSPARMS file, and the SERVER PROFILE from MAINT 295.

3. Creates a tool called RUNSERVE EXEC on the server's 191 minidisk, then releases and detaches the server's 191 minidisk.

4. Issues the AUTOLOG command to log the server on in a disconnected state.

   Note: In the IBM-supplied CP directory, MAINT is designated as the secondary user on the CONSOLE statement in the entry for the server machine. The Single Console Image Facility lets the secondary user (MAINT) become the controlling virtual machine when the primary user (the server) is disconnected. MAINT receives all console output from the server, and MAINT can issue commands on the server by using the CP

SEND command. For more information about the Single Console Image Facility, see *VM/SP System Facilities for Programming*.

5. (On the server) Accesses the MAINT 293 and MAINT 193 minidisks.
6. (On the server) Invokes the RUNSERVE EXEC, which issues the FILESERV GENERATE command to define and initialize the file pool using parameters supplied by the POOLDEF file.

   If the return code from FILESERV GENERATE is zero, RUNSERVE creates a file called FILESERV VALID and sends it to MAINT's reader. If the return code is not zero, RUNSERVE creates a file called FILESERV ERROR and sends it to MAINT's reader. Before proceeding, ITASK looks for a FILESERV file in its reader, then looks at the file type of the file. If the file type is VALID, processing continues. If the file type is ERROR, an error message is issued and processing stops.
7. (On the server) Renames SERVER PROFILE to PROFILE EXEC.
8. (On the server) Issues the FILESERV START command to start the server using start-up parameters supplied by the DMSPARMS file.
9. Links to the AUTOLOG1 191 minidisk and creates a PROFILE EXEC file that includes an AUTOLOG command for the server, then releases and detaches the AUTOLOG1 191 minidisk.
10. (VMSYS only) Issues the SET FILEPOOL VMSYS command to designate VMSYS as the default file pool for any subsequent file pool commands that MAINT issues.
11. (VMSYS only) Issues the ENROLL USER MAINT command to enroll MAINT in the VMSYS file pool.
12. (VMSYS only) Copies the SFS sample files from the MAINT 193 minidisk into the VMSYS:MAINT.SAMPLES directory.
13. (VMSYS only) Issues the ENROLL PUBLIC command to let all users on this processor connect to the file pool.
14. (VMSYS only) Issues GRANT AUTHORITY commands to grant PUBLIC read authority on the SAMPLES directory and on the individual sample files. This authority gives read access to all users on this processor.

LOAD
calls the SPLOAD EXEC to load files from the product tape, the source feature tape, or a national language feature tape to the destinations specified in the SPLOAD PROFILE. Some LOAD operands perform additional operations.

AVS
loads the AVS DELTA, AVS APPLY, AVS INTERFACE and AVS OBJECT tape files from the product tape; then calls the COPYFILE (REPLACE command to copy the AGW LOADLIB and PROFILE GCS files from the VMSYS:MAINT.AVS.OBJECT directory to the AVSVM 191 minidisk.

CMS
loads the CMS BASE, REXX DELTA, REXX APPLY, REXX OBJECT, CMS DELTA, CMS APPLY, and CMS SYSTEM tape files from the product tape; calls the ASMGEND EXEC to build the system assembler; calls the VMFASM EXEC to assemble the CMS nucleus generation profile (DMSNGP ASSEMBLE); and calls the VMFBLD EXEC to build the CMS nucleus.

The CMS nucleus is saved only if specified in the DMSNGP file or in response to the DMSINI729R prompt during the nucleus build.

CMSFTSRC
loads the CMSFT SOURCE tape file from the source feature tape.

CMSSRC
loads the CMS SOURCE tape file from the product tape.

CP
loads the SYSGEN TOOLS, SYSTEM TOOLS, SYSTEM SAMPLES, DASDMODL SAMPLES, DASDTYPE SAMPLES, CP DELTA, CP APPLY, and CP OBJECT tape files from the product tape; then calls the COPYFILE command to create copies (with new names) of the following sample files:

Old Name	New Name
DIRECT *devtype[-mm]*	VMUSERS DIRECT
DMKSNT *devtype[-mm]*	DMKSNT ASSEMBLE
DMKRIO *devtype*	DMKRIO ASSEMBLE
DMKSYS *devtype*	DMKSYS ASSEMBLE
DMSNGP SAMPLE	DMSNGP ASSEMBLE

CPSRC
loads the CP SOURCE tape file from the product tape.

GCS
loads the GCS DELTA, GCS APPLY, GCS INTERFACE, and GCS OBJECT tape files from the product tape.

GCSSRC
loads the GCS SOURCE tape file from the source feature tape.

HELP
loads the HELP FILES tape file from the product tape; asks you if you want to have the files converted to uppercase (from mixed case, as loaded); and issues the SAVEFD command to initialize and save the HELP minidisk directory (file status table) in a physical saved segment named HELP.

HPO
invokes the HPOINST EXEC. For more information about this tool, see the *VM/SP HPO Installation Guide*.

IPCS
loads the IPCS DELTA, IPCS APPLY, and IPCS OBJECT tape files from the product tape.

IPCSSRC
loads the IPCS SOURCE tape file from the product tape.

LANG
loads the specified national language file(s) from a national language feature tape.

ALL
loads the entire contents of the national language feature tape.

ALLOBJ
loads only the object (non-source) tape files: CP OBJECT, CMS BASE, GCS OBJECT, TSAF OBJECT, AVS OBJECT, and HELP FILES.

AVS
loads only the AVS OBJECT tape file.

CMS
loads only the CMS BASE tape file.

CMSSRC
loads only the CMS SOURCE tape file.

CP
loads only the CP OBJECT tape file.

GCS
loads only the GCS OBJECT tape file.

HELP
loads only the HELP FILES tape file.

TSAF
loads only the TSAF OBJECT tape file.

TSAF
loads the TSAF DELTA, TSAF APPLY, TSAF INTERFACE, and TSAF OBJECT tape files from the product tape.

TSAFSRC
loads the TSAF SOURCE tape file from the source feature tape.

Messages

DMSWTK002E	File *fn ft [fm]* not found
DMSWTK003E	Invalid option: *option*
DMSWTK008E	Device *vdev* invalid or nonexistent
DMSWTK050E	Parameter missing after *function*
DMSWTK070E	Invalid parameter *parameter*
DMSWTK095E	Invalid address *vstor*
DMSWTK360E	Invalid response *response*
DMSWTK961E	There are class *class* files in the *device*
DMSWTK965I	You may wish to tailor the following files at this time:

VMUSERS	DIRECT
DMKSNT	ASSEMBLE
DMKRIO	ASSEMBLE
DMKSYS	ASSEMBLE
DMKBOX	ASSEMBLE
DMKFCB	ASSEMBLE
DMSNGP	ASSEMBLE
VMSERVS	DMSPARMS
VMSERVU	DMSPARMS
VMSYS	POOLDEF
VMSYSU	POOLDEF

DMSWTK966R	Do you wish to have the HELP files converted to uppercase? Type: (No) or Yes
DMSWTK968I	The following minidisks defined in the base CP directory will be formatted:

AUTOLOG1 191
AVSVM 191
CMSBATCH 195
CMSUSER 191
EREP 191
MAINT 201, 319
MAINT 595
OPERATNS 191, 193
VMSERVS 191
VMSERVU 191

DMSWTK970I Formatting *userid vdev* minidisk
DMSWTK981R What is the real address of your *volume* volume? Type: real address or SKIP
DMSWTK983E Violation of CMS naming convention found in *args*

For a complete explanation of each message, see *VM/SP System Messages and Codes.*

SAVENCP MODULE

Use the CMS SAVENCP command to read a 3704/3705 control program load module created by the LKED command, and to load it into virtual storage in the CMS user area. Once the load is done, SAVENCP scans the control program image and extracts the control information required by CP. The control information is accumulated in one or more 4096-byte pages in the CMS user area. When all of the necessary control information is extracted, SAVENCP builds the Communications Controllers Parameter List (CCPARM) and issues the DIAGNOSE X'50' instruction to create the page-format copy of the control program on a CP-owned volume.

Format

```
SAVENCP  | fn   [(options ...)]
         |
         | Options:
         |
         | [ENTRY symbol ]      [NAME  ncpname ]
         | [CXFINIT      ]      [fn            ]
         |
         | [LIBE libraryname]   [CAMOD {0} ]
         | [fn              ]   [      {1} ]
```

Operand

fn
is the file name of the LOADLIB file where the 3704/3705 control program load module resides, unless the LIBE option is specified, in which case this is the member name of the image within the LOADLIB. This name is used as the *ncpname* for the DIAGNOSE instruction, unless the NAME option is also specified.

Options

ENTRY *symbol*
is the external symbol of the entry point in the 3704/3705 control program load module. (The standard entry for the Emulation Program is CYASTART.) If the SAVE option of the GEN3705 command is specified, this symbol is set in the output EXEC file according to the GEN3705 input file.

NAME *ncpname*
is the *ncpname* to be used when the DIAGNOSE parameter list is built. The *ncpname* specified must match an entry in the system name table. These entries are created with the NAMENCP macro when VM/SP is generated.

LIBE *libraryname*
is the file name of a load module library file, file type LOADLIB, which contains the control program image as member *fname*.

CAMOD
must be specified if a Type 4 Channel Adapter is being used. VM/SP supports only one Type 4 Channel Adapter at a time, although two may be present.

CAMOD 0 corresponds to -0 following the subchannel address on the ADDRESS operand of the LINE macro in Stage 1 of the EP system generation.

(0 may have been coded or defaulted on the LINE macro; you must specify it on the CAMOD option.)

CAMOD 1 corresponds to -1 following the subchannel address on the ADDRESS operand of the LINE macro in Stage 1 of the EP system generation.

How the SAVENCP Program Works

The DIAGNOSE X'50' instruction invokes the CP module DMKSNC to:

- Interpret the parameter list (CCPARM) built by SAVENCP
- Check the parameter specifications against the NAMENCP macro for the 3704/3705 control program
- Write the page-format image of the control program onto the appropriate CP-owned volume.

The parameter list for the DIAGNOSE instruction must start on a 4096-byte boundary.

When the DIAGNOSE X'50' instruction is executed, the module DMKSNC searches the DMKSNT module for a NAMENCP macro with the same *ncpname* as the one in the CCPARM parameter list. The values specified in the parameter list are compared to those specified in the NAMENCP macro. If any parameters conflict, an error message is displayed at the terminal. If no error conditions are detected, DMKSNC starts to transfer the control program image from CMS virtual storage to the CP-owned volume specified in the NAMENCP macro. Successful completion of this process completes the generation of a 3704/3705 control program for VM/SP use.

SPLOAD EXEC

Use the SPLOAD EXEC to load files from the VM/SP Product Tape, source feature tape, or national language feature tape according to the load instructions contained in the SPLOAD PROFILE file.

Format

```
SPLOAD    group    element    [fn  [ft]]
                              [*   [*] ]
                               -    -
```

Operands

group element
is a tape file identifier that the SPLOAD EXEC uses to locate the entry in the SPLOAD PROFILE that contains the load instructions for this tape file.

fn [ft]
is the file specification template. By default, this template is set to asterisks (* *), meaning all files within a tape file. However, you can specify a file name and/or file type to selectively load specific CMS files from a tape file.

SPLOAD PROFILE Syntax

Note: For a listing of the sample SPLOAD PROFILE, see Appendix C, "Sample SPLOAD PROFILE" on page C-1.

In the SPLOAD PROFILE, data entries are organized by format number (see the definition below), and format number sets are separated by comment lines. A comment line begins with a slash and an asterisk (/*). Each data entry in the SPLOAD PROFILE has the following syntax:

```
group   element   userid   address   format   vol   file
```

group element
is a tape file identifier that identifies the entry in the profile that contains the load instructions for this tape file.

userid
is the owner of the minidisk or the name of the file pool to which the tape file contents are loaded.

address
is the address of the minidisk or the name of the directory to which the tape file is loaded.

format
is the tape format. The SPLOAD PROFILE contains a complete load table for each tape format, describing the locations of the files on the product tape, source feature tape, and national language feature tape. The SPLOAD EXEC selects the appropriate entries in the PROFILE depending on the user's tape format.

vol
is the tape volume on which the tape file resides.

file
is the relative tape file number (location) of the tape file on the tape volume.

Usage Notes

1. SPLOAD EXEC rewinds the tape each time it is invoked to make sure that the proper tape is mounted and to position the tape to the proper tape file.

   The first tape file on each product tape is a header file called $TAPE$ HEADER. This file contains information that the SPLOAD EXEC uses to determine which tape is mounted, and therefore which entries to use from the SPLOAD PROFILE. The SPLOAD EXEC loads the $TAPE$ HEADER file to the user's A file mode each time the EXEC is invoked. If the user does not have a read/write A file mode, SPLOAD terminates with a return code of 36.

2. The SPLOAD EXEC expects that the tape drive being used is attached at virtual address 181. If this condition is not met, SPLOAD cannot continue.

3. SPLOAD EXEC accesses the target minidisk or directory as file mode Z.

4. If a minidisk is already accessed as the target address when SPLOAD is invoked, SPLOAD assumes that it is the correct minidisk and attempts the load, regardless of the ownership of the minidisk.

5. If a directory already exists with the specified name, SPLOAD assumes that it is the correct directory, regardless of the name of the current file pool.

6. If SPLOAD attempts to access a directory and receives a return code of 28, SPLOAD issues the CREATE DIRECTORY command to create the directory.

7. If a target minidisk is not linked, SPLOAD attempts the link. However, if a password is required, SPLOAD terminates.

8. If a minidisk to be loaded is already linked in read/write mode at a virtual address different from the address specified in SPLOAD PROFILE, the load is not performed, and SPLOAD EXEC terminates with a return code of 102.

9. If SPLOAD EXEC links to a minidisk, then accesses the minidisk and receives a return code of 100 from the access, an attempt is made to format the disk.

   The disk to which the HELP FILES tape file is loaded is formatted in 2K block size; all other disks are formatted using the block size to which the FORMAT command defaults, which depends on the DASD type. For more information about the block size defaults of the FORMAT command, refer to the *VM/SP CMS Command Reference*.

   SPLOAD provides a six-character minidisk label to FORMAT. This label consists of the first three characters of the user ID that owns the minidisk (except MAINT, which uses the characters MNT) followed by the three-digit minidisk address.

10. SPLOAD EXEC restores the access order to its original state just before exiting. However, because SPLOAD performs only standard accesses, the access that is restored for a given minidisk might be different from the original access if that minidisk was originally accessed with a subset definition.

Messages

DMSWTK002E	*fn ft* not found
DMSWSL032E	Invalid filetype *ft*
DMSWSL050E	Parameter missing after *value*
DMSWTK070E	Invalid parameter *parameter*
DMSWSG095E	Invalid address *vstor*
DMSWSL252E	Invalid filename *fn*
DMSWSL409I	Loading *fn ft* to *userid vdev*
DMSWSL737R	Enter the minidisk address for the *group element*
DMSWSL963E	*keyword value* not found in *fn ft fm*
DMSWSL964R	Wrong tape mounted; tape mounted is not a product tape Press ENTER when the correct tape is mounted or type QUIT
DMSWSL964R	Wrong tape mounted; mount product tape VOL*n* Press ENTER when the correct tape is mounted or type QUIT
DMSWUT986I	Unable to restore ACCESS to *vdev*

For a complete explanation of each message, see *VM/SP System Messages and Codes.*

VMFDOS MODULE

Use the VMFDOS command to create CMS files containing VSE modules. The VMFDOS command is intended for specific installation purposes only. VMFDOS uses either a VSE distribution library tape or VSE SYSIN tape to install only DOS/VS RPG II and VSE/VSAM licensed programs.

Note: This command was designed to work with files stored on minidisks and might not function completely with files stored in SFS directories.

Format

```
VMFDOS   [LOAD]  [ [181]  ]  [PRIVate]  [fn]  [(options ...) ]
         [SCAN]  [ [182]  ]  [SYSTem ]
                 [        ]  [SYSIN  ]
                 [ [TAP1] ]
                 [ [TAP2] ]

         PRIV or SYST options:

                 [ *           ]
            CSL  { xxx*        }
                 [ module name ]

                 [ *           ]
            RL   { yyy*        }
                 [ module name ]

                 [ *           ]
            SL   { zzz*        }
                 [ module name ]

         SYSIN options:

              { ALL    }
              { SELECT }
```

Operands

LOAD
: loads the specified VSE modules as CMS files from the VSE tape. The modules to be loaded are specified by using:

- The SL or RL option
- A scan file (see usage note 7).

LOAD is the default function.

SCAN
: specifies that the tape is to be scanned and a file created that contains the names of all the VSE modules. This file is named *fn* SCAN A1 if *fn* is specified with the VMFDOS command; otherwise, the file is named VMFDOS SCAN A1. Any options specified are ignored.

181
182
TAP1
TAP2

specifies the device address of the tape (181 or 182) or the symbolic tape identification (TAP1 or TAP2). 181 or TAP1 is the default.

PRIVate

specifies that the VSE private libraries are to be processed. This is the default parameter.

SYSTem

specifies that the VSE system libraries are to be processed.

SYSIN

specifies that the VSE tape is in SYSIN format.

fn

is the name of a CMS file to be used with either the LOAD or SCAN functions. The file type must be SCAN. The LOAD function reads the file to determine which VSE modules are to be loaded as CMS files. The SCAN function uses the file name to create a file called *fn* SCAN that lists the names of all the VSE modules contained on the tape.

Options

The following options can be used only with the PRIV or SYST operands:

CSL

indicates that the VSE modules are to be read in and decompressed. When specified without an option, this operand is valid only if a scan file name has also been specified. All of the macros listed in the scan file are loaded and decompressed.

Note: The CSL option is only supported for use on the VSE/VSAM Optional Source Statement Library.

*

specifies that all the macros on the VSE/VSAM Optional Source Statement Library tape are to be read in and decompressed into CMS files.

*xxx**

specifies that all VSE modules beginning with the characters *xxx* are to be processed.

module name

is a specific module name.

RL

*

specifies that all the VSE modules in the relocatable library are to be processed. This is the default if no options are specified, if *fn* is not specified, or if VMFDOS SCAN does not exist.

*yyy**

specifies that all VSE modules beginning with the characters *yyy* are to be processed.

module name

is a specific module name.

SL

*
: specifies that all the VSE modules in the source statement library are to be processed.

*zzz**
: specifies that all VSE modules beginning with the characters *zzz* are to be processed.

module name
: is a specific module name.

The following options can be used only with the SYSIN operand:

ALL
: specifies that all VSE modules found are to be loaded as CMS files.

SELect
: indicates that you will be prompted prior to writing each VSE module to disk. You must indicate whether or not each module is to be included.

Usage Notes

1. When the scan function is invoked, VMFDOS erases either the *fn* SCAN A1 file or the VMFDOS SCAN A1 file prior to scanning the tape.
2. The tape must be positioned at the file to be processed (using the TAPE FSF command) prior to issuing the VMFDOS command.
3. VMFDOS erases those existing files that have the same file name as a VSE module being loaded.
4. VMFDOS writes all files to file mode A. A message is issued if a R/W file mode A is not available.
5. VSE modules loaded from a relocatable library tape file will have a file type of TEXT when written as a CMS file.
6. VSE modules loaded from a source statement library tape file will have a file type of MACRO when written as a CMS file.
7. If a file name is specified when using the LOAD function, any options specified are ignored and the *fn* SCAN file is used. If a file name is not specified, but a VMFDOS SCAN file exists, any options specified are ignored and the VMFDOS SCAN file is used.
8. VMFDOS can only handle 430 entry requests at a time when using a *fn* SCAN or VMFDOS SCAN file as input. You can load more than 430 modules by using generic names in the scan file or separate scan files.

Examples

The following are examples of invoking the VMFDOS command.

```
vmfdos (rl abc*
```

All VSE modules in the private relocatable library that begin with ABC are loaded as CMS files. The CMS files created have a file type of TEXT; they have a file name that is the same as the VSE module name.

```
vmfdos syst (rl abc*
```

All VSE modules in the system relocatable library that begin with ABC are loaded as CMS files. The CMS files created have a file type of TEXT; they have a file name that is the same as the VSE module name.

vmfdos (sl xyz1*

All VSE modules in the private source statement library and all the private relocatable library modules that begin with XYZ1 are loaded as CMS files. The CMS files created have a file type of MACRO; they have a file name that is the same as the VSE module name.

vmfdos syst (sl *

All VSE modules in the system source statement library and the system relocatable library are loaded as CMS files. The CMS files created have a file type of MACRO; they have a file name that is the same as the VSE module name.

vmfdos priv (rl abc* sl *

All VSE modules that begin with ABC in the private relocatable library and all modules in the private source library are loaded as CMS files. The CMS files have a file name the same as the VSE module name and a file type of TEXT or MACRO, as appropriate.

vmfdos scan syst

The VSE system libraries are scanned. The VMFDOS SCAN file is created to contain the names of the VSE modules.

vmfdos scan sysin fn

The VSE SYSIN tape is scanned. The *fn* SCAN file is created to contain the names of the VSE modules.

vmfdos

All VSE modules in the private relocatable libraries are loaded as CMS files. The CMS files created have a file name the same as the VSE module name and a file type of TEXT.

vmfdos sysin fn

The *fn* SCAN file contains a list of module names. The modules are on the VSE SYSIN tape and are loaded as CMS files. The CMS files created have a file name the same as the VSE module name and a file type of TEXT or MACRO, as appropriate.

vmfdos sysin

The VMFDOS SCAN file contains a list of module names. These modules are on the VSE SYSIN tape and are loaded as CMS files. The CMS files created have a file name the same as the VSE module name and a file type of TEXT or MACRO, as appropriate.

Messages

DMSDOS787E   Error decompressing *macro* MACRO

For a complete explanation of this message, see *VM/SP System Messages and Codes*.

VSEVSAM EXEC

Use the VSEVSAM EXEC to obtain VSE/VSAM Assembler Language macros from the Licensed Optional Machine Readable Materials tape. The VSEVSAM EXEC creates the VSEVSAM MACLIB for you. When the MACLIB is created, it contains all of the VSE/VSAM assembler language macros and the following VSE macros: CDLOAD, CLOSE, CLOSER, GET, OPEN, OPENR, and PUT.

Format

VSEVSAM	

Example

Before invoking the VSEVSAM EXEC, complete the following:

1. Mount the Licensed Optional Machine Readable Materials tape at virtual address 181.
2. Load the seven VSE macros from the Product Tape to MAINT 393 or a minidisk of your choice. (As long as the macros are available when VSEVSAM is invoked, the actual minidisk used is not critical.)

   Note: Because the seven VSE macros (CDLOAD, CLOSE, CLOSER, GET, OPEN, OPENR, and PUT) will be loaded into the MACLIB, they can be erased from the minidisk after the MACLIB is created.

To invoke VSEVSAM EXEC, enter:

vsevsam

VSEVSAM EXEC prompts you for information. For example, the system responds:

```
DMSWVV797I QUIT may be entered in response to any query to
           end processing

DMSWVV788R Are the macros to be read from tape or are
           they already on disk?  Reply TAPE or DISK.
           (Press the ENTER key for a default of TAPE.)
```

For this example we will read the macros from minidisk. Enter:

disk

The system responds:

```
DMSWVV790R If the default library name of VSEVSAM is to be used,
           press the ENTER key; else, enter the name to be
           used for the library
```

We want to call the library VSAMMACS, so enter:

vsammacs

The system responds:

```
DMSWVV791I The library name will be VSAMMACS.  If the name is
           correct, press the ENTER key; else, enter the name
           to be used for the library, or enter QUIT to exit.
```

Because VSAMMACS is the name we want to call the library, press:

ENTER

The system responds:

```
DMSWVV808R Macro library VSAMMACS will be erased.  Press ENTER
           to continue or type QUIT to exit.
```

Because we want to erase old versions of the library that might be on file mode A, press:

ENTER

The system responds:

```
DMSWVV793I MACLIB generation completed

DMSWVV792R Are the macros to be erased from disk?  Reply YES
           or NO (press the ENTER key for default of YES).
```

We want to erase the macros from minidisk, so press:

ENTER

The system responds:

```
DMSWVV802I Macros erased; VSEVSAM processing complete
Ready;
```

Messages

DMSWVV788R	Are the macros to be read from tape or are they already on disk? Reply TAPE or DISK. (Press the ENTER key for a default of TAPE.)
DMSWVV789W	Invalid response.
DMSWVV790R	If the default library name of VSEVSAM is to be used, press the ENTER key; else, enter the name to be used for the library
DMSWVV791I	The library name will be *libname*. If the name is correct, press the ENTER key; else, enter the name to be used for the library, or enter QUIT to exit.
DMSWVV792R	Are the macros to be erased from disk? Reply YES or NO (press the ENTER key for default of YES).
DMSWVV793I	MACLIB generation completed.
DMSWVV794E	Error in MACLIB generation
DMSWVV795E	Error reading macros from tape
DMSWVV796E	Error reading from VSEVSAM SCAN file
DMSWVV797I	QUIT may be entered in response to any query to end processing

DMSWVV798R	The VSE/VSAM optional source statement library tape must be mounted as virtual 181. If it is not, enter QUIT here and have the tape mounted; else, press ENTER to continue.
DMSWVV799E	Error reading from VSEVSAM SCAN file; all macros may not be erased
DMSWVV800E	One of the files needed for MACLIB generation is missing
DMSWVV801I	Arguments entered are ignored
DMSWVV802I	Macros erased; VSEVSAM processing complete
DMSWVV808R	Macro library *libname* will be erased. Press ENTER to continue or type QUIT to exit.
DMSWVV809E	Error copying VSEVSAM SCAN file from filemode S to filemode A.

Appendix A. Minidisks and SFS Directories Reserved for MAINT

Minidisks

Table A-1 lists the minidisks reserved for the MAINT virtual machine. Table A-2 on page A-3 defines the default or recommended size of each minidisk according to DASD type.

Table A-1 (Page 1 of 2). Minidisks Reserved for MAINT

Address	Volume	Used By	Function/Contents
096	VMPK01	HPO	Apply disk (auxiliary control files and update shells)
190	VMSRES	CMS	System disk (MODULEs, EXECs, MACLIBs, TXTLIBs, stand-alone service programs, some updated text decks)
191	VMSRES	CMS	Installation tools and profiles; work area
192	VMSRES	CP	Apply disk (auxiliary control files and update shells)
193	VMSRES[1]	CMS Procedures Language/VM IPCS GCS TSAF AVS	Base disk (system generation and maintenance tools, updated text decks); merge disk for preventive service Base disk (text decks); merge disk for preventive service Base disk (text decks); merge disk for preventive service IPCS interface files IPCS interface files IPCS interface files
194	VMSRES	CP	Base disk (text decks); merge disk for preventive service
195	VMPK01[2]	CMS Procedures Language/VM IPCS	Receive disk for corrective service Receive disk for corrective service Receive disk for corrective service
196	VMSRES	HPO	Base disk (text decks); merge disk for preventive service
19D	VMPK01[2]		System HELP files
19E[3]		CMS	Extension disk used by some optional feature program products
201	VMSRES[1]	EREP	TXTLIBs
293	VMSRES[2]	CMS Procedures Language/VM IPCS	Receive disk for preventive service (PTFs; auxiliary and update files; updated TEXT decks; replacement CNTRL, EXEC, MODULE, and XEDIT files) Receive disk for preventive service (serviced text decks and update files) Receive disk for preventive service (replacement CNTRL, EXEC, MODULE, and XEDIT files)
294	VMSRES[4]	CP	Receive disk for preventive service (PTFs; auxiliary and update files; ASSEMBLE, MACRO, COPY, and source files; updated TEXT decks, MACRO files, and COPY files; replacement CNTRL, EXEC, MODULE, and XEDIT files)

Table A-1 (Page 2 of 2). Minidisks Reserved for MAINT

Address	Volume	Used By	Function/Contents
295	VMSRES	CP CMS Procedures Language/VM IPCS	System definition files; merge disk for corrective service Nucleus generation profile; merge disk for corrective service Merge disk for corrective service Merge disk for corrective service
296	VMPK01	HPO	Receive disk for preventive service
319	VMSRES[1]		Licensed program "Memo to Users"
392	VMPK01[2]	CMS Procedures Language/VM IPCS	Apply disk Apply disk Apply disk
393	VMPK04	CMS IPCS	Source code disk (ASSEMBLE, $EXEC, $XEDIT files) Source code disk
394	VMPK04	CP	Source code disk (ASSEMBLE, MACRO, COPY files)
395	VMSRES	CP	Receive disk for corrective service
396	VMPK04	HPO	Source code disk
401[3]			KANJI HELP files
402[3]			Uppercase English HELP files
403[3]			Brazilian Portuguese HELP files
404[3]			French HELP files
405[3]			German HELP files
406[3]			Spanish HELP files
407[3]			Chinese Simplified (HANZI) HELP files
595	VMPK01[5]	GCS	System disk
596	VMPK01	HPO	Receive disk for corrective service
59E	VMPK01[2]	GCS	Extension disk for optional GCS applications

Notes:

[1] VMPK01 on 9313 and 9332 DASD

[2] VMPK02 on 9313 DASD

[3] Not defined in the sample CP directory

[4] VMPK01 on 9313 DASD

[5] VMSRES on 9332 DASD; VMPK02 on 9313 DASD

Table A-2. Default or Recommended Minidisk Sizes[1] for Each DASD Type

Minidisk	3330	3340	3350	3375	3380	FBA
096	13	31	7	9	6	n/a
190	173	429	80	120	78	72000
191	22	54	10	15	10	8936
192	12	28	6	8	5	4600
193	96	239	45	67	44	40000
194	72	179	34	50	33	30000
195	14	35	7	10	7	5848
196	42	105	20	30	20	n/a
19D[2]	91	227	43	64	41	38000
19E[3]						
201	50	123	23	35	23	20520
293	93	229	43	65	42	38464
294	47	115	22	33	21	19232
295	61	150	28	42	27	25088
296	43	105	20	30	20	n/a
319	12	30	6	9	6	5000
392	14	35	7	10	7	5848
393[4]	168	417	78	117	76	70000
394[4]	168	417	78	117	76	70000
395	12	28	6	8	5	4600
396[4]	92	228	44	65	42	n/a
401[3]						
402[3]						
403[3]						
404[3]						
405[3]						
406[3]						
407[3]						
595	67	167	32	47	31	28000
596	13	31	7	9	6	n/a
59E	22	54	10	15	10	9000

Notes:

[1] The indicated sizes for these minidisks are adequate to install the base VM/SP system. However, application of service might require that some minidisks be increased in size.

[2] If you plan to load user-written or other product HELP files on this minidisk, you will probably have to define it larger.

[3]The size of this minidisk depends on the requirements of any optional feature program products that are installed.

[4]The recommended size of this minidisk provides enough space to contain all of the packed source code loaded from the product tape plus some (but not all) unpacked modules. If you plan to unpack all of your files, you should redefine this minidisk 80% larger.

Shared File System Directories

The following directories are defined in the system file pool (VMSYS):

Table A-3. VMSYS Directories Reserved for MAINT

dirname	**Function**
VMSYS:MAINT.AVS.APPLY	AVS apply service code
VMSYS:MAINT.AVS.LCLSERV	Receiving AVS corrective service
VMSYS:MAINT.AVS.LCLSERV1	Merging AVS corrective service
VMSYS:MAINT.AVS.OBJECT	AVS text decks and LOADLIB; merge directory for PUT
VMSYS:MAINT.AVS.PUTSERV	Receiving AVS preventive service from PUT
VMSYS:MAINT.GCS.APPLY	GCS apply service code
VMSYS:MAINT.GCS.LCLSERV	Receiving GCS corrective service
VMSYS:MAINT.GCS.LCLSERV1	Merging GCS corrective service
VMSYS:MAINT.GCS.OBJECT	GCS text decks and MACLIBs; merge directory for PUT
VMSYS:MAINT.GCS.PUTSERV	Receiving GCS preventive service from PUT
VMSYS:MAINT.GCS.SOURCE	GCS source code
VMSYS:MAINT.SAMPLES	SFS sample (example) files
VMSYS:MAINT.TSAF.APPLY	TSAF apply service code
VMSYS:MAINT.TSAF.LCLSERV	Receiving TSAF corrective service
VMSYS:MAINT.TSAF.LCLSERV1	Merging TSAF corrective service
VMSYS:MAINT.TSAF.OBJECT	Text decks and module; merge directory for PUT
VMSYS:MAINT.TSAF.PUTSERV	Receiving TSAF preventive service from PUT
VMSYS:MAINT.TSAF.SOURCE	TSAF source code

Appendix B. Restricted Logon Passwords

The CP nucleus includes a system security feature called ADRP (Auto-Deactivation of Restricted Passwords). ADRP works with a CMS file named RPWLIST DATA that contains the list of restricted logon passwords shown below.

Issuing the DIRECT command to process the directory file causes the system to search the directory for the restricted passwords contained in this list. All passwords that match are changed to NOLOG in the directory before the directory is placed on line. You can not log on to any userid whose password has been changed to NOLOG. Therefore, make sure that you have changed **all** restricted passwords in the directory to unique non-restricted passwords before you issue the DIRECT command.

Table B-1. Restricted Logon Passwords

ACNT	AUTOLOG	AUTOLOG1	BATCH
CE	CMSUSER	CMS1	CMS2
CMS3	CPCMS	DIRM	ECMODE
GCS	IBMCE	IPCS	ISMAINT
ITPS	IVPASS	LEV2VM	MAINT
MASTER	MDVR	OPASS	OPERATNS
OPERATOR	OSVS1	PASSWORD	PRODBM
PROMAIL	PSR	ROUTER	RSCS
SFBATCH	SFCAL	SQLDBAPW	SQLUSER
SYSADMIN	SYSDUMP	VMAP	VSEIP
VSEIPO	VSEMAINT		

Appendix C. Sample SPLOAD PROFILE

```
/*---------------------------------------------------------------*/
/*               1600 BPI VM/SP Product Tape                     */
/*---------------------------------------------------------------*/
/*                                                               */
/* Group   Element    USERID        Address       Format    Vol File */
/*-------- --------   ------- ------------------   --------  --- -----*/
Install   Tools      MAINT          292          R6M05314  1    2
Sysgen    Tools      MAINT          193          R6M05314  1    3
System    Tools      MAINT          190          R6M05314  1    4
System    Samples    MAINT          295          R6M05314  1    5
DASDMODL  Samples    MAINT          295          R6M05314  1    6
DASDTYPE  Samples    MAINT          295          R6M05314  1    7
CP        Delta      MAINT          294          R6M05314  1    8
CP        Apply      MAINT          192          R6M05314  1    9
CP        Object     MAINT          194          R6M05314  1   10
CMS       Base       MAINT          193          R6M05314  1   11
/*                                                                 */
REXX      Delta      MAINT          293          R6M05314  2    2
REXX      Apply      MAINT          392          R6M05314  2    3
REXX      Object     MAINT          193          R6M05314  2    4
CMS       Delta      MAINT          293          R6M05314  2    5
CMS       Apply      MAINT          392          R6M05314  2    6
CMS       System     MAINT          190          R6M05314  2    7
/*                                                                 */
IPCS      Delta      MAINT          293          R6M05314  3    2
IPCS      Apply      MAINT          392          R6M05314  3    3
IPCS      Object     MAINT          193          R6M05314  3    4
Help      Files      MAINT          19D          R6M05314  3    5
/*                                                                 */
GCS       Delta      VMSYS:    MAINT.GCS.PUTSERV R6M05314  4    2
GCS       Apply      VMSYS:    MAINT.GCS.APPLY   R6M05314  4    3
GCS       Interface  MAINT          193          R6M05314  4    4
GCS       Object     VMSYS:    MAINT.GCS.OBJECT  R6M05314  4    5
TSAF      Delta      VMSYS:    MAINT.TSAF.PUTSERV R6M05314 4    6
TSAF      Apply      VMSYS:    MAINT.TSAF.APPLY  R6M05314  4    7
TSAF      Interface  MAINT          193          R6M05314  4    8
TSAF      Object     VMSYS:    MAINT.TSAF.OBJECT R6M05314  4    9
AVS       Delta      VMSYS:    MAINT.AVS.PUTSERV R6M05314  4   10
AVS       Apply      VMSYS:    MAINT.AVS.APPLY   R6M05314  4   11
AVS       Interface  MAINT          193          R6M05314  4   12
AVS       Object     VMSYS:    MAINT.AVS.OBJECT  R6M05314  4   13
/*                                                                 */
CP        Source     MAINT          394          R6M05314  5    2
/*                                                                 */
IPCS      Source     MAINT          393          R6M05314  6    2
CMS       Source     MAINT          393          R6M05314  6    3
/*                                                                 */
```

```
/*------------------------------------------------------------------*/
/*                   1600 BPI Source Code Tape                      */
/*------------------------------------------------------------------*/
/*                                                                  */
GCS       Source     VMSYS:    MAINT.GCS.SOURCE   R6M06002  1    2 *
TSAF      Source     VMSYS:    MAINT.TSAF.SOURCE  R6M06002  1    3 *
CMSFT     Source     MAINT            393         R6M06002  1    4 *
/*                                                                  */
/*------------------------------------------------------------------*/
/*        38K & 6250 BPI VM/SP Product Tape                         */
/*------------------------------------------------------------------*/
/*                                                                  */
Install   Tools      MAINT            191         R6M05315  1    2
Sysgen    Tools      MAINT            193         R6M05315  1    3
System    Tools      MAINT            190         R6M05315  1    4
System    Samples    MAINT            295         R6M05315  1    5
DASDMODL  Samples    MAINT            295         R6M05315  1    6
DASDTYPE  Samples    MAINT            295         R6M05315  1    7
CP        Delta      MAINT            294         R6M05315  1    8
CP        Apply      MAINT            192         R6M05315  1    9
CP        Object     MAINT            194         R6M05315  1   10
CMS       Base       MAINT            193         R6M05315  1   11
REXX      Delta      MAINT            293         R6M05315  1   12
REXX      Apply      MAINT            392         R6M05315  1   13
REXX      Object     MAINT            193         R6M05315  1   14
CMS       Delta      MAINT            293         R6M05315  1   15
CMS       Apply      MAINT            392         R6M05315  1   16
CMS       System     MAINT            190         R6M05315  1   17
IPCS      Delta      MAINT            293         R6M05315  1   18
IPCS      Apply      MAINT            392         R6M05315  1   19
IPCS      Object     MAINT            193         R6M05315  1   20
Help      Files      MAINT            19D         R6M05315  1   21
GCS       Delta      VMSYS:    MAINT.GCS.PUTSERV  R6M05315  1   22
GCS       Apply      VMSYS:    MAINT.GCS.APPLY    R6M05315  1   23
GCS       Interface  MAINT            193         R6M05315  1   24
GCS       Object     VMSYS:    MAINT.GCS.OBJECT   R6M05315  1   25
TSAF      Delta      VMSYS:    MAINT.TSAF.PUTSERV R6M05315  1   26
TSAF      Apply      VMSYS:    MAINT.TSAF.APPLY   R6M05315  1   27
TSAF      Interface  MAINT            193         R6M05315  1   28
TSAF      Object     VMSYS:    MAINT.TSAF.OBJECT  R6M05315  1   29
AVS       Delta      VMSYS:    MAINT.AVS.PUTSERV  R6M05315  1   30
AVS       Apply      VMSYS:    MAINT.AVS.APPLY    R6M05315  1   31
AVS       Interface  MAINT            193         R6M05315  1   32
AVS       Object     VMSYS:    MAINT.AVS.OBJECT   R6M05315  1   33
/*
IPCS      Source     MAINT            393         R6M05315  2    2 *
CP        Source     MAINT            394         R6M05315  2    3 *
CMS       Source     MAINT            393         R6M05315  2    4 *
/*                                                                  */
```

```
/*------------------------------------------------------------------*/
/*             38K & 6250 BPI Source Code Tape                      */
/*------------------------------------------------------------------*/
/*                                                                  */
GCS       Source     VMSYS:    MAINT.GCS.SOURCE   R6M06003  1    2 *
TSAF      Source     VMSYS:    MAINT.TSAF.SOURCE  R6M06003  1    3 *
CMSFT     Source     MAINT           393          R6M06003  1    4 *
/*                                                                  */
/*------------------------------------------------------------------*/
/*     38K & 6250 & 1600 BPI VM/SP NLS Feature Tape                 */
/*------------------------------------------------------------------*/
/*                                                                  */
CMS       Base       MAINT           193          R6XXXXXX  1    2
CP        Object     MAINT           194          R6XXXXXX  1    3
Help      Files      MAINT           ?            R6XXXXXX  1    4
CMS       Source     MAINT           393          R6XXXXXX  1    5
TSAF      Object     VMSYS:    MAINT.TSAF.OBJECT  R6XXXXXX  1    6
GCS       Object     VMSYS:    MAINT.GCS.OBJECT   R6XXXXXX  1    7
AVS       Object     VMSYS:    MAINT.AVS.OBJECT   R6XXXXXX  1    8
/*                                                                  */
/*                                                                  */
```

Appendix D. Special Options for CP

This appendix contains information about the following options that you can select when building a CP nucleus:

- Small CP (see page D-2)
- CP FRET Trap (see page D-4)
- Virtual=Real (see page D-7).

These options are selected by using a specific control file and/or build list (loadlist) to build the CP nucleus. The IBM-supplied product parameter file (5664167E $PPF) defines a CP control file and build list (DMKSP and CPLOAD, respectively) that build a CP nucleus **without** these options.

The method for selecting any of these options depends on the context of the nucleus build:

- If you are installing VM/SP using the Starter System procedure, you must tailor the product parameter file to define the necessary CP control file and/or loadlist. The procedures in this appendix provide instructions for tailoring the product parameter file.

- If you are installing VM/SP using the existing VM/SP system procedure, or if you are rebuilding an existing CP nucleus, the recommended method is to define the alternate CP control file and/or build list in a product parameter override file. The procedures in this appendix provide instructions for creating the override file. However, for complete information about creating override files and how they work, see the *VM/SP Service Guide*.

Warning: Before you select any of these options, you should read the sections of the *VM/SP Planning Guide and Reference* that discuss planning for system generation. Significant system programming experience might be required.

Small CP

Overview

The Small CP option lets you reduce the size of the CP nucleus by removing specific types of support, such as virtual = real (V = R) support, attached processor (AP) and multiprocessor (MP) support, and support for:

SUPPORT	MODULES
3066	DMKGRH
Remote 3270	DMKRGA, DMKRGB, DMKRGC, DMKRGD, DMKRGE
3340 Alternate Track	DMKTRK
3375/3380	DMKDAD
3704/3705/3725	DMKRNH
3800 Printers	DMKTCS, DMKTCT
3850 MSS	DMKSSS, DMKSSU, DMKSSV
MVS Guest	DMKFPS, DMKQVM, DMKVSC
SNA (CCS)	DMKVCP, DMKVCQ, DMKVCR, DMKVCS, DMKVCT, DMKVCU, DMKVCV, DMKVCW, DMKVCX, DMKVCY
TTY Terminal	DMKTTZ

If you need **any** of this support, you cannot use the Small CP option.

Note: If you cannot use the Small CP option, you can still remove some of this support individually to reduce the size of your nucleus. For more information, see the *VM/SP Planning Guide and Reference*.

Installation Procedure

Note: This option can be used only when building a uniprocessor system.

- (Installing VM/SP using the Starter System) XEDIT the product parameter file and locate the CP parameter area (beginning with :CP.). In the CP area, locate the build list record, which looks like this:

```
*BLDLST  EXEC      TARGET      DESCRIPTION
*
CPLOAD   VMFBDNUC              * Create CP nucleus based on loadlist
```

  In the build list record, change CPLOAD to **CPLOADSM** (you must use uppercase). You might also want to modify the description to indicate that the specified build list builds a Small CP nucleus.

- (Installing VM/SP using an existing VM/SP system, or rebuilding an existing CP nucleus) Create a product parameter override file (or modify an existing override file) and define a CP parameter override area. An override file can have any file name except the product ID (which is reserved for the base product parameter file), but the file type must be $PPF. Figure D-1 on page D-3 shows an override file that contains a CP override area called SMCP. The SMCP override area contains a build list statement (specifying CPLOADSM as the new build list) that updates (overrides) the build list statement in the base product parameter file.

```
:OVERLSTP. SMCP
:OVERLST.
:SMCP. CP 5664167E
:BLD. UPDATE
* build list for Small CP nucleus
CPLOADSM VMFBDNUC
:END.
```

Figure D-1. Example of an Override File Containing a CP Override Area for Selecting the Small CP Option

- Build the CP nucleus:
  - If you are building this nucleus as part of a complete VM/SP system installation, go back to to the procedure that you have been using to do the installation (Chapter 3 or Chapter 4).
  - If you are rebuilding an existing CP nucleus to include the Small CP option, use the nucleus build procedure described in the *VM/SP Service Guide*.

  If you used an override file to select this option, make sure that you specify the names of the override file and CP override area when you issue the VMFBLD command to build the CP nucleus.

CP FRET Trap

Overview

CP FRET Trap detects the release of areas of CP free storage that were previously released, or outside the boundaries of the storage given, or not assigned. You can use CP FRET Trap as an aid in solving problems caused by improper use of CP free storage and to solve many storage overlay problems.

Table D-1 lists the control files and MACLIBs that are provided for installing the FRET Trap option in uniprocessor (UP), attached processor (AP), and multiprocessor (MP) systems.

Table D-1. Control Files and MACLIBs for Building a CP Nucleus with the FRET Trap Option

System	Control File	MACLIB
UP	DMKSPT CNTRL	DMKSPT MACLIB
AP	DMKSPAT CNTRL	DMKSPAT MACLIB
MP	DMKSPMT CNTRL	DMKSPMT MACLIB

Each MACLIB contains an updated OPTIONS COPY file that has the option &FRETRAP changed from the default value of 1 to 0. When building the CP nucleus, the VMFBLD EXEC uses the applicable control file and MACLIB to assemble the modules DMKCPI, DMKFRE, and DMKFRT and generate text files that include the conditionally assembled FRET Trap code. Table D-2 lists the generated text files.

Table D-2. Text Files for Building a CP Nucleus with the FRET Trap Option

System	Text Files
UP	DMKCPI TXTFT DMKFRE TXTFT DMKFRT TXTFT
AP and MP	DMKCPI TXTAPFT DMKFRE TXTAPFT DMKFRT TXTAPFT

For information about using CP FRET Trap, see the *VM/SP Diagnosis Guide.*

Installation Procedure

- (Installing VM/SP using the Starter System) XEDIT the product parameter file and locate the CP parameter area (beginning with :CP.).

  Note: If you are building an attached processor (AP) or multiprocessor (MP) system, locate the CPAP or CPMP override area (beginning with :CPAP. or :CPMP., respectively).

  In the CP (or CPAP or CPMP) area, locate the control file record (beginning with :CNTRL.) and change the control file name to the appropriate control file shown in Table D-1 on page D-4. You might also want to modify the comment (beginning with *) in the control file record to indicate that the specified control file builds a CP nucleus with the FRET Trap option.

- (Installing VM/SP using an existing VM/SP system, or rebuilding an existing CP nucleus) Create a product parameter override file (or modify an existing override file) and define a CP parameter override area. An override file can have any file name except the product ID (which is reserved for the base product parameter file), but the file type must be $PPF. Figure D-2 shows an override file that contains a CP override area called CPOPT. The CPOPT override area contains a control file record that updates (overrides) the control file record in the base product parameter file. Use the control file name from Table D-1 on page D-4 that corresponds to the type of system you are building. The example in Figure D-2 shows the control file used to build a uniprocessor system.

  Note: If you are building an attached processor (AP) or multiprocessor (MP) system, you must also include a build list record in the override file to specify APLOAD as the build list (loadlist) used to build the CP nucleus. The example in Figure D-3 shows the control file and build list used to build an MP nucleus with the FRET Trap option.

```
:OVERLSTP. CPOPT
:OVERLST.
:CPOPT. CP 5664167E
:CNTRLOP.
* control file for FRET Trap option
:CNTRL. DMKSPT
:END.
```

Figure D-2. Example of an Override File Containing a CP Override Area for Building a CP Nucleus with the FRET Trap Option

```
:OVERLSTP. CPOPT
:OVERLST.
:CPOPT. CP 5664167E
:CNTRLOP.
* control file for MP nucleus with FRET Trap option
:CNTRL. DMKSPMT
:BLD. UPDATE
* build list for MP nucleus
APLOAD VMFBDNUC BUILD1
:END.
```

Figure D-3. Example of an Override File Containing a CP Override Area for Building an MP Nucleus with the FRET Trap Option

- If you also want to install the Virtual=Real option, see the procedure on page D-7.
- Build the CP nucleus:
  - If you are building this nucleus as part of a complete VM/SP system installation, go back to to the procedure that you have been using to do the installation (Chapter 3 or Chapter 4).
  - If you are rebuilding an existing CP nucleus to include the CP FRET Trap option, use the nucleus build procedure described in the the *VM/SP Service Guide*.

  If you used an override file to select this option, make sure that you specify the names of the override file and CP override area when you issue the VMFBLD command to build the CP nucleus.

Virtual = Real

Overview

The Virtual = Real (V = R) option lets you improve the performance of a specified virtual machine by removing the need for paging operations for that machine. If you build your system with this option, you define an area in real storage as the V = R area. When a virtual machine with the VIRT = REAL statement in its directory entry logs on to the system, the virtual machine's storage is allocated from this real storage area. All pages except page zero are allocated to the corresponding real storage locations. (CP normally controls real page zero.)

Keep in mind that:

- Your real storage size must be large enough to accommodate the CP nucleus, the V = R area (containing the entire V = R virtual machine), and the remaining pageable storage requirements of CP and the other virtual machines. The recommended size is 4M.
- Any V = R virtual machine that you define must be large enough to contain CMS (16M if you load CMS at the default address).
- You can define only one V = R area.
- Only one virtual machine can occupy the V = R area at one time.
- A CP nucleus that defines a V = R area can be loaded into a real machine (first level installation) or into a virtual machine (second level installation).
- Because this option removes pages from the dynamic paging area, it could degrade the performance of the other virtual machines in the system.

Installation Procedure

■ (Installing VM/SP using the Starter System) XEDIT the product parameter file and locate the CP parameter area (beginning with :CP.).

Note: If you are building an attached processor (AP) or multiprocessor (MP) system, locate the CPAP or CPMP override area (beginning with :CPAP. or :CPMP., respectively).

In the CP (or CPAP or CPMP) area, locate the build section (beginning with :BLD.) and change the build list name to the appropriate build list shown in Table D-3. You might also want to modify the comment (beginning with *) to indicate that the specified build list builds a CP nucleus with the V = R option.

Table D-3. Build List for Building a CP Nucleus with the V = R Option

System	Build List
UP	VRLOAD
AP	AVLOAD
MP	AVLOAD

■ (Installing VM/SP using an existing VM/SP system, or rebuilding an existing CP nucleus) Create a product parameter override file (or modify an existing override file) and define a CP parameter override area. An override file can have any file name except the product ID (which is reserved for the base product parameter file), but the file type must be $PPF. Figure D-4 on page D-8 shows an override file that contains a CP override area called CPOPT. The CPOPT override area contains a build list (loadlist) statement that updates (overrides) the build list statement in the base product parameter file. Use the build list name from Table D-3 on page D-7 that corresponds to the type of system you are building. The example in Figure D-4 shows the build list used to build a uniprocessor system.

Note: If you are building an attached processor (AP) or multiprocessor (MP) system, you must also include a control file record in the override file to specify DMKSPA or DMKSPM as the control file used to build the CP nucleus. The example in Figure D-5 shows the control file and build list used to build an AP nucleus with the V=R option.

```
:OVERLSTP. CPOPT
:OVERLST.
:CPOPT. CP 5664167E
:BLD. UPDATE
* build list for V=R option
VRLOAD VMFBDNUC BUILD1
:END.
```

Figure D-4. Example of an Override File Containing a CP Override Area for Building a CP Nucleus with the V=R Option

```
:OVERLSTP. CPOPT
:OVERLST.
:CPOPT. CP 5664167E
:CNTRLOP.
* control file for AP nucleus
:CNTRL. DMKSPA
:BLD. UPDATE
* build list for AP nucleus with V=R option
AVLOAD VMFBDNUC BUILD1
:END.
```

Figure D-5. Example of an Override File Containing a CP Override Area for Building an AP Nucleus with the V=R Option

- If you also want to install the CP FRET Trap option, see the procedure on page D-5.
- Modify the CP directory and specify the VIRT=REAL option in the entries of all the virtual machines that you want to have the capability of running in the V=R area. (For more information about this option, see the *VM/SP Planning Guide and Reference*.) Then place the modified directory on line:

direct ***fn ft***

- Specify the amount of storage that you want to reserve for the V=R area:

vrsize

```
VIRTUAL=REAL OPTION REQUIRED (YES,NO):
```

yes

```
STORAGE SIZE OF VIRT=REAL <MINIMUM IS 32K>:
```

***nnnn*k**

The *VM/SP Planning Guide and Reference* provides a formula that you can use to calculate the size of the V=R area. You must enter the size in K.

```
nnnnK STORAGE SIZE FOR VIRTUAL=REAL
IS THE ABOVE ENTRY CORRECT (YES,NO):
```

yes or **no**

- Build the CP nucleus:
  - If you are building this nucleus as part of a complete VM/SP system installation, go back to the procedure that you have been using to do the installation (Chapter 3 or Chapter 4).
  - If you are rebuilding an existing CP nucleus to include the Virtual=Real option, use the nucleus build procedure described in the *VM/SP Service Guide*.

  If you use an override file to select this option, make sure that you specify the names of the override file and CP override area when you issue the VMFBLD command to build the CP nucleus.

Appendix E. Migrating Spool Files Using SHUTDOWN/WARM IPL

Overview

You can use the SHUTDOWN/WARM IPL procedure to migrate spool files if:

- You are migrating from VM/SP Release 4 or later
- You are **not** using the Starter System procedure
- You explicitly control your own minidisk layout for CP-owned DASD
- Your system residence volume is **not** on a 3380-E4 or 3380-K4 DASD
- You are familiar with:
  - DIRECT command
  - UTILITY EXEC
  - DMKSYS ASSEMBLE
  - SYSOWN macro
  - SYSRES macro
  - DASD Dump/Restore (DDR) program
  - CP Format/Allocate program
  - Operating VM at second level
  - Using TEMP space for spool files.

If you do not meet **all** of these requirements, you must use the SPTAPE command to save your spool files on tape. For more information about SPTAPE, see the *VM/SP CP System Command Reference.*

In the SHUTDOWN/WARM IPL procedure, the new VM/SP system is WARM START IPLed from the information written to disk by the SHUTDOWN process of the old system.

WARNING

During SHUTDOWN processing, code is read into storage from the SYSNUC area on DASD and is used to complete the shutdown. Therefore, the CP nucleus at shutdown time must be at the **same** level as when the system was originally IPLed. **Do not** replace the CP nucleus of your production system with a new release level nucleus prior to the shutdown, or else the shutdown could fail, requiring a COLD START IPL.

Procedure

Note: Before you begin this procedure, make sure that you understand all of the operations required to complete the process.

- Build a copy of your first level VM/SP system in a second level virtual machine. A procedure for doing this is contained in the "VM Under VM" section of the *VM Running Guest Operating Systems* manual. However, make sure that you use the same DASD device types, volume labels, DASD allocation specifications, and complete CP directory information that you plan to use in the real first level system.

- Migrate the second level system to the new release level using **Steps 1-7** of the procedure contained in Chapter 4, "Installing VM/SP Using an Existing VM/SP System" on page 4-1, with the following additional requirements:

  1. Use UTILITY EXEC (in Step 5) to create a service utility tape containing stand-alone versions of the DDR and CP Format/Allocate programs.

  2. When you tailor the second level DMKSYS ASSEMBLE file (in Step 6):

     - Make sure that the SYSOWN list specifies the same volumes as the first level SYSOWN list, in the same order. **Do not** remove or re-order any volumes. If you want to add new volumes to the second level SYSOWN list, add them at the end of the list.

     - Make sure that the SYSWRM and SYSCKP parameters of the SYSRES macro specify the label of the first level system residence volume and the same beginning cylinder/block numbers as the first level SYSWRM and SYSCKP.

  See the following examples of how to modify DMKSYS when using existing DASD plus new DASD (Example 1), or when using existing DASD only (Example 2).

Example 1: Using Existing DASD Plus New DASD

Assume that the DMKSYS ASSEMBLE file for your first level system contains the following values:

```
SYSOWN   PACKAA,
         SP4RES,
         PACKBB
SYSRES   SYSVOL=SP4RES,
         SYSTYPE=3380,
         SYSWRM=(10,10),
         SYSCKP=(20,10),
```

If the system residence device in your new system is a new 3380 DASD with label SP6RES, and you are also adding a new CP-owned DASD with label PACKCC, the second level DMKSYS ASSEMBLE file should contain the following values:

```
SYSOWN   PACKAA,
         SP4RES,
         PACKBB,
         SP6RES,
         PACKCC
SYSRES   SYSVOL=SP6RES,
         SYSTYPE=3380,
         SYSWRM=(10,10,SP4RES),
         SYSCKP=(20,10,SP4RES),
```

The order of the new volumes (SP6RES and PACKCC) in the SYSOWN list is not critical, as long as they follow the existing volumes.

Example 2: Using Existing DASD Only

Assume that the DMKSYS ASSEMBLE file for your first level system contains the following values:

```
SYSOWN   PACKAA,
         VMSRES,
         PACKBB
SYSRES   SYSVOL=VMSRES,
         SYSTYPE=3380,
         SYSWRM=(10,10),
         SYSCKP=(20,10),
```

If the system residence device in your new system remains VMSRES, and you are not adding any new CP-owned DASD, the second level DMKSYS ASSEMBLE file should contain the following values:

```
SYSOWN   PACKAA,
         VMSRES,
         PACKBB
SYSRES   SYSVOL=VMSRES,
         SYSTYPE=3380,
         SYSWRM=(10,10,VMSRES),
         SYSCKP=(20,10,VMSRES),
```

- After you build the new CP nucleus (in Step 7), while still running at second level:

  1. Use the DDR command with the DUMP NUcleus control statements to dump the second level CP nucleus on tape.

  2. Send a copy of the second level system's tailored CP directory (VMUSERS DIRECT, for example) to the first level user ID (MAINT, for example) that you plan to use to update the first level system.

  3. Send a copy of the second level system's DIRECT MODULE to the same user ID where you sent the CP directory.

- Stop all other use of the first level system. Have all users log off, then DRAIN the unit-record devices.

- Use the CP Format/Allocate program to make any required changes to the allocation of the volumes to be used with the new system.

  WARNING

  Do not change any space defined as TEMP in the first level configuration (it might contain spool files). **Do not** destroy the SYSWRM and SYSCKP spaces on the first level system residence volume.

- Log on to the first level virtual machine that you are using to update the first level system and make sure that you have the following:

  – A write (W) link to the first level DASD volume on which the CP directory for the new system will be written

- The **tailored** CP directory (that you sent from the second level system) accessed ahead of your old CP directory
- The **new** DIRECT MODULE (that you sent from the second level system) accessed ahead of the old version.

Using the **new** DIRECT MODULE, enter the DIRECT command to process the tailored CP directory and write it to the DASD volume defined for the new system. This lets you log on to the new system after the shutdown.

- Shut down the first level production system.
- Use the stand-alone DDR program to do a backup of each volume listed in the SYSOWN macro in the DMKSYS ASSEMBLE file.
- Use the stand-alone DDR program with the RESTORE NUcleus control statements to restore the CP nucleus from tape to the real system residence volume defined for the new system.
- Use the CP Format/Allocate program to verify (and update, if necessary) the allocation values on the new system residence volume.
- IPL the system residence volume, and specify WARM for the type of IPL. You now have the new level CP and CP directory on your production system, and the old level spool files are usable.
- Use the VMFPLC2 command with the DUMP and LOAD parameters to move your second level CP and CMS minidisks (MAINT 190, 191, 193, 194, 295, etc.) to your first level production system. For more information about the VMFPLC2 command, see the *VM/SP Service Guide.*
- Continue the migration procedure with “Step 8. Load HELP (Optional)” on page 4-39.

Appendix F. Enlarging the CMS Nucleus to Contain the Y Minidisk Directory (Y-STAT)

Overview

Because CMS is designed to run as a shared system, performance is generally improved when the S and Y minidisk directories (also called the file status tables, or S-STAT and Y-STAT) are included in the CMS nucleus.

When you IPL the CMS saved segment by name (**ipl cms**), if you see the message:

```
DMSACC723I Y (19E) R/O
```

the Y-STAT could not fit within the CMS nucleus at the time the system was saved. In that case, every user who IPLs CMS gets the Y minidisk accessed in non-shared user free storage, which is less efficient.

To enlarge the CMS nucleus to contain the Y-STAT, you must:

- Determine the number of additional segments needed and compute the new CMS starting address
- Modify the CMS loadlist (CMSLOAD EXEC)
- Modify the DMKSNT entry for CMS (and possibly other segments)
- Assemble DMKSNT
- Create a new SLC L*xxxxxx* file referenced by CMSLOAD EXEC
- Rebuild the CP nucleus
- Shut down and reIPL to make the changes effective
- Build and save the CMS nucleus.

Procedure

Note: This procedure assumes that you have a standard CMSLOAD EXEC, as shipped with your system.

- Log on as MAINT.
- Compute the new CMS starting address (NUCALPHA):

1. Obtain the number of files (*F*) on both the S and Y minidisks. Enter:

   query disk

   and record the number of files on both minidisks.

2. Obtain the hexadecimal addresses for DMSSIG, DMSOME, and DMSALP from the CMS nucleus load map.

   Compute the difference (*D*) between DMSOME and DMSSIG and convert *D* to a decimal value.

   Note: If the hexadecimal address for DMSOME shown in the load map is 000000, use 1000000 in your calculations.

3. Compute the additional amount of storage needed (*A*) by using the applicable formula:

   $A = 40 * F - D$ (for 800-byte formatted minidisk)

   - or -

   $A = 64 * F - D$ (for all others)

   where *F* is the number of files on S and Y minidisks, and *D* is the address difference in bytes.

4. Use the following table to determine the address hex value that you must subtract from the old CMS starting address:

If A is more than	But less than or =	Segments needed	Address hex value
-	00000	None	-
00000	65536	1	X'10000'
65537	131072	2	X'20000'
131073	196608	3	X'30000'

5. Determine the new CMS starting address (NUCALPHA) by subtracting the address hex value from the address for DMSALP.

Note: This method of computing address increments yields approximate values. It does not take into account variables such as doubleword boundaries or headers. Therefore, you might need to make an adjustment to the next lower segment to compensate. Also, you might have to recompute the CMS starting address each time you apply service.

- Edit the CMS loadlist (CMSLOAD EXEC) and change the Set Location Counter (SLC) value that marks the start of the CMS nucleus in virtual storage:

```
xedit cmsload exec
```

```
top
```

```
c/ SLC LE00000 / SLC Lxxxxxx / *
```

xxxxxx is the new CMS starting address (NUCALPHA) that you computed above.

```
file
```

■ Modify the CMS entry in the DMKSNT file:

```
xedit dmksnt assemble
```

Change the following four parameters on the CMS NAMESYS macro:

Note: To make these modifications, you might have to reposition other saved segments defined in DMKSNT.

SYSSTRT Alter this parameter to point to an area on disk that can hold the CMS system now that its size has increased.

SYSPGNM Increase the third group of pages to cover all of the extra segments to be shared.

SYSHRSG Increase this parameter to include the extra segments to be shared.

SYSPGCT Increase this parameter to include the additional pages to be saved.

■ Assemble DMKSNT:

```
access 194 c
```

vmfasm dmksnt *ppfname* *compname* (ppf

ppfname is the file name of the product parameter file that contains the CP parameters you want to use:

- If you are using only the base product parameter file, enter **5664167e**.
- If you want to use alternative and/or additional parameters defined in a product parameter override file, enter the name of the override file.

compname is the name of the component area in the product parameter file (base file or override file) that contains the CP parameters you want to use:

- If you are using only the base CP area in the base file (to build a uniprocessor system), enter **cp**.
- If you want to use alternative and/or additional CP parameters defined in an override area, either in the base file or in an override file, enter the name of the override area. For example, to use the control file and loadlist defined in the CPAP override area or CPMP override area in the base file (to build an attached processor system or multiprocessor system), enter **cpap** or **cpmp**.

■ Create a new SLC L*xxxxxx* file that is referenced by CMS:

access 193 b/a

copyfile SLC LF00000 b = L*xxxxxx* a

xxxxxx is the new CMS starting address (NUCALPHA) that you previously computed.

xedit SLC L*xxxxxx* a

top

c / F00000 / *xxxxxx* / *

file

■ Rebuild the CP nucleus:

vmfbld ***ppfname*** ***compname*** ***bldlist***

ppfname is the file name of the product parameter file (base file or override file) that contains the CP parameters you want to use.

compname is the name of the component area in the product parameter file (base file or override file) that contains the CP parameters you want to use.

bldlist is the file name of the loadlist you want to use to build this nucleus, **if that loadlist is different from the one specified in the product parameter file (base file or override file) you are using**. If you do not want to change the loadlist, you can omit this operand.

- Shut down and re-IPL to make the changes effective:

#cp shutdown reipl

Because this command performs an automatic warm start that automatically logs you on to the OPERATOR virtual machine, you might want to enter the command from the system operator's console.

- Log on as MAINT.
- Examine the CMS nucleus generation profile (DMSNGP ASSEMBLE) to make sure that the parameters are defined correctly for this nucleus. If you make any changes, reassemble the file:

vmfasm dmsngp ***ppfname*** ***compname*** **(ppf**

ppfname is the file name of the product parameter file that contains the CMS parameters you want to use:

- If you are using only the base product parameter file, enter **5664167e**.
- If you want to use alternative and/or additional parameters defined in a product parameter override file, enter the name of the override file.

compname is the name of the component area in the product parameter file (base file or override file) that contains the CMS parameters you want to use:

- If you are using only the base CMS area in the base file, enter **cms**.
- If you want to use alternative and/or additional CMS parameters defined in aL override area, either in the base file or in an override file, enter the name of the override area.

■ Build and save the CMS nucleus:

vmfbld *ppfname* *compname* **cmsload**

ppfname is the file name of the product parameter file (base file or override file) that contains the CMS parameters you want to use.

compname is the name of the component area in the product parameter file (base file or override file) that contains the CMS parameters you want to use.

Appendix G. Example of Alternate CMS Nucleus Placement

Overview

This appendix demonstrates how to save your CMS system at a storage location other than the one provided by the product tape samples. You might want such an alternative for users with low storage requirements, or when a high nucleus location will cause real storage constraints in the construction of segment tables.

The sample CMS saved system is contained in segments 224-255, which are page numbers 3584-4095, and storage location X'E00000'-X'FFFFFF'. This 16M address requires 1024 bytes of real storage for each CMS user. Before you relocate the nucleus, you should evaluate the trade-off between real storage availability and CMS usage. In addition, when you decide on the optimal CMS location for your user mix, you should consider the size of the page tables that CP builds for the addressable memory for a given machine.

To relocate your CMS saved system, you need to:

- Change the SLC names in the CMSLOAD EXEC
- Create SLC files to contain the new load address locations
- Modify the DMKSNT entries for CMS
- Assemble DMKSNT
- Rebuild the CP nucleus
- Shut down and reIPL to make the changes effective
- Build and save the CMS nucleus.

Procedure

This procedure demonstrates how to relocate your CMS saved system to a storage address of 4M. It requires 256 bytes of real storage for each CMS user. The CMS saved system is contained in segments 64-95, which are pages 1024-1535, and storage location X'400000'-X'5FFFFF'.

The *VM/SP Planning Guide and Reference* provides a worksheet to help you convert segment and page numbers to hexadecimal addresses.

■ Log on as MAINT and access the CMS base minidisk:

```
access 193 a
```

■ Edit the CMS loadlist (CMSLOAD EXEC A) and change the Set Location Counter (SLC) values that determine where the shared portion of CMS is loaded into virtual storage.

In the following example, "SLC LE00000" is the name of a CMS file that contains the address of the starting location for loading the main portion of the CMS nucleus. This SLC statement precedes the DMSALP entry in the loadlist. "SLC L000000" marks the end of the CMS nucleus, and it precedes the DMSOME entry.

Note: The CMS loadlist might be changed by service. After applying a PUT, you should check the file and change it if necessary before you rebuild CMS.

Enter the following commands:

```
xedit cmsload exec a

set case upper

top

change / SLC LE00000 / SLC L400000 /*

top

change / SLC L000000 / SLC L600000 /*

file
```

■ Create two new SLC files to match the new loadlist:

```
xedit SLC L400000 a

input $SLC  400000
```

There must be two blanks between SLC and the address.

```
set hex on
```

change /$/X'02'/

X'02' is an unprintable loader control character.

file

xedit SLC L600000 a

input $SLC 600000

There must be two blanks between SLC and the address.

set hex on

change /$/X'02'/

X'02' is an unprintable loader control character.

file

- Modify the SYSPGNM and SYSHRSG parameters in the CMS NAMESYS macro in DMKSNT.

  The following figures show a sample (3380 DASD) CMS NAMESYS macro and how the revised CMS NAMESYS macro might look.

```
**********************************************************************
*     HEX LOAD ADDRESS FOR SEGMENT 224 = E00000                      *
*     THE SPACE FOR CMS IS ALLOCATED ON VMSRES, AS FOLLOWS:          *
*     (THE ALLOCATIONS ARE BASED ON 150 PAGES/3380 CYLINDER)         *
*       CYL 6, PAGE 1   TO CYL 9, PAGE 96    (546 PAGES)             *
*       545 PAGES FOR CMS, 1 FOR CP INFORMATION.                     *
**********************************************************************
CMS       NAMESYS    SYSNAME=CMS,                                    X
               SYSVOL=VMSRES,                                        X
               SYSSTRT=(006,1),                                      X
               SYSPGNM=(0-13,16-34,3584-4095),                       X
               SYSPGCT=545,                                          X
               SYSHRSG=(224-255),                                    X
               SYSSIZE=256K,                                         X
               VSYSADR=190,                                          X
               SYSCYL=533,                                           X
               PARMRGS=(0,15),                                       X
               VSYSRES=VMSRES
          EJECT
```

Figure G-1. Sample CMS NAMESYS Macro (3380 DASD)

```
**********************************************************************
*     HEX LOAD ADDRESS FOR SEGMENT 64 = 400000                       *
*     THE SPACE FOR CMS IS ALLOCATED ON VMSRES, AS FOLLOWS:          *
*     (THE ALLOCATIONS ARE BASED ON 150 PAGES/3380 CYLINDER)         *
*       CYL 6, PAGE 1   TO CYL 9, PAGE 96    (546 PAGES)             *
*       545 PAGES FOR CMS, 1 FOR CP INFORMATION.                     *
**********************************************************************
CMS       NAMESYS    SYSNAME=CMS,                                    X
               SYSVOL=VMSRES,                                        X
               SYSSTRT=(006,1),                                      X
               SYSPGNM=(0-13,16-34,1024-1535),                       X
               SYSPGCT=545,                                          X
               SYSHRSG=(64-95),                                      X
               SYSSIZE=256K,                                         X
               VSYSADR=190,                                          X
               SYSCYL=533,                                           X
               PARMRGS=(0,15),                                       X
               VSYSRES=VMSRES
          EJECT
```

Figure G-2. Example of Alternate CMS Nucleus Placement

- Assemble DMKSNT:

vmfasm dmksnt ***ppfname*** ***compname*** **(ppf**

ppfname is the file name of the product parameter file that contains the CP parameters you want to use:

- If you are using only the base product parameter file, enter **5664167e**.
- If you want to use alternative and/or additional parameters defined in a product parameter override file, enter the name of the override file.

compname is the name of the component area in the product parameter file (base file or override file) that contains the CP parameters you want to use:

- If you are using only the base CP area in the base file (to build a uniprocessor system), enter **cp**.
- If you want to use alternative and/or additional CP parameters defined in an override area, either in the base file or in an override file, enter the name of the override area. For example, to use the control file and loadlist defined in the CPAP override area or CPMP override area in the base file (to build an attached processor system or multiprocessor system), enter **cpap** or **cpmp**.

- Rebuild the CP nucleus:

vmfbld *ppfname* *compname* *bldlist*

ppfname is the file name of the product parameter file (base file or override file) that contains the CP parameters you want to use.

compname is the name of the component area in the product parameter file (base file or override file) that contains the CP parameters you want to use.

bldlist is the file name of the loadlist you want to use to build this nucleus, **if that loadlist is different from the one specified in the product parameter file (base file or override file) you are using**. If you do not want to change the loadlist, you can omit this operand.

- Shut down and re-IPL to make the changes effective:

#cp shutdown reipl

Because this command performs an automatic warm start that automatically logs you on to the OPERATOR virtual machine, you might want to enter the command from the system operator's console.

- Log on as MAINT.
- Examine the CMS nucleus generation profile (DMSNGP ASSEMBLE) to make sure that the parameters are defined correctly for this nucleus. If you make any changes, reassemble the file:

vmfasm dmsngp *ppfname* *compname* (ppf

ppfname is the file name of the product parameter file that contains the CMS parameters you want to use:

- If you are using only the base product parameter file, enter **5664167e**.
- If you want to use alternative and/or additional parameters defined in a product parameter override file, enter the name of the override file.

compname is the name of the component area in the product parameter file (base file or override file) that contains the CMS parameters you want to use:

- If you are using only the base CMS area in the base file, enter **cms**.
- If you want to use alternative and/or additional CMS parameters defined in an override area, either in the base file or in an override file, enter the name of the override area.

■ Build and save the CMS nucleus:

vmfbld *ppfname* *compname* cmsload

ppfname is the file name of the product parameter file (base file or override file) that contains the CMS parameters you want to use.

compname is the name of the component area in the product parameter file (base file or override file) that contains the CMS parameters you want to use.

Appendix H. Example of Alternate GCS Nucleus Placement

Overview

This appendix demonstrates how to save your GCS system at a virtual storage location other than the one provided by the product tape samples. You might want the GCS segment at a different location, depending on the other segments being used by your GCS group.

This section also describes how to increase the size of the GCS saved system. You might want to change the size of the GCS saved system to increase the amount of available common storage. See the *VM/SP Planning Guide and Reference* for information about calculating common storage requirements for GCS.

To relocate your GCS saved system and change its size, you need to:

- Change the SLC names in the GCSLOAD EXEC
- Create SLC files to contain the new address locations
- Modify the DMKSNT entries for GCS
- Assemble DMKSNT
- Rebuild the CP nucleus
- Shut down and re-IPL to make the changes effective
- Build and save the GCS nucleus.

Procedure

This procedure demonstrates how to relocate your GCS saved system in a 10M virtual machine. Originally 1M in size and loaded at X'400000', the segment is moved to X'800000' and increased in size to 2M.

- Log on as MAINT and access the GCS system minidisk:

```
access 595 a
```

- Modify the GCS loadlist (GCSLOAD EXEC A) and change the Set Location Counter (SLC) values that determine where the shared portion of GCS is loaded into virtual storage.

  In the following example, "SLC L400000" is the name of a CMS file that contains the address of the starting location for loading the main portion of the GCS nucleus. This SLC statement precedes the CSIALP entry in the loadlist. "SLC L500000" marks the end of the GCS nucleus (also the end of common storage), and it precedes the CSIZET entry.

  Note: The GCS loadlist might be changed by service. After applying a PUT, you should check the file and change it if necessary before you rebuild the GCS nucleus.

  Enter the following commands:

```
xedit gcsload exec a

set case upper

top

change / SLC L400000 / SLC L800000 /*

top

change / SLC L500000 / SLC LA00000 /*

file
```

- Create two new SLC files to match the new loadlist:

```
xedit SLC L800000 a

input $SLC  800000
```

There must be two blanks between SLC and the address.

```
set hex on

change /$/X'02'/
```

X'02' is an unprintable loader control character.

file	
xedit SLC LA00000 a	
input $SLC A00000	There must be two blanks between SLC and the address.
set hex on	
change /$/X'02'/	X'02' is an unprintable loader control character.
file	

- Modify the SYSPGNM, SYSPGCT, and SYSHRSG parameters in the GCS NAMESYS macro in DMKSNT. As a result, you might also have to change other NAMESYS entries in DMKSNT.
  - To calculate the new values for SYSPGNM:
    1. Calculate the page number where the GCS nucleus begins:
       a. Use the corresponding SLC value (in this example the new SLC value is X'800000').
       b. Divide X'800000' by the page size, X'1000' (decimal equivalent is 4096). The result is X'800'.
       c. Convert X'800' to its decimal equivalent (2048). This is the new starting page of the GCS nucleus.
    2. In a similar manner, calculate the page number where GCS ends:
       a. Divide X'A00000' by X'1000'.
       b. Convert the result (A00) to decimal (2560).
       c. Subtract one; this gives you the last page of the nucleus (2559).
    3. Enter the new values in the SYSPGNM parameter list.
  - To calculate the new value for SYSPGCT:
    1. Count all the pages that are specified in the SYSPGNM entry.

       In this case, the count is 519 (pages 0-6 plus pages 2048-2559).
    2. Enter this value on the SYSPGCT parameter.

       Note: If your page count is greater than 1948, more than one extra page is needed for use by CP. For more information on saved system DASD requirements, refer to the *VM/SP Planning Guide and Reference.*
  - To calculate the new values for SYSHRSG (the shared segment numbers):
    1. Find the number of the first segment.

       To do this, divide the first page number (in this case, 2048) by 16, which is the number of pages per segment. The result in this example is 128.
    2. Calculate the remaining segment numbers.

In our example, the next segment starts at page 2064, so the segment number is 129. Continuing in this manner, we find that the last segment number is 159.

3. Enter the numbers of the shared segments in the SYSHRSG parameter list.

   These are the numbers of the segments that contain the GCS nucleus.

The following figures show a sample (3380 DASD) GCS NAMESYS macro and how the revised GCS NAMESYS macro might look.

```
*********************************************************************
*     HEX LOAD ADDRESS FOR SEGMENT  64 = 400000                     *
*     THE SPACE FOR GCS IS ALLOCATED ON VMPK01, AS FOLLOWS:         *
*         CYL 12, PAGE 1  TO CYL 13, PAGE 114  (264 PAGES)          *
*********************************************************************
GCS      NAMESYS SYSNAME=GCS,                                       X
               SYSVOL=VMPK01,                                       X
               SYSSTRT=(12,001),                                    X
               SYSPGNM=(0-6,1024-1279),                             X
               SYSPGCT=263,                                         X
               SYSHRSG=(064-079),                                   X
               SYSSIZE=256K,                                        X
               VSYSADR=595,                                         X
               SYSCYL=709,                                          X
               VSYSRES=VMPK01,                                      X
               PROTECT=OFF,                                         X
               VMGROUP=YES
         EJECT
```

Figure H-1. Sample GCS NAMESYS Macro

```
*********************************************************************
*     HEX LOAD ADDRESS FOR SEGMENT 128 = 800000                     *
*     THE SPACE FOR GCS IS ALLOCATED ON VMPK01, AS FOLLOWS:         *
*         CYL 12, PAGE 1  TO CYL 15, PAGE 70  (520 PAGES)          *
*********************************************************************
GCS      NAMESYS SYSNAME=GCS,                                       X
               SYSVOL=VMPK01,                                       X
               SYSSTRT=(12,001),                                    X
               SYSPGNM=(0-6,2048-2559),                             X
               SYSPGCT=519,                                         X
               SYSHRSG=(128-159),                                   X
               SYSSIZE=256K,                                        X
               VSYSADR=595,                                         X
               SYSCYL=709,                                          X
               VSYSRES=VMPK01,                                      X
               PROTECT=OFF,                                         X
               VMGROUP=YES
         EJECT
```

Figure H-2. Example of Alternate GCS Nucleus Placement

■ Assemble DMKSNT:

| **vmfasm dmksnt *ppfname* *compname* (ppf**

ppfname is the file name of the product parameter file that contains the CP parameters you want to use:

- If you are using only the base product parameter file, enter **5664167e**.
- If you want to use alternative and/or additional parameters defined in a product parameter override file, enter the name of the override file.

compname is the name of the component area in the product parameter file (base file or override file) that contains the CP parameters you want to use:

- If you are using only the base CP area in the base file (to build a uniprocessor system), enter **cp**.
- If you want to use alternative and/or additional CP parameters defined in an override area, either in the base file or in an override file, enter the name of the override area. For example, to use the control file and loadlist defined in the CPAP override area or CPMP override area in the base file (to build an attached processor system or multiprocessor system), enter **cpap** or **cpmp**.

■ Rebuild the CP nucleus:

| **vmfbld *ppfname* *compname* *bldlist***

ppfname is the file name of the product parameter file (base file or override file) that contains the CP parameters you want to use.

compname is the name of the component area in the product parameter file (base file or override file) that contains the CP parameters you want to use.

bldlist is the file name of the loadlist you want to use to build this nucleus, **if that loadlist is different from the one specified in the product parameter file (base file or override file) you are using**. If you do not want to change the loadlist, you can omit this operand.

- Shut down and re-IPL to make the changes effective:

```
#cp shutdown reipl
```

Because this command performs an automatic warm start that automatically logs you on to the OPERATOR virtual machine, you might want to enter the command from the system operator's console.

- Log on as MAINT.
- Build and save the GCS nucleus:

  Note: Before you build the nucleus, you might want to create a new GCS configuration file for this nucleus. See the GCS step in the VM/SP installation procedure. The name of the configuration file is specified in the GCS loadlist. The name of the GCS loadlist is specified in the product parameter file.

```
vmfbld ppfname compname gcsload
```

ppfname is the file name of the product parameter file (base file or override file) that contains the GCS parameters you want to use.

compname is the name of the component area in the product parameter file (base file or override file) that contains the GCS parameters you want to use.

```
ipl cms
```

Appendix I. Considerations for Placement of the DOS-Related Saved Segments

If you need to move the DOS-related physical saved segments (CMSDOS, CMSBAM, CMSVSAM, and CMSAMS) to storage locations different from the default definitions in DMKSNT, remember the following guidelines:

- For installation purposes, CMSDOS must be defined at a higher address than CMSBAM, CMSVSAM, and CMSAMS. CMSDOS must be discontiguous from virtual storage while it is in use, and CMSDOS is used to install CMSBAM, CMSVSAM, and CMSAMS, each of which must be within your virtual storage when they are being installed. CMSBAM, CMSVSAM, and CMSAMS can be in any order below CMSDOS.

- Before loading any of the DOS-related segments, you must define your virtual machine with at least 512K free storage at a higher address than the end of the segment to be saved. This provides room for the loader tables and file status tables that occupy the top of virtual storage during the building of the segment. After you load and save the segment, the 512K free storage area is no longer required. This means that the end of the CMSDOS saved segment must be defined at an address at least 512K lower than the beginning address of the CMS nucleus in DMKSNT.

- You must install CMSDOS before you can load and save CMSBAM, CMSVSAM, and CMSAMS. You must also define your virtual machine with at least 512K free storage above the end of the highest of the three segments (CMSBAM, CMSVSAM, or CMSAMS), but not greater than the address where you defined CMSDOS. This means that the end of the highest segment must be defined at an address at least 512K less than the beginning address of CMSDOS in DMKSNT.

  Notes:

  1. CMSDOS is composed of one 64K shared segment.
  2. CMSBAM is composed of three 64K shared segments.
  3. CMSVSAM is composed of seven 64K segments, of which the first six are shared and the last one is non-shared.
  4. CMSAMS is composed of nine 64K segments, of which the first six are shared and the last three are non-shared.

- The same DOS-related saved segments are used by both CMS and GCS, so you must make sure that they are all defined at an address higher than the end of any GCS virtual machines that use them.

Figure I-1 on page I-2 shows the DOS-related saved segments loaded at locations other than those defined in the supplied DMKSNT. Figure I-2 on page I-2 shows the revised DMKSNT entries that match the locations in Figure I-1.

```
X'300000' -----|---------------------|
               | CMSDOS              |
               | size = 1 segment    |
X'2F0000' -----|---------------------|
               | inter-segment gap   |
               | size = 8 segments   |
X'270000' -----|---------------------|
               | CMSBAM              |
               | size = 3 segments   |
X'240000' -----|---------------------|
               | CMSVSAM             |
               | size = 7 segments   |
X'1D0000' -----|---------------------|
               | CMSAMS              |
               | size = 9 segments   |
X'140000' -----|---------------------|
```

Figure I-1. Example of Other Storage Locations for the DOS-Related Saved Segments

```
***********************************************************************
*     THE SPACE FOR CMSDOS IS ALLOCATED ON VMPK01, AS FOLLOWS:
*    ( THE ALLOCATIONS ARE BASED ON 150 PAGES/3380 CYLINDER )
*          CYL 007, PAGE 001 TO CYL 007, PAGE 017  (017 PAGES)
*         17 PAGES = 16 PAGES FOR CMSDOS 1 FOR CP INFORMATION
***********************************************************************
CMSDOS   NAMESYS SYSNAME=CMSDOS,                                       X
               SYSVOL=VMPK01,                                          X
               SYSSTRT=(007,01),                                       X
               SYSPGNM=(752-767),                                      X
               SYSPGCT=16,                                             X
               SYSHRSG=(47),                                           X
               SYSSIZE=256K,                                           X
               SYSCYL=,                                                X
               VSYSRES=,                                               X
               VSYSADR=IGNORE
         EJECT
```

Figure I-2 (Part 1 of 4). Example of Revised DMKSNT Entries for the DOS-Related Saved Segments

```
***********************************************************************
*     THE SPACE FOR CMSBAM IS ALLOCATED ON VMPK01, AS FOLLOWS:
*    ( THE ALLOCATIONS ARE BASED ON 150 PAGES/3380 CYLINDER )
*          CYL 008, PAGE 001 TO CYL 008, PAGE 049  (049 PAGES)
*         49 PAGES = 48 PAGES FOR CMSBAM 1 FOR CP INFORMATION
***********************************************************************
CMSBAM   NAMESYS SYSNAME=CMSBAM,                                       X
               SYSVOL=VMPK01,                                          X
               SYSSTRT=(008,01),                                       X
               SYSPGNM=(576-623),                                      X
               SYSPGCT=48,                                             X
               SYSHRSG=(36-38),                                        X
               SYSSIZE=256K,                                           X
               SYSCYL=,                                                X
               VSYSRES=,                                               X
               VSYSADR=IGNORE
         EJECT
```

Figure I-2 (Part 2 of 4). Example of Revised DMKSNT Entries for the DOS-Related Saved Segments

```
***********************************************************************
*     THE SPACE FOR CMSVSAM IS ALLOCATED ON VMPK01, AS FOLLOWS:
*    ( THE ALLOCATIONS ARE BASED ON 150 PAGES/3380 CYLINDER )
*          CYL 009, PAGE 001 TO CYL 009, PAGE 113  (113 PAGES)
*         113 PAGES = 112 PAGES FOR CMSVSAM 1 FOR CP INFORMATION
***********************************************************************
CMSVSAM  NAMESYS SYSNAME=CMSVSAM,                                      X
               SYSVOL=VMPK01,                                          X
               SYSSTRT=(009,01),                                       X
               SYSPGNM=(464-575),                                      X
               SYSPGCT=112,                                            X
               SYSHRSG=(29-34),                                        X
               SYSSIZE=256K,                                           X
               SYSCYL=,                                                X
               VSYSRES=,                                               X
               VSYSADR=IGNORE
         EJECT
```

Figure I-2 (Part 3 of 4). Example of Revised DMKSNT Entries for the DOS-Related Saved Segments

```
***********************************************************************
*     THE SPACE FOR CMSAMS IS ALLOCATED ON VMPK01, AS FOLLOWS:
*    ( THE ALLOCATIONS ARE BASED ON 150 PAGES/3380 CYLINDER )
*          CYL 010, PAGE 001 TO CYL 010, PAGE 145  (145 PAGES)
*         145 PAGES = 144 PAGES FOR CMSAMS 1 FOR CP INFORMATION
***********************************************************************
CMSAMS   NAMESYS SYSNAME=CMSAMS,                                       X
               SYSVOL=VMPK01,                                          X
               SYSSTRT=(010,01),                                       X
               SYSPGNM=(320-463),                                      X
               SYSPGCT=144,                                            X
               SYSHRSG=(20-25),                                        X
               SYSSIZE=256K,                                           X
               SYSCYL=,                                                X
               VSYSRES=,                                               X
               VSYSADR=IGNORE
         EJECT
```

Figure I-2 (Part 4 of 4). Example of Revised DMKSNT Entries for the DOS-Related Saved Segments

Appendix J. Sample EXEC Procedure for Copying VSE Macros into a CMS MACLIB

Overview

This appendix tells you how to create an EXEC procedure called DOSMAC EXEC. You can use DOSMAC EXEC to help you create a VSE macro library under CMS.

Notes:

1. This procedure applies to VSE Version 1 **only**.
2. This procedure has not been formally tested by IBM; it is presented here for your convenience only.

To invoke DOSMAC, you must be in CMS/DOS mode. If you are using a private source statement library, you must issue the appropriate ACCESS, ASSGN, and DLBL commands, specifying the VSE minidisk where that library resides.

DOSMAC creates a DSERV listing on your CMS minidisk and uses the source statement directory listing to create an EXEC file that issues a separate ESERV command for each VSE macro. You can then use the CMS Editor or the System Product Editor (XEDIT) to delete all the ESERV commands for macros you do not wish to move at this time. DOSMAC then creates a MACLIB (CMS macro library) with a file name that you specify. If you do not specify a file name, the default is DOSMAC.

If you have too many VSE macros to move to your CMS minidisk, the MACLIB build process may exceed the CMS file system limitation and abnormally terminate. All macros prior to the one that caused the error message were probably cataloged correctly. Reinvoke DOSMAC EXEC and use the CMS Editor or System Product Editor to delete the ESERV commands for all the macros previously cataloged. You must also specify some other file name (such as DOSMAC2) for this new macro library.

Alternatively, if you want to avoid the abnormal termination of the MACLIB build process, you can delete some or all of the ESERV commands for the following VSE macros the first time that you invoke DOSMAC:

BTMOD
CDMOD
DAMOD
DAMODV
FOPT
IOINTER
IOTAB
ISMOD
MCRAS
MTMOD
SGCCWT
SGEND
SGPMAIN
SGPSUB
SGSVC
COBBG
COBF2

Note: Check a DSERV listing and delete the ESERV commands for the largest VSE macros first. Then manually create a second set of ESERV commands, specifying those macros not included in the first CMS MACLIB.

Procedure

Note: This procedure applies to VSE Version 1 **only**.

- Create a file called DOSMAC EXEC:

```
xedit dosmac exec
```

- Issue the INPUT subcommand to get into input mode.
- Enter the following lines:

```
&CONTROL OFF
&GENSWT = 0
CP PURGE RDR ALL
CP SP 9 * CLASS A
&TYPE ENTER THE ADDRESS OF YOUR SYSRES VOLUME ( DEFAULT = 350 )
&READ ARGS
&IF &INDEX EQ 0 ACCESS 350 Z
&IF &INDEX NE 0 ACCESS &1 Z
SET DOS ON Z  ( VSAM
&TYPE IF YOU WISH TO ASSGN AND DLBL A PRIVATE SOURCE STATEMNT LIBRARY
&TYPE NOW IS THE TIME ( ENTER YOUR ASSGN ). IF YOU DO NOT ENTER A NULL LINE
&READ
&TYPE A DLBL IS ALSO REQUIRED FOR SSL
&READ
-MACGEN &CONTINUE
&TYPE ENTER THE NAME OF THE MACLIB TO BE CREATED THE DEFAULT IS DOSMAC
&READ ARGS
&IF &INDEX EQ 0 &LIB = DOSMAC
&IF &INDEX NE 0 &LIB = &1
*
*  The output of the DSERV command is spooled to
*  your virtual card reader and is read in as $ESER EXEC.
*
CP SPOOL CONS START NOTERM
DSERV SD ( TERM
CP SPOOL CONS STOP TERM
CP CLOSE 9
READ $ESER EXEC
*
*  The $ESER EXEC file is copied, edited, and formatted.
*  All DSERV header and trailer lines are deleted.
*
COPYFILE $ESER EXEC A $ESERV EXEC A ( LRECL 80  REPLACE
&BEGSTACK
DEL 9
F CP
DEL 5
TOP
C / /&1 &2/*
FILE
&END
EDIT $ESERV EXEC
```

```
ERASE $ESER EXEC
-STACKER &CONTINUE
&BEGTYPE
*
*  If you wish to delete any of the generated ESERV commands, enter
*  any nonblank character. If you do not wish to delete any ESERV
*  commands (or after you have deleted them), enter a null line.
*
*  If you wish to alter the list of macros, now is the time to do so.
*  You may bypass alteration by entering a null line or else enter a
*  non-blank character to begin alteration.  Alteration is accomplished
*  via editing the exec file containing the macro names.  You must
*  issue the edit subcommand "file" to re-enter this exec and continue.
*
&END
&READ ARGS
&IF &INDEX NE 0 EDIT $ESERV EXEC
&CONTROL ALL
*
*  Stack the remaining lines of the $ESERV EXEC
*  in the console stack.
*
EXEC $ESERV &STACK SPACE
ASSGN SYSIN A
ASSGN SYSLST PRINTER
ASSGN SYSPCH PUNCH
CP SPOOL D TO *
&CONTROL ALL
-GETNEXT &CONTINUE
&READ ARGS
*
*  Read a line from the console stack and check that
*  the first letter begins with E (for ESERV).
*  If not an E, ignore the line and read the next one.
*
&IF &2 NE E &GOTO -STAKTST
*
*  If it is an E, create a DSPCH fn for this macro.
*  Note:  PUNCH or DSPLY may be substituted for DSPCH.
*
&STACK LIFO FILE
&STACK LIFO C /$/ / 4
&STACK LIFO TOP
&STACK LIFO I $DSPCH &3
EDIT &3 ESERV
*
*  Execute the ESERV command.  The re-edited macro is
*  spooled to your virtual card reader.
*
EXEC ESERV &3
ERASE &3 ESERV
*
*  Read the macro file onto the CMS minidisk.
*  Delete the CATALS statement.
*
READ &3 MACRO
```

```
&STACK LIFO FILE
&STACK LIFO DEL
&STACK LIFO BO
&STACK LIFO DEL
&STACK LIFO L /CATALS/
EDIT &3 MACRO
&IF &GENSWT NE 0 &GOTO -MACADD
&GENSWT = 1
MACLIB GEN &LIB &3
ERASE &3 MACRO
&GOTO -STAKTST
-MACADD &CONTINUE
*
*  Add the macro to the indicated CMS macro library.
*
MACLIB ADD &LIB &3
ERASE &3 MACRO
&IF &READFLAG EQ STACK &GOTO -GETNEXT
-FINALE &CONTINUE
&STACK QUIT
&BEGTYPE
THE MACLIB &LIB HAS BEEN CREATED AND THE FOLLOWING IS A MAP OF THE LIBRARY
&END
&STACK MACLIB MAP &LIB ( TERM
&EXIT
-STAKTST &CONTINUE
&IF &READFLAG EQ STACK &GOTO -GETNEXT
&GOTO -FINALE
```

- Close the file:

file

- Invoke the DOSMAC EXEC that you just created:

dosmac

Note: The ESERV process could take a substantial length of time, up to several hours for a large macro library. For more information about the ESERV command, see the *VM/SP CMS Command Reference*. For more information about how to use the ESERV command, see the *VM Application Development Guide for CMS*. For a detailed description of the VSE ESERV control statements, see the *Guide to the DOS/VSE Assembler*, GC33-4024.

Summary of Changes

How To Obtain Previous Editions of This Book: You can obtain previous editions of this book by using the pseudo order numbers listed in the *VM/SP Library Guide and Master Index.*

Summary of Changes
for SC24-5237-4
for VM/SP Release 6.0

Integration of Between-Release Support Information to VM/SP Release 6:

- *VM/SP VM/VTAM and NETVIEW™ Enhancements,* GC24-5310
- *VM/SP 9370 Processors, 9332 and 9335 Direct Access Storage Devices, and 9347 Tape Drive,* GC24-5315
- *VM IBM 3380 Direct Access Storage Models AJ4/BJ4 and AK4/BK4,* GC24-5371
- *VM IBM 3990 Storage Controls Models 1 and 2 and IBM 3380 Direct Access Storage Direct Channel Attach Model CJ2,* GC24-5372.

Deleted:

- All service information (formerly Chapters 11-14 and Appendixes K-N) has been deleted. Service information is now located in a new book, the *VM/SP Service Guide,* SC24-5389.
- All EXEC and command descriptions formerly in Chapter 15 (except ITASK, SPLOAD, VMFDOS, and VSEVSAM) have been moved to the *VM/SP Service Guide.*
- The Starter System version for the 3330-11 DASD has been deleted.
- The appendix describing the procedure for "Installing the 3704/3705 Control Program" has been dropped. However, the descriptions of the ASM3705, GEN3705, and SAVENCP commands have been retained in Chapter 8, "Tools."

Added:

- FBA Starter System versions have been added for the 9313/9332 and 9335/3370 DASD.
- Planning and installation information has been added for the APPC/VM VTAM Support (AVS) component.
- The ITASK EXEC has new operands for:
  - Specifying the component (CP or CMS) of a file to be assembled
  - Building an attached processor or multiprocessor CP nucleus
  - Building a system file pool and a user file pool
  - Loading AVS code.
- Steps have been added to the Starter System procedure and existing VM/SP system procedure for:
  - Building an SFS system file pool
  - Building an SFS user file pool
  - Loading AVS code.
- Delta and apply service code for all components is supplied on the product tape and loaded to minidisks and SFS directories during the installation procedure.

Trademark of International Business Machines Corporation

- Installation procedures have been added for two new saved segments:
  - CMSVMLIB contains the VMLIB callable services library.
  - CMSFILES contains SFS server code.
- A table identifying the SFS directories assigned to MAINT has been added to Appendix A.
- Appendix I, "Considerations for Placement of the DOS-Related Saved Segments," contains information about installing CMSDOS, CMSBAM, CMSVSAM, and CMSAMS at locations other than those defined in the supplied DMKSNT file.

Changed:

- The VM/SP product parameter file (5664167E $PPF) has replaced SPGEN PROFILE as the repository of system generation parameters.
- VMFBLD EXEC has replaced SPGEN EXEC as the nucleus build tool invoked by ITASK EXEC.
- System Product Interpreter code has been separated from CMS and GCS as a new component called Procedures Language/VM; the component identifier is IXX. Tape files for this component on the VM/SP Product Tape have the file name REXX.
- Object code and source code for GCS and TSAF is loaded to SFS directories.
- The procedures for installing CP options (Small CP, CP FRET Trap, and Virtual = Real) when using the existing VM system procedure to install VM/SP now use product parameter override files.

Summary of Changes
for SC24-5237-3
for VM/SP Release 5.0

Deleted:

- Co-requisite manual, *VM/SP System Definition Files*
- Starter System installation procedure for 3310, 3330 Models 1 and 2, and 3340 DASD
- Non-Merged Product Tape installation procedure
- CMSL nucleus option, due to revised DMKSNT layout.

Added and Changed:

- Starter System Tape, VM/SP Product Tape, and feature tapes available in 18-track 38K bpi cartridge format.
- Transparent Services Access Facility (TSAF) component added to installation procedure.
- "Introduction to VM/SP Installation" includes the following additional information:
  - Pre-installation checklist
  - Overview of VM/SP, including descriptions of system components
  - Definitions of real and virtual machines
  - Descriptions of new tools and profiles for installation and system generation
  - Expanded definitions of first level installation and second level installation.
- Chapter 2, "Planning Your VM/SP Installation" is new, and outlines areas that may require planning (with pointers to other documentation for detailed information), such as:
  - Selecting the installation procedure
  - Migrating spool files

- Identifying DASD requirements
- Defining the starter system configuration
- Planning for the Group Control System (GCS)
- Planning for the Transparent Services Access Facility (TSAF)
- Tailoring SPGEN PROFILE
- Tailoring the system definition files
- Installing national languages.

- Chapter 3, "Installing VM/SP Using the Starter System" describes the new starter system installation procedure:
  - Steps within the procedure have been reorganized.
  - New tools and profiles for installation and system generation replace GENERATE EXEC and PREP EXEC.
  - GROUP EXEC panels for GCS installation are included. This information was previously contained in the *VM/SP Group Control System Guide.*

- Chapter 4, "Installing VM/SP Using an Existing VM/SP System" describes the new non-starter system migration procedure:
  - Steps within the procedure have been reorganized.
  - New tools and profiles for installation and system generation replace GENERATE EXEC and PREP EXEC.
  - GROUP EXEC panels for GCS installation are included. This information was previously contained in the *VM/SP Group Control System Guide.*

- Chapter 6, "Installing Saved Segments" includes a new procedure to install a CMSINST segment.

  The CMSINST segment is designed to hold frequently-used EXECs and System Product Editor macros, so that multiple users can share the same executing copies.

- Chapter 7, "Installing a New System National Language" is new.

  VM/SP is supplied with American English as the system national language. This chapter describes the procedure to install a new system national language to replace American English.

- Chapter 11, ″Introduction to VM/SP Service″ contains more details about general service concepts for VM/SP. It contains descriptions of:
  - Types of service (corrective, preventive, and local updates)
  - MAINT virtual machine
  - Service minidisks defaults
  - Disk Access Profiles

- Chapter 12, ″Servicing Source Maintained VM/SP Products″ contains new information and information previously contained in Chapters 6, 7, and 8. In addition to the examples using VMFMAC, VMFASM, and VMFLOAD, three new sections have been added.
  - "Using PSPACE to Determine Maximum Module Size"
  - "Using VMFTXT to Create Text Libraries"
  - "Using VMFNLS to Update National Language Source Files."

- Chapter 13, ″Servicing Object Maintained VM/SP Products″ is new. This chapter describes how to apply corrective and preventive service to the VM/SP components which are object code maintained (i.e. GCS, TSAF, and part of CMS).

- Chapter 14, ″Updating Service Programs, Modules, and Shared Segments″ is new but the only new information is the section "Updating Interactive Problem Control System (IPCS)." This information was previously contained in the *VM/SP IPCS User's Guide,* which has been deleted.

- Chapter 15, ″EXEC and Command Procedures″ includes the following new installation, system generation, and service EXECs:
  - GENTSF EXEC
  - ITASK EXEC

- SPLOAD EXEC
- SPGEN EXEC
- UTILITY EXEC
- VMFNLS EXEC
- VMFREMOV EXEC
- VMFTXT EXEC

In addition, descriptions of the following existing EXECs and commands are now included in this chapter:

- DIRECT MODULE
- VMSERV EXEC
- ZAP MODULE
- ZAPTEXT EXEC

Note that all EXECs (and messages issued by those EXECs) used during the installation and service procedures are contained in this chapter.

- Appendix A, "Minidisks Reserved for the MAINT Userid" is new, and provides information about the address, size, and contents of the minidisks defined for the MAINT userid in the base CP directory.

- Appendix B, "Restricted Logon Passwords" is new, and contains information about the ADRP (Auto-Deactivation of Restricted Passwords) feature, including a list of the restricted passwords contained in the RPWLIST DATA file.

- Appendix C, "Sample Installation/System Generation Profiles" is new, and contains samples of SPLOAD PROFILE, SPGEN PROFILE, and the CMS nucleus generation profile (DMSNGP ASSEMBLE).

- Appendix E, "Migrating Spool Files Using SHUTDOWN/WARM IPL" is new, and provides a procedure for migrating spool files from a VM/SP Release 3 or Release 4 system to a Release 5 system.

- Appendix K, "Building CP and CMS Nuclei Using SPGEN" contains information previously contained in Chapter 8. The information has been revised and reorganized to make use of the new SPGEN EXEC functions.

- Appendix L, "Control File Identifiers" is new; however, the information contained in the appendix was previously in Chapter 7.

- Appendix M, "Regenerating CP/CMS/IPCS Modules/Segments" has the following information added:

  - Creating CMS disk resident modules
  - Loading CMS disk resident modules
  - Modules to Regenerate from IPCS Text Files

- Appendix N, "Servicing Systems Network Architecture (SNA) Products," previously Appendix J. (with a different chapter title), includes the following new information:

  - Reqby Log
  - Remove List
  - Examples of how to merge service
  - Examples of how to remove service
  - Different procedure for removing a fix-in-error

Summary of Changes
for SC24-5237-2
for VM/SP Release 4.0

- VM/SP now supports the following hardware devices:

  - 3290 Information Panel
  - 3370 Direct Access Storage Models A2 and B2
  - 3480 Magnetic Tape Subsystem Unit
  - 4248 Printer

- MAINT's 190 minidisk is split into MAINT's 190 (System disk) and MAINT's 193 (Base disk). MAINT's 193 minidisk is used for system maintenance and generation.
- MAINT's 19E minidisk is no longer defined in the sample directories.
- The Device Support Facilities (DSF) Program is now on the first tape file of the Starter System tape.
- The CLEAR option when saving the CMS or CMSL nucleus has been removed.
- The Group Control System (GCS) support is required if you plan to install System Network Architecture Products or RSCS Version 2.
- Due to revised DMKSNT layouts, the INSTVSAM segment is no longer required.
- "Chapter 2. Installation When Using the Starter System Tape" reflects the following:
  - New considerations for restructuring user classes in your directory
  - Loading the Device Support Facilities (DSF) Program
  - Completing the Customer Profile file for IPCS
  - Shared/Nonshared Restriction

    With the addition of this support, you cannot construct a virtual device configuration that would mix SHARED and NONSHARED device types on the same virtual control unit.
  - Performing new optional PREP options for the Group Control System (GCS) and generating the GCS nucleus. These include:

    PREP GCSFMT
    PREP GCSLOAD
    GENERATE GCS (systemname)
  - Installing EREP
- "Chapter 3. Installation When Not Using the Starter System Tape" reflects the following:
  - New considerations for restructuring user classes in your directory
  - Completing the Customer Profile file for IPCS
  - Performing new optional PREP options for the Group Control System (GCS) and generating the GCS nucleus. These include:

    PREP GCSFMT
    PREP GCSLOAD
    GENERATE GCS (systemname)
- "Chapter 7. Recommended Procedures for Applying Local Updates" reflects the fact that the control file you specify on the VMFMAC command is optional.
- "Chapter 8. Updating and Installed VM/SP System" reflects:
  - The new Stand-Alone Dump enhancement to VM/SP service. The Stand-Alone Dump facility is an enhancement to VM/SP service. It lets support personnel dump up to sixteen megabytes of real storage. It is required to dump real storage when VM/SP is not able to create a CP abend dump.
  - Using the COMPARE MODULE (instead of ACCESS) to Load CMS Disk-Resident Modules.
- "Chapter 9. EXEC Procedures and Command Format Summaries" reflects the following:
  - The addition of a new planning tool, the SNTMAP EXEC
  - New PREP EXEC options for the Group Control System (PREP GCSFMT, PREP GCSLOAD, and GENERATE GCS (systemname)).
  - The addition of a new service EXEC, VMFLKED.
  - Changes to the VMFMAC EXEC since specifying a control file is now optional.
- "Appendix F. Example of Alternate GCS Nucleus Placement" is new.

  The format and contents of this appendix is similar to what is in "Appendix E. Example of Alternate CMS Nucleus Placement."

- "Appendix H. Installing the CP FRET Trap" is new.

  The CP FRET Trap can be used as an aid in solving problems caused by improper use of CP free storage overlay. It can detect the release of areas of free storage that were not assigned, previously released, or outside the boundaries of the storage given.

- "Appendix J. Other Service Exec Files and Procedures" and "Appendix K. Messages and Codes" are both new. They contain information regarding new service execs (VMFMERGE and VMFZAP) and the messages and codes for those two service EXECs and for VMFLKED.

Installation Guide Reorganization: The sample files that originally appeared in Appendixes J through P have been removed. They now appear in a manual entitled: *VM/SP System Definition Files.* Material about DASD SNTMAPS, Virtual MEMORY SNTMAPS, and sample allocations for the different DASD packs, has also been added to the manual mentioned above.

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

Glossary of Terms and Abbreviations

A

abend. (1) Abnormal end of task. (2) Synonym for *abnormal termination.*

abend dump. The contents of main storage, or part of main storage, written to an external medium for debugging an error condition that resulted in the termination of a task before its regular completion.

abnormal end of task (abend). Termination of a task before its completion because of an error condition that cannot be resolved by recovery facilities while the task is executing.

abnormal termination. The ending of processing before planned termination. Synonymous with *abend.*

Advanced Program-to-Program Communications (APPC). The inter-program communication service within SNA LU 6.2 on which the APPC/VM interface is based.

Advanced Program-to-Program Communications/VM (APPC/VM). An API for communicating between two virtual machines that is mappable to the SNA LU 6.2 APPC interface and based on IUCV functions. Along with the TSAF virtual machine, APPC/VM provides this communication within a single system and throughout a collection of systems.

AP. Attached processor.

APAR. Authorized program analysis report.

APPC. Advanced Program-to-Program Communications.

APPC/VM. Advanced Program-to-Program Communications/VM.

APPC/VM VTAM Support (AVS). A component of VM/SP that lets application programs using APPC/VM communicate with programs anywhere in a network defined by IBM's SNA. AVS transforms APPC/VM into APPC/VTAM protocol.

attached processor (AP). A processor that has no I/O capability and is always linked to the processor initialized for I/O handling.

attention interrupt. An I/O interrupt caused by a terminal user pressing the attention key (or equivalent). See *attention key (ATTN key).*

attention key (ATTN key). A function key on terminals that, when pressed, causes an I/O interruption in the processing unit.

ATTN key. Attention key.

authority. In SFS, the permission to access a file or directory. You can have read authority or write authority (which includes read authority). You can also have file pool administration authority, which is the highest level of authority in a file pool.

authorized program analysis report (APAR). An official request to the responsible IBM Change Team to look into a suspected problem with IBM code or documentation. APARs describe problems giving conditions of failure, error messages, abend codes, or other identifiers. They also contain a problem summary and resolution when applicable.

authorized user ID. In GCS, a user ID that provides access to the GCS supervisor, supervisor state, and (in some cases) certain restricted CP commands. This access is provided by including the user ID on a list of authorized user IDs compiled with the GCS GROUP EXEC. The virtual machine associated with an authorized user ID is an *authorized* machine, and programs running in that machine are *authorized* applications.

auxiliary directory. In CMS, an extension of the CMS file directory for a minidisk, which contains the names and locations of certain CMS modules not included in the minidisk's CMS file directory.

AVS. APPC/VM VTAM Support.

AVS virtual machine. The virtual machine that manages a gateway that allows communication between VM systems and an SNA network.

B

basic control (BC) mode. A mode in which additional System/370 features, such as new machine instructions, are not operational. Contrast with *extended control (EC) mode.*

bootstrap program. A technique or device that brings itself into a desired state by its own action. For example, a machine routine whose first few instructions are sufficient to bring the rest of itself into the computer from an input device.

buffer. An area of storage, temporarily reserved for performing input or output, into which data is read, or from which data is written.

build. In reference to installation and service of a product, to do the necessary steps to produce executable code or systems. This is often called the *build process.*

build list. An EXEC file that names the parts included in an object being built.

C

callable services library (CSL). A package of CMS assembler routines that can be stored as an entity and made available to application programs.

channel. A path in a system that connects a processor and main storage with an I/O device.

checkpoint (CKPT) start. A VM/SP system restart that attempts to recover information about closed spool files previously stored on the checkpoint cylinders. The spool file chains are reconstructed, but the original sequence of spool files is lost. Unlike warm start, CP accounting and system message information is also lost. Contrast with *cold start, force start,* and *warm start.*

CKD. Count-key-data.

class authority. Privilege assigned to a virtual machine user in the user's directory entry; each class specified allows access to a subset of all the CP commands. See *privilege class* and *user class restructure (UCR).*

class override file. A file containing control statements defining changes in the privilege classes of CP commands and diagnose codes. The override program, DMKOVR, uses this file to establish a new class structure of commands under UCR.

CMS. Conversational Monitor System.

CMS file directory. A directory on each CMS disk that contains the name, format, size, and location of each of the CMS files on that disk. When a disk is accessed by the ACCESS command, its directory is read into virtual storage and identified with any letter from A through Z. Synonymous with *minidisk directory.*

CMS loader work file. The CMS file loaded into storage by the LOAD command.

CMS nucleus. The portion of CMS that is resident in the user's virtual storage whenever CMS is executing. Each CMS user receives a copy of the CMS nucleus when the user IPLs CMS. See *saved system* and *shared segment.*

CMS system minidisk. The minidisk located at virtual address 190. It contains the system commands for which the user has read-only access.

cold start. A VM/SP system restart that ignores previous data areas and accounting information in main storage, and the contents of paging and spool files on CP-owned disks. Contrast with *checkpoint (CKPT) start, force start,* and *warm start.*

collection. See *TSAF collection.*

command privilege class. See *privilege class.*

common dump receiver. One user ID in a virtual machine group appointed to receive other group members' storage dumps. Unless the user specifies otherwise, all dumped information automatically goes to this user ID (identified with the GCS GROUP EXEC). It should be an authorized user ID in order to receive fetch-protected data as well as storage with a key other than 14.

common storage. A shared segment of reentrant code that contains free storage space, the GCS supervisor, control blocks, and data that all members of a virtual machine group share.

component. A collection of objects that together form a separate functional unit. A product may contain many components (for example, VM/SP has components of CP, CMS, GCS, TSAF, IPCS, AVS, and Procedures Language/VM). A component can be part of many products. (CP spans both VM/SP and VM/HPO products.)

component override. Synonym for *component parameter override.*

component override area. An area of the product parameter file or of a product parameter override file that contains one or more component parameter overrides. Synonymous with *override area.*

component parameter override. A component parameter, defined in a component override area, that updates or replaces a component parameter defined in a component area of the product parameter file. Synonymous with *component override* and *override.*

console spooling. Synonym for *virtual console spooling.*

contention. The situation where two LUs try to allocate a conversation over the same session at the same time.

control block. A storage area that a computer program uses to hold control information.

Control Program (CP). A component of VM/SP that manages the resources of a single computer so multiple computing systems appear to exist. Each virtual machine is the functional equivalent of an IBM System/370.

control section (CSECT). The part of a program specified by the programmer to be a relocatable unit, all elements of which are loaded into adjoining main storage.

Conversational Monitor System (CMS). A virtual machine operating system and component of VM/SP that provides general interactive time sharing, problem solving, program development capabilities, and operates only under the control of the VM Control Program (CP).

copy file. A file having file type COPY that contains nonexecutable real storage definitions that are referred to by macros and assemble files.

count-key-data (CKD) device. A disk storage device that stores data in the format: count field, usually followed by a key field, followed by the actual data of a record. The count field contains the cylinder number, head number, record number, and the length of the data. The key field contains the record's key (search argument).

CP. Control Program.

CP directory. Synonym for *VM/SP directory.*

CSECT. Control section.

CSL. Callable services library.

D

DASD Dump Restore (DDR) program. A service program that copies all or part of a minidisk onto tape, loads the contents of a tape onto a minidisk, or sends data from a DASD or from tape to the virtual printer.

DBCS. Double-byte character set.

DDR program. DASD Dump Restore program.

device support facilities. A program for doing operations on disk volumes so that they can be accessed by IBM and user programs. Examples of these operations are initializing a disk volume and assigning an alternate track.

directory name (dirname). A fully-qualified directory name that can incorporate a period (.) to indicate the user's own top directory (used in commands).

dirname. Directory name.

discontiguous saved segment. One or more 64K segments of storage that were previously loaded, saved, and assigned a unique name. The segment(s) can be shared among virtual machines if the segment(s) contains reentrant code. Discontiguous segments used with CMS must be loaded into storage at locations above the address space of a user's CMS virtual machine. They can be detached when no longer needed.

DMSPARMS file. A CMS file with a file type of DMSPARMS that contains the start-up parameters that file pool server processing uses.

double-byte character set (DBCS). A character set that requires 2 bytes to uniquely define each character. This contrasts with EBCDIC, in which each printed character is represented by 1 byte.

dump. To write the contents of part or all of main storage, or part or all of a minidisk, to auxiliary storage or a printer. See *abend dump.*

dynamic paging area (DPA). An area of real storage that CP uses for virtual machine pages and pageable CP modules.

E

emulation program (EP). A control program that lets an IBM 3704 or 3705 Communications Controller emulate the functions of an IBM 2701 Data Adapter Unit, an IBM 2702 Transmission Control Unit, or an IBM 2703 Transmission Control Unit.

entry point. An address or label of an instruction performed upon entering a computer program, a routine, or a subroutine. A program can have several different entry points, each corresponding to a different function or purpose.

environmental record editing and printing program (EREP). A program that makes the data contained in the system recorder file available for further analysis.

EP. Emulation program.

EREP. Environmental record editing and printing program.

error recording area. The DASD space that the system programmer defines during system generation on the system residence volume that CP uses to record formatted outboard error recordings, machine check records, and channel check records. For CKD devices, this area is between 2 and 9 contiguous cylinders in size; for FB-512 devices, the size of this area can be any number of contiguous pages.

extended control (EC) mode. A mode in which all features of a System/370 computing system, including dynamic address translation, are operational. Contrast with *basic control (BC) mode.*

F

FB-512. An FBA device that stores data in 512-byte blocks (refers to the IBM 9335, 9332, 9313, 3370, and 3310 DASDs).

FBA. Fixed-block architecture.

file pool. A collection of minidisks managed by SFS. It contains user files and directories and associated control information. Many users' files and directories can be contained in a single file pool.

file pool server machine. A virtual machine that is properly configured to manage a file pool. (Its VM system directory entries must, for example, contain the MDISK statements for a file pool.)

file space. A user's allocation of space within a file pool.

file status table (FST). A table that describes the attributes of a file on a CMS disk, including file name, file type, file mode, date last written, and other status information.

fixed-block architecture (FBA) device. A disk storage device that stores data in blocks of fixed size or records; these blocks are addressed by block number relative to the beginning of the particular file.

force start. A VM/SP system restart that attempts to recover information about closed spool files previously stored on the checkpoint cylinders. All unreadable or invalid spool file information is ignored. Contrast with *checkpoint (CKPT) start, cold start,* and *warm start.*

forms control buffer (FCB). In the 3800 Printing Subsystem, a buffer for controlling the vertical format of printed output. The FCB is analogous to the punched-paper, carriage-control tape that IBM 1403 Printers use.

free storage. Storage not allocated. The blocks of memory available for temporary use by programs or by the system.

FST. File status table.

G

gateway. The LU name of a TSAF collection that is a source for communications to an SNA-defined network or the target of communications from an SNA-defined network.

gateway manager. A virtual machine in which one or more gateways are active. AVS is a gateway manager.

GCS. Group Control System.

group. Synonym for *virtual machine group.*

group configuration file. A file that the GROUP EXEC creates. It contains the *blueprint* for building the user's virtual machine group. The name of the file is *systemname* GROUP, where *systemname* is the name of the user's GCS saved system.

Group Control System (GCS). A component of VM/SP, consisting of a shared segment that the user can IPL and run in a virtual machine. It provides simulated MVS services and unique supervisor services to help support a native SNA network.

I

IML. Initial microprogram load.

initial microprogram load (IML). The action of loading a microprogram, which is a set of elementary instructions maintained in special storage that corresponds to a specific computer operation.

initial program load (IPL). The initialization procedure that causes an operating system to begin operation. A VM user must IPL the specific operating system into the virtual machine that will control the user's work. Each virtual machine can be loaded with a different operating system.

initialize. To set counters, switches, addresses, or contents of storage to starting values.

Interactive Problem Control System (IPCS). A component of VM/SP that permits online problem management, interactive problem diagnosis, online debugging for disk related CP or virtual machine abend dumps or CPTRAP files, problem tracking, and problem reporting.

interrupt. A suspension of a process, such as execution of a computer program, caused by an external event and done in such a way that the process can be resumed.

IPCS. Interactive Problem Control System.

IPL. Initial program load.

IPL processor. In an AP or MP system, the processor on which the control program was first initialized during system generation. Note that both the IPL and the non-IPL processors in a real MP configuration have I/O capabilities.

L

load. In reference to installation and service, to move files from tape to disk, auxiliary storage to main storage, or minidisks to virtual storage within a virtual machine.

loader. A routine, commonly a computer program, that reads data into main storage.

load map. A map containing the storage addresses of control sections and entry points of a program loaded into storage.

local. Two entities (for example, a user and a server) are said to be local to each other if they belong to the same system within a collection or to the same node within an SNA system. Contrast with *remote*.

logical line. A command or data line that can be separated from one or more additional command or data lines on the same input line by a logical line end symbol.

logical saved segment. A portion of a physical saved segment that CMS can manipulate. Each logical segment can contain different types of program objects, such as modules, text files, execs, callable services libraries, language repositories, user-defined objects, or a single minidisk directory. A system segment identification file (SYSTEM SEGID) associates a logical saved segment to the physical saved segment in which it resides. See *physical saved segment* and *saved segment*.

logical unit (LU). An entity addressable within an SNA-defined network, similar to a node within a VM network. LUs are categorized by the types of communication they support. A TSAF collection in an SNA network is viewed as one or more LUs.

M

map. In CMS, the file that contains a CMS output listing, such as (1) a list of macros in the MACLIB library, including macro size and location within the library, (2) a listing of the directory entries for the DOS/VS system or private source, relocatable, or core image libraries, (3) a linkage editor map for CMS/DOS programs, and (4) a module map containing entry point locations.

message repository. A source file that contains message texts for a VM component or user application. It is compiled into internal form by the GENMSG command. The message text in a repository file can be translated and used to support national languages.

MIH. Missing interrupt handler.

minidisk. A logical subdivision (or all) of a physical disk pack that has its own virtual device address, consecutive virtual cylinders (starting with virtual cylinder 0), and a VTOC or disk label identifier. Each user virtual disk is preallocated and defined by a VM/SP directory entry as belonging to a user.

minidisk directory. Synonym for *CMS file directory*.

missing interrupt handler (MIH). A VM/SP facility that detects incomplete I/O conditions by monitoring I/O activity. It also tries to correct incomplete I/O conditions without operator intervention.

module. (1) A unit of a software product that is discretely and separately identifiable with respect to modifying, compiling, and merging with other units, or with respect to loading and execution. For example, the input to, or output from, a compiler, the assembler, the linkage editor, or an exec routine. (2) A nonrelocatable file whose external references have been resolved.

MP. Multiprocessor.

multiprocessor (MP). A computer using two or more processing units under integrated control.

N

node. (1) A single processor or a group of processors in a teleprocessing network. (2) A computer, workstation, or printer, when it is participating in a network.

node ID. Node identifier.

node identifier (node ID). The name by which a node is known to all other nodes in a network.

non-IPL processor. In an AP or MP system, the attached or second processor initialized at system generation time. Note that both the IPL processor and the non-IPL processor in a real MP configuration have I/O capabilities.

nucleus. The part of CP, CMS, or GCS resident in main storage.

O

object code. Compiler or assembler output that is executable machine code or is suitable for more processing to produce executable machine code. Contrast with *source code*.

operand. Information entered with a command name to define the data on which a command processor operates and to control the execution of the command processor.

overhead. The additional processor time charged to each virtual machine for the CP functions needed to simulate the virtual machine environment and for paging and scheduling time.

overlay. The technique of repeatedly using the same areas of internal storage during different stages of a program.

override. Synonym for *component parameter override.*

override area. Synonym for *component override area.*

override file. Synonym for *class override file* and *product parameter override file.*

P

page. A fixed-length block that has a virtual address and can be transferred between real storage and auxiliary storage.

page number. The part of a virtual storage address needed to refer to a page.

page zero. Storage locations 0 to 4095.

paging. Transferring pages between real storage and external page storage.

paging area. An area of direct access storage (and an associated area of real storage) that CP uses for the temporary storage of pages when paging occurs.

parameter. A variable that is given a constant value for a specified application and that may denote the application.

physical saved segment. One or more pages of storage that have been named and retained on a CP-owned volume (DASD). Once created, it can be loaded within a virtual machine's address space or outside a virtual machine's address space. Multiple users can load the same copy. A physical saved segment can contain one or more logical saved segments. A system segment identification file (SYSTEM SEGID) associates a physical saved segment to its logical saved segments. See *logical saved segment* and *saved segment.*

POOLDEF file. A CMS file with a file type of POOLDEF that contains information that a file pool server uses to locate all the minidisks in the file pool.

printer universal character set. A printer feature that permits a variety of character arrays. Synonymous with *universal character set.*

private storage. A combination of application code and GCS code available to only one particular virtual machine. No virtual machine can access or share another's private storage area.

privilege class. One or more classes assigned to a virtual machine user in a VM/SP directory entry; each privilege class specified lets a user access a logical subset of the CP commands. There are eight IBM-defined privilege classes that correspond to specific administrative functions. They are:

Class A	- primary system operator
Class B	- system resource operator
Class C	- system programmer
Class D	- spooling operator
Class E	- system analyst
Class F	- service representative
Class G	- general user
Class H	- reserved for IBM use
Class Any	- available to any user.

The privilege classes can be changed to meet the needs of an installation. See *class authority* and *user class restructure (UCR).*

Procedures Language/VM. A component of VM/SP. It contains the VM/SP System Product Interpreter, which processes the REXX language. This component contains the VM/SP implementation of the Systems Application Architecture™ Procedures Language in addition to the VM/SP System Product Interpreter function available in VM/SP Releases 3, 4, and 5. Procedures Language/VM provides a single source base for the VM/SP System Product Interpreter in both the CMS and GCS components.

product parameter file. A file containing installation and service parameters for a product: control options, minidisk and SFS directory assignments, and component part type/function lists.

product parameter override file. A file containing one or more component override areas.

PROFILE EXEC. A special EXEC procedure with a file name of PROFILE that a user can create. The procedure is usually executed immediately after CMS is loaded into a virtual machine (also known as IPL CMS).

program function (PF) key. On a terminal, a key that can do various functions selected by the user or determined by an application program.

program status word (PSW). An area in storage that indicates the order in which instructions are executed,

Trademark of International Business Machines Corporation and to hold and indicate the status of the computer system.

PSW. Program status word.

R

real address. The address of a location in real storage or the address of a real I/O device.

real machine. The actual processor, channels, storage, and I/O devices required for VM/SP operation.

recovery machine. The first machine to join a virtual machine group. It has responsibility for executing routines that were set with the GCS MACHEXT macro and cleaning up system resources when machines leave the group.

remote. Two entities (for example, a user and a server) are said to be remote to each other if they belong to different systems within a collection, or to different nodes within an SNA network. Contrast with *local.*

resource. A program, a data file, a specific set of files, a device, or any other entity or a set of entities that the user can uniquely identify for application program processing in a VM system.

resource ID. A one-to-eight character name that identifies a resource.

resource manager. An application running in a server virtual machine that directly controls one or more VM resources. There are three categories of VM resource managers: global, local, and private.

Restructured Extended Executor (REXX) language. A general purpose programming language, particularly suitable for EXEC procedures, XEDIT macros, or programs for personal computing. Procedures, XEDIT macros, and programs written in this language can be interpreted by the System Product Interpreter.

REXX language. Restructured Extended Executor language.

S

saved segment. A segment of storage that has been saved and assigned a name. The saved segment(s) can be physical saved segment(s) that CP recognizes or logical saved segments that CMS recognizes. The segments can be loaded and shared among virtual machines, which helps use real storage more efficiently, or a private, nonshared copy can be loaded into a virtual machine. See *logical saved segment* and *physical saved segment.*

saved system. A special nonrelocatable copy of a virtual machine's virtual storage and associated registers kept on a CP-owned disk and loaded by name instead of by I/O device address. Loading a saved system by name substantially reduces the time it takes to IPL the system in a virtual machine. In addition, a saved system such as CMS can also share one or more 64K segments of reenterable code in real storage between virtual machines. This reduces the cumulative real main storage requirements and paging demands of such virtual machines.

secondary user. When a user is disconnected — that is, has no virtual console on line — a secondary user can be designated to receive the disconnected user's console messages and to enter commands to the disconnected user's console.

segment. A contiguous 64K or 1024K area of virtual storage (not necessarily contiguous in real storage) allocated to a job or system task. VM/SP does not use 1024K segments, but supports any VM operating system that uses 1024K segments.

segment number. The part of a virtual storage address needed to refer to a segment.

server. The general name for a virtual machine that provides a service for a requesting virtual machine.

SFS. Shared file system.

SFS directory. A group of files. SFS directories can be arranged to form a hierarchy in which one directory can contain one or more subdirectories as well as files.

shared file system (SFS). A part of CMS that lets users organize their files into groups known as *directories* and to selectively share those files and directories with other users.

shared segment. A feature of a saved system or physical saved segment that lets one or more segments of reentrant code or data in real storage be shared among many virtual machines. For example, if a saved CMS system was generated, the CMS nucleus is shared in real storage among all CMS virtual machines loaded by name; that is, every CMS machine's segment of virtual storage maps to the same 64K of real storage. See *discontiguous saved segment* and *saved system.*

simultaneous peripheral operations online (SPOOL). (1) (Noun) An area of auxiliary storage defined to temporarily hold data during its transfer between peripheral equipment and the processor. (2) (Verb) To use auxiliary storage as a buffer storage to reduce processing delays when transferring data between peripheral equipment and the processing storage of a computer.

single console image facility (SCIF). (1) Lets a user, who is disconnected from a primary virtual console,

continue to have console communications by way of the console of the secondary user. See *secondary user*.
(2) Enables a virtual machine operator to control multiple virtual machines from one physical terminal.

SNA. Systems Network Architecture.

source code. The input to a compiler or assembler, written in a source language. Contrast with *object code*.

source file. A file that contains source statements for such items as high-level language programs and data description specifications.

SPOOL. Simultaneous peripheral operations online.

spool file class. A one-character class associated with each virtual unit record device. For input spool files, the spool file class lets the user control which input spool files are read next; and, for output spool files, it lets the spooling operator better control or reorder the printing or punching of spool files having similar characteristics or priorities. The spool file class value can be A through Z or 0 through 9.

spool ID. A spool file identification number automatically assigned by CP when the file is closed. The spool ID number can be from 0001 to 9900; it is unique for each spool file. To identify a given spool file, a user must specify the owner's user ID, the virtual device type, and the spool ID.

S-STAT. A block of storage that contains the FSTs associated with the S-disk. The FSTs are sorted so that a binary search can search for files. The S-STAT usually resides in the CMS nucleus so it can be shared. Only files with file mode of 2 will have their associated FSTs in the S-STAT.

stand-alone. Pertaining to an operation independent of another device, program, or system.

stand-alone dump. A dump acquired without regular system functions. For example, to obtain a CP dump when the regular system is unable to dump the machine, the stand-alone dump facility gets a CP stand-alone dump.

stand-alone program. A program independent of any other program or system.

starter system. A very basic VM/SP system that the user can use to build a production VM/SP system.

starter system tape. The tape that contains the starter system. Starter system tapes are DASD-type specific; for instance, a 3350 starter system cannot be used on any DASD type other than a 3350.

system name table. In CP, the table that contains the name and location of saved systems, including shared and nonshared segments.

System Product Interpreter. The language processor of the VM/SP operating system that processes procedures, XEDIT macros, and programs written in the REXX language.

system profile. An EXEC (SYSPROF) that resides in a saved system or on a system disk and called by CMS initialization. It contains some initialization functions, and provides a means for installations to override the default CMS environment by tailoring the exec to suit the installation.

system restart. The restart that allows reuse of previously initialized areas. System restart usually requires less time than IPL. See *warm start*.

Systems Application Architecture. A defined set of interfaces, conventions, and protocols that can be used across various IBM systems.

Systems Network Architecture (SNA). The description of the logical structure, formats, protocols, and operational sequences for transmitting information units through and controlling the configuration and operation of networks.

T

time-of-day (TOD) clock. A hardware feature required by VM/SP. The TOD clock is incremented once every microsecond, and provides a consistent measure of elapsed time suitable for the indication of date and time; it runs regardless of the processor state (running, wait, or stopped).

TOD clock. Time-of-day clock.

Transparent Services Access Facility (TSAF). A component of VM/SP that handles communication between systems by letting APPC/VM paths span multiple VM systems. TSAF lets a source program connect to a target program by specifying a name that the target has made known, instead of specifying a user ID and a node ID.

TSAF. Transparent Services Access Facility (TSAF).

TSAF collection. A group of VM processors, each with a TSAF virtual machine, connected by CTC, binary synchronous lines, or LANs.

TSAF virtual machine. The virtual machine that lets user programs connect to and communicate with virtual machines on different VM systems.

U

UCR. User class restructure.

universal character set (UCS). A printer feature that permits a variety of character arrays. Synonym for *printer universal character set.*

user class. A privilege category assigned to a virtual machine user in the user's directory entry; each class specified allows access to a logical subset of all the CP commands. See*privilege class.*

user class restructure (UCR). The extension of the class structure of CP instructions from 8 to 32 classes for each user, command, and diagnose code within the system. This extension allows the installation greater flexibility in authorizing CP instructions.

V

virtual address. The address of a location in virtual storage. A virtual address must be translated into a real address in order to process the data in processor storage.

virtual console. A console simulated by CP on a terminal such as a 3270. The virtual device type and I/O address are defined in the VM/SP directory entry for that virtual machine.

virtual console spooling. The writing of console I/O on disk as a printer spool file instead of, or in addition to, having it typed or displayed at the virtual machine console. The console data includes messages, responses, commands, and data from or to CP and the virtual machine operating system. The user can invoke or terminate console spooling at anytime. When the console spool file is closed, it becomes a printer spool file. Synonymous with *console spooling.*

virtual machine group. The concept in GCS of two or more virtual machines associated with each other through the same named system (for example, IPL GCS1). Virtual machines in a group share common read/write storage and can communicate with one another through facilities provided by GCS. Synonymous with *group.*

virtual=real area. The part of real storage, starting with real page 1, where a virtual=real machine can execute. CP maintains control of real page zero; only page zero of the virtual=real machine is relocated. Only one virtual machine at a time can occupy the virtual=real area. The area must be defined during VM/SP system generation to contain the largest virtual=real machine likely to run. See *virtual=real option.*

virtual=real option. A VM/SP performance option that lets a virtual machine run in VM/SP's virtual=real area. This option eliminates CP paging and, optionally, CCW translation for this virtual machine. Synonymous with *V=R.*

virtual storage. Storage space that can be regarded as addressable main storage by the user of a computer system in which virtual addresses are mapped into real addresses. The size of virtual storage is limited by the addressing scheme of the computing system and by the amount of auxiliary storage available, and not by the actual number of main storage locations.

virtual storage access method (VSAM). An access method for direct or sequential processing of fixed and variable-length records on direct access devices. The records in a VSAM data set or file can be organized in logical sequence by a key field (key sequence), in the physical sequence in which they are written on the data set or file (entry-sequence), or by relative-record number.

virtual storage extended (VSE). The generalized term that indicates the combination of the DOS/VSE system control program and the VSE/Advanced Functions program product. Note that in certain cases, the term DOS is still used as a generic term; for example, disk packs initialized for use with VSE or any predecessor DOS or DOS/VS system are sometimes called DOS disks. Also note that the DOS-like simulation environment provided under the VM/SP CMS component and CMS/DOS exists on VM/SP and VM/SP HPO program products and continues to be called CMS/DOS.

Virtual Telecommunications Access Method (VTAM). An IBM licensed program that controls communication and the flow of data in a computer network. It provides single-domain, multiple-domain, and multiple-network capability. VTAM runs under MVS, OS/VS1, VM/SP, and VSE.

VMLIB. The name of the CSL supplied with VM/SP and that contains routines to do various VM functions.

VM/SP directory. A CP disk file that defines each virtual machine's typical configuration; the user ID, password, regular and maximum allowable virtual storage, CP command privilege class or classes allowed, dispatching priority, logical editing symbols to be used, account number, and CP options desired. Synonymous with *CP directory.*

volid. Volume identifier.

volume identifier (volid). The volume identification label for a disk.

V=R. Synonym for *virtual=real option.*

VSAM. Virtual storage access method.

VSE. Virtual storage extended.

VTAM. Virtual Telecommunications Access Method.

W

warm start. (1) The result of an IPL that does not erase previous system data. (2) The automatic reinitialization of the VM/SP control program that occurs if the control program cannot continue processing. Closed spool files and the VM/SP accounting information are not lost. Contrast with *checkpoint (CKPT) start, cold start*, and *force start*.

Y

Y-STAT. A block of storage that contains the FSTs associated with file mode Y. The FSTs are sorted so that a binary search can search for files. The Y-STAT usually resides in the CMS nucleus so it can be shared. Only files with file mode of 2 will have their associated FSTs in the Y-STAT.

Bibliography

Related Publications

This section lists IBM books that can help you plan and install your system. The *VM/SP Library Guide and Master Index*, GC19-6207 describes the books in the VM/SP library and provides an index to all of them. If you do not see the book you need in the following list, you might want to check the *IBM System/370, 30xx, 4300, and 9370 Processors Bibliography*, GC20-0001.

- **Planning**

  Virtual Machine/System Product Release 6:

  Release 6 Guide, SC24-5368
  Planning Guide and Reference, SC19-6201
  CMS Shared File System Administration, SC24-5367
  Connectivity Planning, Administration, and Operation, SC24-5378

  Remote Spooling Communications Subsystem Networking Version 2:

  Planning and Installation, SH24-5057

  Other Non-VM/SP titles:

  Using IBM 3380 Direct Access Storage in a VM Environment, GC26-4493

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

  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 - 5742-0J7) SCP 5742, 5744-AN1/BA2, 5747-AG1/AJ2, GC30-3008

- **Installation**

  Virtual Machine/System Product Release 6:

  Planning Guide and Reference, SC19-6201
  CMS Shared File System Administration, SC24-5367
  Connectivity Planning, Administration, and Operation, SC24-5378
  Service Guide, SC24-5389

  Remote Spooling Communications Subsystem Networking Version 2:

  Planning and Installation, SH24-5057

  Other Non-VM/SP titles:

  Using IBM 3380 Direct Access Storage in a VM Environment, GC26-4493

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

  VM/VTAM VTAM Installation and Resource Definition, SC23-0111

  VM/VTAM Communication Network Application Installation, Operation, and Terminal Use, SC27-0502

  ACF/NCP-SSP, V3 Installation and Resource Definition Guide, SC30-3253

  EP/3725 Installation and Resource Definition Guide and Reference, SC30-3172

  EP/3705 Generation and Utilities Guide and Reference, GC30-3242

ACF/NCP V4, ACF/SSP V3 Diagnosis Guide, SC30-3255

- **Administration, Operation, and Programming**

  Virtual Machine/System Product Release 6:

  Introduction, SC19-6200
  Operator's Guide, SC19-6202
  System Messages and Codes, SC19-6204
  Terminal Reference, GC19-6206
  CMS Command Reference, SC19-6209
  CP General User Command Reference, SC19-6211
  Interactive Problem Control System Guide and Reference, SC24-5260
  Application Development Reference for CMS, SC24-5284
  Administration, SC24-5285
  Application Development Guide for CMS, SC24-5286
  CMS Shared File System Administration, SC24-5367
  Connectivity Programming Guide and Reference, SC24-5377
  Connectivity Planning, Administration, and Operation, SC24-5378
  Service Guide, SC24-5389
  CP System Command Reference, SC24-5402
  Diagnosis Guide, LY24-5241

  Virtual Machine:

  System Facilities for Programming, SC24-5288
  Running Guest Operating Systems, GC19-6212

  Remote Spooling Communications Subsystem Networking Version 2:

  Operation and Use, SH24-5058

  Other non-VM/SP titles:

  3704 and 3705 Communications Controllers

  Introduction to the IBM 3704 and 3705 Communications Controllers, GA27-3051
  IBM 3704 Control Panel Guide, GA27-3086
  IBM 3705 Control Panel Guide, GA27-3087

  3800 Printing Subsystem

  *Introducing the 3800 Printing Subsystem,*GC26-3829
  Introducing the IBM 3800 Model 3 Printing Subsystem, GA32-0049
  Concepts of the IBM 3800 Printing Subsystem, GC20-1775
  Reference Manual for the IBM 3800 Printing Subsystem, GA26-1635
  IBM 3800 Printing Subsystem Programmer's Guide, GC26-3846
  Reference Manual for the IBM 3800 Printing Subsystem Model 3, GA32-0050
  IBM Printing Subsystem Model 3 Programmer's Guide: Compatibility, SH35-0051

  3850 Mass Storage System

  IBM 3850 Mass Storage System (MSS) Principles of Operation: Theory, GA32-0035
  IBM 3850 Mass Storage System (MSS) Principles of Operation: Reference, GA32-0036

  EREP

  EREP User's Guide and Reference, GC28-1378

Miscellaneous

3270 Information Display System Library User's Guide, GA23-0058
IBM OS/VS Linkage Editor and Loader, GC26-3813
Input/Output Configuration Program User's Guide and Reference, GC28-1027
Device Support Facilities User's Guide and Reference, GC35-0033

VM/SP RELEASE 6 LIBRARY

Evaluation

	General Information	GC20-1838
▨	Introduction	GC19-6200
	Introduction to Security	SC24-5316

Installation and Service

▨	Installation Guide	SC24-5237
▨	Service Guide	SC24-5389

Planning

▨	Planning Guide and Reference	SC19-6201
	VM Running Guest Operating Systems	GC19-6212
▨	Release 6 Guide	SC24-5368

Operation

▨	Operator's Guide	SC19-6202

Administration

▨	VM System Facilities for Programming	SC24-5288
▨	Administration	SC24-5285
▨	Connectivity Planning, Administration, and Operation	SC24-5378
▨	CMS Shared File System Administration	SC24-5367
▨	CP System Command Reference	SC24-5402

Application Development

▨	Application Development Guide for FORTRAN and COBOL	SC24-5247
▨	Programmer's Guide to the SRPI for VM/SP	SC24-5291
▨	Application Development Guide for CMS	SC24-5286
▨	Application Migration Guide for CMS	SC24-5366
▨	Application Development Reference for CMS	SC24-5284
▨	Connectivity Programming Guide and Reference	SC24-5377
▨	GCS Command and Macro Reference	SC24-5250
▨	**SAA Common Programming Interface Communications Reference**	**SC26-4399**
▨	Directory of Programming Interfaces for Customers	GC24-5417

Index/Glossary

▨	Library Guide and Master Index	GC19-6207
	Glossary	SC24-5379

End Use

	Quick Reference	SX20-4400
▨	CMS Primer	SC24-5236
	CMS Primer for Line-Oriented Terminals	SC24-5242
▨	CMS User's Guide	SC19-6210
▨	CMS Command Reference	SC19-6209
▨	System Product Editor User's Guide	SC24-5220
▨	System Product Editor Command and Macro Reference	SC24-5221
▨	System Product Interpreter User's Guide	SC24-5238
▨	System Product Interpreter Reference	SC24-5239
▨	EXEC 2 Reference	SC24-5219
▨	CP General User Command Reference	SC19-6211
▨	Terminal Reference	GC19-6206

▨ one copy of each shaded manual automatically received with product tape

VM/SP RELEASE 6 LIBRARY

Diagnosis

- System Messages and Codes SC19-6204
- System Messages Cross-Reference SC24-5264
- Service Routines Program Logic LY20-0890
- Interactive Problem Control System Guide and Reference SC24-5260
- Diagnosis Guide LY24-5241
- CP Diagnosis Reference LY20-0892
- CP Data Areas and Control Blocks LY24-5220
- CMS Diagnosis Reference LY20-0893
- CMS Data Areas and Control Blocks LY24-5221
- VM CP Trace Table (Poster) SX24-5225
- Problem Determination Summary SX24-5224

Reference Summaries

- SP Editor Command Language Reference Summary SX24-5122
- EXEC 2 Language Reference Summary SX24-5124
- CMS Primer Summary of Commands SX24-5151
- SP Interpreter Reference Summary SX24-5126
- CMS Primer for Line-Oriented Terminals Summary of Commands SX24-5159
- HELP Facility Introduction SX24-5221
- CP General User Command Reference Summary SX24-5219
- CP System Command Reference Summary SX24-5222
- CMS Command Reference Summary SX24-5220
- VM Summary of End Use Tasks and Commands (Poster) SX24-5173

Auxiliary Communication Support

- VTAM Installation and Resource Definition SC23-0111
- VTAM Customization SC23-0112
- VTAM Operation SC23-0113
- VTAM Messages and Codes SC23-0114
- VTAM Programming SC23-0115
- VTAM Diagnosis Guide LY30-5601
- VTAM Programming for LU 6.2 SC30-3400
- VTAM Data Areas (VM) LY30-5593
- VTAM Reference Summary LY30-5600
- VM/Pass-Through Facility Overview GC24-5373
- VM/Pass-Through Facility: Managing and Using SC24-5374
- RSCS Exit Customization SH24-5197
- RSCS General Information GH24-5055
- RSCS Planning and Installation SH24-5057
- RSCS Messages and Codes SH24-5196
- RSCS Operation and Use SH24-5058
- RSCS Diagnosis Reference LY24-5228
- RSCS Reference Summary SX24-5135

Index

Program Number
5664-167

File Number
S370/4300-34

Printed in U.S.A.

VM/SP
Installation Guide
Order No. SC24-5237-04

READER'S COMMENT FORM

Is there anything you especially like or dislike about this book? Feel free to comment on specific errors or omissions, accuracy, organization, or completeness of this book.

If you use this form to comment on the online HELP facility, please copy the top line of the HELP screen.

______________ ______________ _______ Help Information line ____ of ____

IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you, and all such information will be considered nonconfidential.

Note: Do not use this form to report system problems or to request copies of publications. Instead, contact your IBM representative or the IBM branch office serving you.

Would you like a reply? ___YES ___NO

Please print your name, company name, and address:

____

____

____

____

IBM Branch Office serving you: ____

Thank you for your cooperation. You can either mail this form directly to us or give this form to an IBM representative who will forward it to us.

SC24-5237-04

Reader's Comment Form

CUT
OR
FOLD
ALONG
LINE

Fold and tape Please Do Not Staple Fold and tape

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NY

POSTAGE WILL BE PAID BY ADDRESSEE:

IBM

INTERNATIONAL BUSINESS MACHINES CORPORATION
DEPARTMENT G60
PO BOX 6
ENDICOTT NY 13760-9987

Fold and tape Please Do Not Staple Fold and tape

IBM®

Program Number
5664-167

File Number
S370/4300-34

SC24-5237-04

Printed in U.S.A.